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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2015 (19.03.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/035492 A1

(51) International Patent Classification:
H04R 3/04 (2006.01) *H04S 3/00* (2006.01)
G10L 19/008 (2013.01)

(21) International Application Number:
PCT/CA2013/050706

(22) International Filing Date:
13 September 2013 (13.09.2013)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MIXGENIUS INC.** [CA/CA]; 809-160 Saint-Viateur Street East, Montreal, Québec H2T 1A8 (CA).

(72) Inventors: **MANSBRIDGE, Stuart P.**; North View, Renwick, Penrith, Cumbria CA10 1JL (GB). **REISS, Joshua D.**; 26 Regency Court, Park Close, London E9 7TP (GB). **DE MAN, Brecht**; William Goodenough House 5123, Mecklenburgh Square, London WC1N 2AN (GB). **PESTANA, Pedro Duarte Leal Gomes**; Rua das Flores 5, 4º esq, P-2800-078 Almada (PT). **HAFEZI, Seyed Sina**; 24 Holland Road, London NW10 5AU (GB).

(74) Agents: **GEST, Johann** et al.; Fasken Martineau DuMoulin LLP, Stock Exchange Tower, Suite 3700, P.O. Box 242, 800 Place Victoria, Montreal, Quebec H4Z 1 E9 (CA).

(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, 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,

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR PERFORMING AUTOMATIC MULTI-TRACK AUDIO MIXING

(57) Abstract: Methods of performing equalization of audio tracks, performing gain compensation, generating a target spectrum for equalization of audio content, and performing panning and compression of audio tracks are provided. In at least one method, frequency bands are analyzed and compared with corresponding bands of at least one other track or audio file to perform equalization.

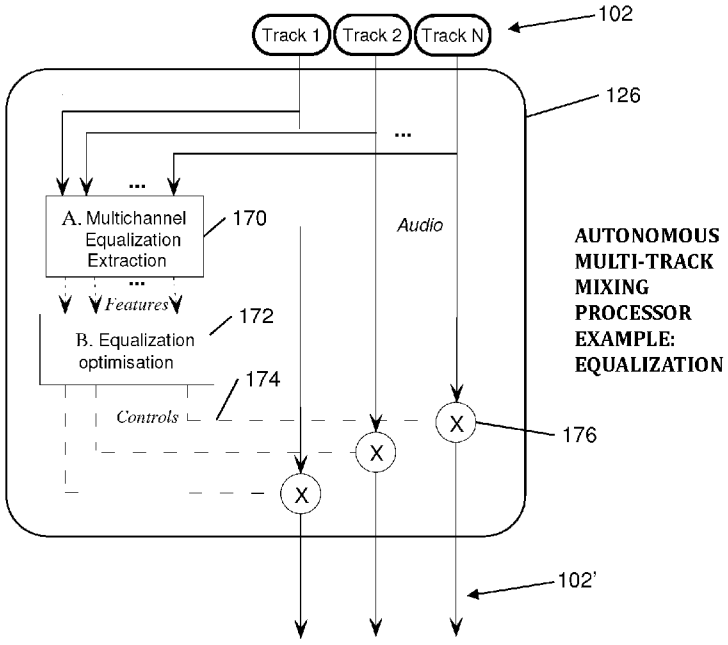


FIG. 6

WO 2015/035492 A1

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:

— *with international search report (Art. 21(3))*

SYSTEM AND METHOD FOR PERFORMING AUTOMATIC MULTI-TRACK AUDIO MIXING

TECHNICAL FIELD

[0001] The following relates to systems and methods for performing multi-track automatic audio mixing.

DESCRIPTION OF THE RELATED ART

[0002] In the field of both sound recording and live sound production it is common to process multiple tracks of audio simultaneously, each track corresponding to a separate audio signal. Many live and studio multi-track audio production tasks require dynamic user adjustment of various sound editing and manipulation parameters in order to combine multiple audio tracks into a high quality mixture. In a studio environment, such multi-track processing can be time consuming, even to a very skilled audio engineer. Furthermore, in a live environment, the real-time nature of the processing means that there is scope for error when determining how to edit and adjust parameters of the multiple tracks. A need has therefore developed to assist engineers, particularly in the live-mixing environment, in minimising the difficulty of their work.

[0003] Some digital mixing desks have been developed that allow for user controlled parameters, such as fader levels, or pan positions, to be saved. As such, when a live audio engineer is dealing with multiple bands, or multiple arrangements of instruments, the engineer can pre-set the parameters, for example during a sound check, and thereby reduce some of the complexity of his or her job. However, there is still a need in such systems for the engineer to adjust these parameters during a performance due to fluctuations in track levels.

[0004] Automatic mixing systems have been developed to automate various tasks, such as balancing levels, panning signals between channels, dynamic range compression, and equalization. However, current systems have not provided a way to assist the audio engineer with managing the control of multiple parameters for multiple tracks.

SUMMARY

[0005] In one aspect, there is provided a method of performing automatic equalization in an automatic multi-track audio mixing system, the method comprising: dividing a frequency spectrum for an audio track into a plurality of bands; determining a value representative of a spectral characteristic in each band; comparing the value in each band to a corresponding band in another signal; and applying a filter to at least one band in either the audio track or the other signal based on the comparison.

[0006] In another aspect, there is provided a method of performing automatic mastering equalization of audio tracks, the method comprising: dividing a frequency spectrum for an audio signal into a plurality of bands; determining a value representative of each band for the audio signal; comparing the value in each band of the audio signal with corresponding bands in a target spectrum; and applying at least one filter to the audio signal to perform an equalization towards the target spectrum.

[0007] In another aspect, there is provided a method of performing gain compensation, the method comprising: determining a first filter response to be applied to a plurality of frequency bands in a signal; computing a first gain to be applied at each frequency band; computing a second filter response using the first gains to be applied at each band and corresponding target gains; and applying the second filter response.

[0008] In another aspect, there is provided a method of generating a target spectrum for equalization of audio content, the method comprising: determining a magnitude spectrum for a target audio file; determining an average magnitude spectrum in each of the bands; and generating the target spectrum and storing as a predetermined target profile.

[0009] In another aspect, there is provided a method of performing equalization of audio tracks, the method comprising: obtaining a plurality of audio tracks; for at least one pair of tracks: comparing each of a plurality of frequency bands in a frequency spectrum for one track to corresponding frequency bands in another track; applying at least one rule according to which tracks are being compared, the at least one rule being indicative of a type of filtering to be performed; and determining at least one filter to be applied to at least one of the pair of tracks; and applying the at least one filter.

[0010] In another aspect, there is provided a method of performing equalization of audio tracks, the method comprising: obtaining a plurality of audio tracks; and for at least one track: determining a cut-off frequency for setting a high pass filter; and applying the high pass filter to the at least one track to remove low frequency content in the track.

[0011] In other aspects, there are provided systems, devices, and computer readable media configured to perform the above methods.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Embodiments will now be described by way of example only with reference to the appended drawings wherein:

[0013] FIG. 1 is a block diagram of an example of an autonomous multi-track music production system and a semantic processing module for such a system;

[0014] FIG. 2 is a block diagram illustrating an autonomous multi-track music production system having a semantic processing module;

[0015] FIG. 3 is a block diagram illustrating a multi-track subgroup for an autonomous multi-track music production system;

[0016] FIG. 4 is a block diagram illustrating a cross adaptive feature processing element for an autonomous multi-track music production system;

[0017] FIG. 5 is a block diagram illustrating an example multi-track loudness processor for an autonomous multi-track music production system;

[0018] FIG. 6 is a block diagram illustrating an example multi-track equalization processor for an autonomous multi-track music production system;

[0019] FIG. 7 is a block diagram illustrating an example of a configuration for a multi-track equalization optimization module;

[0020] FIG. 8 is a graph illustrating a pair of audio tracks being compared;

[0021] FIG. 9 is a graph illustrating a pair of audio tracks being compared in another example;

[0022] FIG. 10A is a segmented frequency spectrum for one of the tracks shown in FIG.9;

[0023] FIG. 10B is a segmented frequency spectrum for the other of the tracks shown in FIG.9;

[0024] FIG. 11 is a graph illustrating a dB spread between tracks in a particular frequency band;

[0025] FIG. 12 is a flow chart illustrating an example of computer executable instructions that may be executed in de-masking one or more tracks in an audio input;

[0026] FIG. 13 is a flow chart illustrating an example of computer executable instructions that may be executed in removing low frequency components from one or more tracks in an audio input;

[0027] FIG. 14 is a flow chart illustrating an example of computer executable instructions that may be executed in removing noise from an audio track;

[0028] FIG. 15A is a flow chart illustrating an example of computer executable instructions that may be executed in removing harsh sound components from an audio track;

[0029] FIG. 15B is a block diagram illustrating a de-essing process;

[0030] FIG. 16 is a block diagram illustrating an example multi-track master equalization processor for an autonomous multi-track music production system;

[0031] FIG. 17 is a graph illustrating a frequency spectrum for a track against a target spectrum;

[0032] FIG. 18 is a graph illustrating signal levels for the track in FIG. 17 within each of a plurality of frequency bands;

[0033] FIG. 19 is a graph illustrating signal levels for the target in FIG. 17 within each of the plurality of frequency bands used in FIG. 18;

[0034] FIG. 20 is a flow chart illustrating an example of computer executable instructions that may be executed in performing a master equalization;

[0035] FIG. 21 is a graph illustrating a gain compensation applied to a band in a frequency spectrum for an audio track;

[0036] FIG. 22 illustrates an iterative process of refining gain compensation to account for unwanted affects in neighbouring frequency bands;

[0037] FIGS. 23A and 23B illustrate an application of a gain compensation process;

[0038] FIG. 24 provides a series of charts illustrating centre frequencies and Q factors;

[0039] FIG. 25 is a flow chart illustrating an example of computer executable instructions that may be executed in performing a gain compensation in a master equalization;

[0040] FIG. 26 is a flow chart illustrating an example of computer executable instructions that may be executed in generating a target profile for audio content;

[0041] FIG. 27 is a block diagram illustrating an example multi-track pan positioning processor for an autonomous multi-track music production system;

[0042] FIG. 28 is a block diagram illustrating an example of a pan positioning plug-in configuration;

[0043] FIG. 29 is a graph showing pan positions in stereo field depending on frequency;

[0044] FIG. 30 illustrates panning positions with consistent spread for different values of maximum spectral centroid, and the effect of panning width control;

[0045] FIG. 31 illustrates an equal power sine-cosine -3dB panning law;

[0046] FIG. 32 illustrates a screen shot of an example user interface for an automatic panning plug-in;

[0047] FIG. 33 illustrates mean and 85% confidence interval results for individual genre and mix type in one case;

[0048] FIG. 34 illustrates mean and 85% confidence interval results for individual genre and mix type in another case;

[0049] FIG. 35 illustrates overall mean and medium results for the two cases shown in FIGS. 33 and 34;

[0050] FIG. 36 is a block diagram illustrating an example multi-track compression processor for an autonomous multi-track music production system;

[0051] FIG. 37 is a block diagram of an example configuration for a multi-track dynamic range compression processor;

[0052] FIGS. 38A and 38B illustrate input-output characteristics of an automatic dynamic range compressor in a threshold mode and a ratio mode respectively;

[0053] FIGS. 39A and 39B illustrate post-dynamic range compression loudness ranges in a threshold mode and a ratio mode respectively;

[0054] FIG. 40 illustrates an example of pre- and post-dynamic range compression LRA for a multi-track audio recording;

[0055] FIG. 41 is an example of a user interface for the manual application of multi-track dynamic range compression;

[0056] FIG. 42 illustrates a comparison of loudness range reductions for manual and automatic dynamic range compression;

[0057] FIG. 43 is an example of a user interface for performing subjective listening tests;

[0058] FIG. 44 illustrates mean scores and 90% confidence intervals for appropriateness of relative amounts of dynamic range compression during a listening test;

[0059] FIG. 45 illustrates mean scores and 90% confidence intervals for sound quality of dynamic range compression during a listening test;

[0060] FIG. 46 illustrates mean scores and 90% confidence intervals for overall quality of a mix in a listening test;

[0061] FIG. 47 is a block diagram of a compression configuration in another example;

[0062] FIG. 48 illustrates a static compression characteristic with make-up gain and hard or soft knee;

[0063] FIG. 49 illustrates sine wave with varied amplitude and frequency and crest factor and spectral flux measurement on the sine wave;

[0064] FIG. 50 illustrates crest factor and spectral flux for calculating attack and release times;

[0065] FIG. 51 illustrates compression input/output curves with various knee widths for a set of thresholds;

[0066] FIG. 52 illustrates spectral flux of a drums sample and a bass sample;

[0067] FIG. 53 provides box plots for an attack time evaluation;

[0068] FIG. 54 provides box plots for a release time evaluation;

[0069] FIG. 55 provides box plots for a knee width drums evaluation;

[0070] FIG. 56 provides box plots for a bass sample knee width evaluation;

[0071] FIG. 57 illustrates individual choices for a drums sample knee-width experiment;

[0072] FIG. 58 provides box plots for a make-up gain evaluation;

[0073] FIG. 59 illustrates transfer characteristics of a compressor with different ratios;

[0074] FIG. 60 illustrates the attack and release phases in a compressor;

[0075] FIG. 61 illustrates hard-knee and soft-knee compression;

[0076] FIG. 62 illustrates an example architecture for yet another example configuration for a compressor;

[0077] FIG. 63 illustrates average crest factor and average loudness range per octave;

[0078] FIG. 64 illustrates a comparison of a commercial mix dataset to an uncompressed equal-loudness mix;

[0079] FIG. 65 is a block diagram of a configuration for a compression in the yet another example; and

[0080] FIG. 66 illustrates a curve for a percussivity weighting function.

DETAILED DESCRIPTION

[0081] It will be appreciated that for simplicity and clarity of illustration, where considered appropriate, reference numerals may be repeated among the figures to indicate

corresponding or analogous elements. In addition, numerous specific details are set forth in order to provide a thorough understanding of the examples described herein. However, it will be understood by those of ordinary skill in the art that the examples described herein may be practiced without these specific details. In other instances, well-known methods, procedures and components have not been described in detail so as not to obscure the examples described herein. Also, the description is not to be considered as limiting the scope of the examples described herein.

[0082] It will be appreciated that the examples and corresponding diagrams used herein are for illustrative purposes only. Different configurations and terminology can be used without departing from the principles expressed herein. For instance, components and modules can be added, deleted, modified, or arranged with differing connections without departing from these principles.

[0083] *Autonomous Multi-Track Mixing System:*

[0084] Turning now to FIG. 1, an autonomous multi-track music production system (the "production system 10" hereinafter) is shown, which processes a multi-track audio input 12 and generates an audio output 14 often referred to as a "mix" to be played by a sound system 16. The sound system 16 in turn generates an audio output 18 that is played in a listening space, environment, "room", or other volume of space in which the audio output 18 can be/is played and heard. As shown in FIG. 1, the production system 10 may include an autonomous mixing engine 104 (see also FIGS. 2-5).

[0085] FIG. 2 illustrates further detail for an example production system 10 having a semantic processing module 20, which may be implemented using program instructions or modules within the system 10. The production system 10 includes an incoming data processor 100 for receiving a multi-track audio input 12, e.g., streaming data or a data file and output tracks 102 to be processed. The data file processor 100 processes its input to effectively provide an "audio source" to be input to an autonomous multi-track music production engine 104 (the "engine 104" hereinafter). The engine 104 includes a source control block 106 to perform source recognition and other types of semantic or high-level mixing (e.g. by utilizing a semantic processing module – not shown in FIG. 2), subgroup allocation and genre settings. Source recognition uses machine learning and feature extraction methods to automatically determine the audio source type or instrument. This information can then be used to divide the tracks into subgroups, for example a vocal or percussion subgroup, to form the audio production system. Subgroup allocation and routing can also be controlled externally by the user, and will ultimately feed into a final 'main'

subgroup that outputs the finished stereo mix. Genre settings are also determined by source detection or by user control. This allows each subgroup and the processors contained within to have different parameter settings and pre-sets, depending on the choice or detection of genre. In the typical example shown in FIG. 2, the signals are separated into multiple multi-track subgroups 108 which output the final mixed audio at 110.

[0086] The designation of sub-groups can be achieved automatically using source recognition, such as vocal and percussion detection techniques, or manually based on descriptors or tagging entered by the user(s). The automatic detection techniques are based on machine learning algorithms on numerous low and high-level extracted audio features, and incoming tracks are analyzed in real time and can be judged by their relation to the results of off-line machine learning analysis. Another feature of sub-grouping is the sharing of extracted features between processors, to prevent repeated calculation of extracted features and thus improve efficiency. Additionally, the engine 104 may include an active learning module or related functionality to implement machine learning techniques that adapt to new data input from the user.

[0087] Although not shown in FIG. 2, the production system 10 may also include or provide functionality for an offline analyzer, which may be integrated into the production system 10 to enable a user to conduct offline analyses of audio data. The offline analyzer may be separate from or a component of the system. The offline analyzer contains time stamps of the audio data being analyzed, along with associated data points. The offline analyzer may be configured to generate new long-term extracted features, e.g., for features that require accumulated data over time, different measures using the same extracted features, etc., and that were previously unavailable, such as loudness range, to use in the signal processing algorithms relied upon by the production system 10. For example, locating changes in a song's dynamics using long term measures of loudness, crest factor, etc. can be performed to generate a new extracted feature.

[0088] The offline analyzer may also perform instrument recognition by analyzing each whole track, and then using that knowledge to build the subgroups 108 before running the mix. Previously, real time systems would need some buffering to analyze the incoming audio before being able to generate subgroups 108.

[0089] The offline analyzer may also be used to generate data points by running the audio through the pre-existing feature extraction and cross-adaptive analysis stages of the subgroups 108 (see also FIGS. 3-5), and returning the data for storage in, for example, the offline analyzer or in a block or module accessible to the offline analyzer.

[0090] The offline analyzer may also communicate with the source control block 106, which in turn, communicates with the subgroups 108, in order to set parameters of the mix at the appropriate times.

[0091] An offline analysis example will now be described. In this example, a set of multi-track audio files (also known as stems) are made available to the engine 104. The stems are analyzed frame by frame, and audio features (such as Loudness, Spectral Centroid, Crest Factor) are extracted, with values for each stored as data points against time. An analysis stage is then run to monitor variations in feature values, within individual tracks and across all tracks, and to adjust the engine 104 accordingly. For example, with loudness as the chosen extracted feature, the offline analyzer may notice that all tracks suddenly become significantly less loud and one track, e.g. an electric guitar, continues at its original level. This is maintained for a period of time (e.g., 20 seconds) before the tracks all return to their original loudness state. This is interpreted by the offline analyzer 98 as a solo section, and would affect the engine 104 in a number of ways: i) the guitar is selected as a lead track and is panned to the centre of the mix, ii) the guitar fader level is boosted (e.g., by 3dB), and iii) the smoothing function of the guitar fader is bypassed at the start of this section to allow the fader to jump and give the guitar immediate prominence in the mix. These parameter changes are stored as data points against time by the offline analyzer.

[0092] Next, the mix can be processed, following the usual signal processing algorithms present in the real time implementation, but with various parameters changed at the points in time corresponding with events discovered in the analysis stage.

[0093] It can be appreciated that there are numerous other examples and possibilities that offline analysis, and the knowledge of future audio events that we gain as a result, would have on the engine 104. For example, the overall output frequency spectrum, which selected algorithms can optimize using equalization filters to push it towards a target. The frequency content of the individual tracks, or the final mix-down, can be monitored frame by frame. The filters can then be pre-emptively controlled to adjust to changes in the spectrum that are about to occur, rather than reacting afterwards. The same theory applies for any of the tools: compression, pan, EQ, faders – they can be made to react before the event.

[0094] It can also be appreciated that the above-noted principles concerning the offline analyzer can be achieved in quasi-real-time using a look-ahead buffer, which allows pre-emptive knowledge of upcoming events without requiring the full audio files to be available.

[0095] Although a particular example configuration for the production system 10 is shown in FIG. 2, it can be appreciated that various system configurations can be achieved

using the principles described above, e.g. by adapting the structure in FIG. 5 (see below) in multiple flexible ways to create processors 122-128 (e.g. faders, compression, etc.) and subgroup 108 placements that adapt to a particular application. For example, the stages shown in FIG. 3 can be reconfigured to be in different orders, quantities and routing. As such, it can be appreciated that the examples shown herein are illustrative only.

[0096] When combined, the production system 10 continuously adapts to produce a balanced mix, with the intent to maximize panning as far as possible up to the limits determined by each track's spectral centroid. All parameters, including the final pan controls are passed through EMA filters to ensure that they vary smoothly. Lead track(s), typically vocals, can be selected to bypass the panning algorithm and be fixed in the centre of the mix.

[0097] FIG. 3 illustrates an example of a configuration for a multi-track subgroup 108 which performs the processing and mixing as a series operation for autonomous, real-time, low latency multi-track audio production. Each track 102 is received by the multi-track subgroup 108 and firstly undergoes loudness processing in a loudness processing module that includes a loudness processor 122 for each individual track, and performs the actual processing of the loudness characteristics of the associated track.

[0098] The tracks 102 are then processed by respective compression processors 124 associated with each track, and then by respective equalization (EQ) processors 126 to apply a sequence of filters to alter the frequency content of a track. The processed audio signals corresponding to each of the tracks 102 are then processed by respective left and right stereo panning processors 128a/128b. The left and right signals are then combined at 130 and 132 respectively and are processed by a mastering module 134 to be output at 138 by the subgroup 108 and eventually the production system 10.

[0099] A generic illustration of a processor 122, 124, 126, 128 used in the production engine 104 is shown in FIG. 4, which is arranged to automatically produce mixed audio content 102' from multi-track audio input content 102. The processor 122, 124, 126, 128 shown in FIG. 4 is arranged to perform the automated audio mixing by carrying out the following steps:

[00100] Receive input signals 102: digital audio signals 102 from multiple tracks are received at an input of the production system 10 and routed to multiple parallel signal processing channels of the production system 10;

[00101] Feature extraction 150: each of the digital audio signals 102 is analysed and specific features of each of the digital audio signals are extracted;

[00102] Feature Analysis (cross-adaptive feature processing 154): the extracted features and the relationship between extracted features of different signals are analysed and, in accordance with one or more processing control rules 158, the processing required for each track is determined;

[00103] Signal Processing 156: The audio signals are then processed in accordance with the feature analysis; and

[00104] Output processed signals 102': the processed signals 102' are then output as modified digital audio signals corresponding to each track.

[00105] The automated mixing process, including each of the above-mentioned steps, shall now be described in greater detail making reference to the figures.

[00106] An input of the processor 122, 124, 126, 128 is arranged to receive a plurality of stereo digital audio signals 102, in the example shown in FIG. 4, first, second, and third stereo audio signals. Each stereo audio signal 102 corresponds to an audio track to be processed, and has a left channel and a right channel. The input of the processor 122, 124, 126, 128 receives each track as a separate audio signal 102. The processor 122, 124, 126, 128 is arranged to accept any number of input audio tracks; the number of tracks only being limited by the processing capability of the production system 10 and the requirements of the audio to be output.

[00107] It can be appreciated that, as noted above, the production system 10 may also use sub-grouping 108 to achieve an optimal mix of the audio signals 102, as shown in FIGS. 4 and 5, as herein described. Individual groups of tracks can be assigned to sub-groups 108, inside which mixing and mastering processors can be placed. Sub-groups 108 can be linked together so that the mix-down or individual tracks from one subgroup 108 act as an input to another (as shown in FIG. 2). Pre-sets can be used to apply specific settings to sub-groups 108, e.g., for genre-specific or instrument-specific mixes.

[00108] In the example shown in FIG. 4, the received audio signals 102 are processed in real-time. Such real-time processing is particularly useful when the received signals 102 are real-time signals recorded live or deriving from streamed content. In such an example, feature extraction 150 is performed on the streaming audio in real-time as the audio is received. The features of the audio to be extracted includes features or characteristics of the audio signal such as gain loudness, loudness range, spectral masking, spatial masking, spectral balance, spatial balance, and others.

[00109] The received audio signals are passed into a parallel processing operation or "side-chain", i.e. using the cross-adaptive feature processing module 154 for the extraction and analysis of audio features. A plurality of feature extraction modules 150 provides such parallel feature extraction as shown in FIG. 4.

[00110] Instantaneous feature values are extracted by the feature extraction modules 150 on a sample-by-sample or frame-by-frame basis, depending on implementation. In the latter case, frame size is as low as required to ensure real-time operation with minimal latency. Accumulative averaging is applied to features to implement real-time feature estimation, the rate of which adjusts according to frame size and sample rate, which is carried out closely following the latest update of the feature value.

[00111] The extracted stream of data indicative of the certain features of an audio signal is smoothed over time using an exponential moving average filter with associated time attack and release constants, as shown by equation 1 below:

[00112] $$F_m(n+1) = (1-\alpha)F'_m(n+1) + \alpha F_m(n)$$ (Equation 1)

[00113] In Equation 1, F'_m represents an instantaneous estimation of a feature from the m^{th} track, F represents the smoothed feature estimation, n is the current sample being processed, and α is a constant between 0 and 1 that determines the weighting of recent samples in the smoothed feature estimation. Alpha values adjust according to frame size/sample rate ratio to ensure a non-varying filter response.

[00114] The cross-adaptive multi-track feature processing module 154, shown in FIG. 4, receives each of the features extracted by each of the feature extraction modules 150. The cross-adaptive processing module 154 determines processing control functions which dictate the processing operations to be applied to each of the tracks 102. The processing control functions are also determined based on pre-determined constraints 152 or rules 158, along with the extracted features. The predetermined constraints may be set by a user prior to starting the mixing process and stored in a constraints module 152. The processing rules 158 may set certain required relationships between tracks, or upper/lower limits for specific features. Constraints include, but are not limited to, the following:

[00115] For autonomous multi-track faders, all active sources tend towards equal perceived loudness;

[00116] For autonomous multi-track stereo positioning, all tracks are positioned such that spatial and spectral balance is maintained;

[00117] For autonomous multi-track dynamic range compression, compressors are applied on each track such that variation in loudness range of active sources is minimised;

[00118] For autonomous multi-track equalization, filters are applied on each track such that spectral bandwidth of sources does not overlap; and

[00119] For autonomous delay and polarity correction, delays can be added to each track to synchronize each track to a common reference.

[00120] The cross-adaptive feature processing module 154 includes a feedback operation to ensure convergence towards the desired features in the output. That is, the controls produced by the cross-adaptive feature processing block may be analysed before they are applied. If they fail to produce the desired result within a given tolerance, then the control values are adjusted before they are applied.

[00121] The processing control functions take the form of time varying filters, such as gains, delays, and infinite impulse response filters. More specifically, a control vector may be utilized, which is a weighted sum of previous control vectors and a function of the extracted features. In the case of loudness faders, multi-track processing is used to derive a decibel level control for each track. The result of this processing is then converted back to the linear domain, and applied as a time varying gain to each track, as discussed below. Similarly, in the case of autonomous stereo positioning, multi-track processing is used to derive a panning position for each track 102, which is then applied as two gains, producing a left and a right output for stereo positioning.

[00122] In the case of autonomous delay and polarity correction, the delays between all tracks 102 and a reference are analyzed, and an artificial delay introduced to synchronize the audio.

[00123] Once the above-mentioned control functions have been determined they are used to process each of the tracks in the parallel signal processing modules 156. Each track is then output by the respective processing block 156 as a separate audio signal 102' which has been processed in accordance with the controls determined by the cross-adaptive processing module 154. Each processed signal 102' is then combined by a summation process into a single audio output in the output module 110, 136. The output 102' can be of any suitable format, but in this example, is a stereo output 110, 136.

[00124] Typically, the main aspects of audio signals to be mixed include, without limitation: the relative loudness levels of each track on a frame-by-frame basis; the relative loudness of the audio signal over a period of time; equalization (EQ); compression,

mastering, the stereo panning of each track (for mixing of stereo audio signals), etc. Hence, the automated feature extraction and processing for these aspects of an audio signal shall now be considered in detail.

[00125] *Loudness Processor:*

[00126] FIG. 5 shows a multi-track mixing processor 122 that is configured to extract loudness and loudness range to allow for independent control of the relative loudness levels of multiple audio tracks to implement a fader as an example use case. In the example shown in FIG. 5, the feature extraction corresponds to loudness extraction and the cross adaptive processing corresponds to loudness optimization.

[00127] As shown in FIG. 5, audio signals 102 corresponding to multiple tracks have information relating to their loudness extracted by a multi-channel loudness extraction module 160 at each sample of frame. The multi-channel loudness extraction module 160 takes the perceptual loudness of all tracks into consideration when determining the associated loudness. A loudness optimization module 162 then determines the control functions to be applied to one or more of the tracks, as appropriate, in accordance with the perceptual loudness determination. The tracks to have their loudness altered are then altered by the respective processing modules 166, e.g., by having a gain applied to increase or decrease a signal level according to control signals 164. The output 102' therefore has been processed for loudness correction.

[00128] As discussed below, the production system 10 is configured to include a modification of the standard loudness model, ITU-R BS.1770-2, to enhance its use in the production system 10. Filter coefficients are mapped to continuous time values, and then remapped to a sample rate of the multi-track audio. This allows the loudness measure to work with any sample rate. While the loudness is extracted at each sample or frame in accordance with the EBU R 128 standard, the standard extraction technique is customised for smooth real-time loudness estimation. An overlapping window approach, as in EBU R 128 and in the gating of ITU-R BS.1770-2, is not used for estimation of loudness. Instead, the exponential moving average filter of Equation 1 is applied, with a time constant set equal to the 3000ms window length specified in EBU R 128 for gating, i.e., detection of an active signal, and for short-term loudness estimation.

[00129] During the feature extraction 160, the fader processor 154 is arranged to distinguish between silences and wanted audio and is also able to accommodate for silent portions within the audio, with a binary state of activity for each track determined by the current loudness value immediately after its calculation. The silence threshold can be

estimated based on relative levels in the audio stream. Adaptive thresholds are used to gate out any periods of silence or background noise. In particular, two independent gates are used for silence detection, a gate for determining when silence begins, set at -30LUFS (Loudness Units Full Scale) and -25LUFS for determining when silence ends. Use of two gates in this way prevents random noise resulting in fluctuating loudness estimation, and hence prevents over adjustment of the control vector in the processing stage. This silence processing is carried out as part of the feature extraction block 102, and included in loudness estimation to ensure that it does not interfere with signal flow through the system. Thus, computationally intensive operations can be performed without any interference with the real-time signal flow.

[00130] It can be appreciated that the example configurations shown in FIGS. 2 to 5 are for illustrative purposes only and that various other configurations can be used to adapt to different applications and scenarios.

[00131] *Multi-track Equalization Processor:*

[00132] FIG. 6 shows a multi-track mixing processor 126 that is configured to extract frequency characteristics of the audio tracks to allow for cross-adaptive control of the equalizers of multiple audio tracks. In the example shown in FIG. 6, the feature extraction at 170 corresponds to extraction of information from the signal used to perform an equalization, and the cross-adaptive processing corresponds to at least one equalization optimization at 172. It can be appreciated that, similar to FIG. 5, the optimization stage 172 generates controls 174 that are used by respective processors to modify the respective track at 176 and generate the output 102'.

[00133] Further detail of the equalization optimization stage 172 is shown in FIG. 7. In the example shown in FIG. 7, the optimization stage 172 includes a de-masker 180 to account for masking effects between tracks, a de-muddier 182 to eliminate low frequency components for at least some tracks, a de-noiser 184 to eliminate noise during the equalization stage, and a de-esser 186 to remove harsh sounds such as "s" and "f" sounds.

[00134] The de-masker 180 can be operated by the equalization processor 126 to correct for "masking" effects wherein a spectral region of one track masks a spectral region of another track (e.g., track A masks track B around 500 Hz while track B masks track A around 2000 Hz), and/or otherwise interferes with another signal, e.g., where a particular dB spread is desired between the tracks. As such, it can be appreciated that the de-masker 180 can be operated to compensate, adjust, eliminate, or otherwise control spectral masking effects in particular frequency regions, when comparing one track to another. As discussed

in greater detail below, by dividing the frequency spectrum into bands, and comparing corresponding bands in multiple signals (tracks), equalization effects can be applied to address masking requirements or rules which are relevant to the particular audio content, e.g. according to genres, styles, preferences, venues, etc. For example, in a particular venue or for a particular song, the relative level of the vocals in a particular frequency region may need to be consistently higher than a guitar in that frequency region, to ensure the vocals are not drowned out or "masked". It has been found that spectral processing provides an improved effect when compared to simply changing the relative levels of the tracks. It can be appreciated that by performing the spectral processing as herein described, specific frequency regions can be emphasized while leaving others alone (or even decreasing other regions at the same time).

[00135] FIG. 8 illustrates an example frequency plot 190 of exemplary frequency signals for a vocals track 192 and a guitar track 194. In this example, in a particular region of the spectrum, a difference or gap 196 occurs between the levels of the guitar and vocals, wherein the guitar is louder than the vocals. This example illustrates a scenario wherein if the vocals are meant to be as loud or louder than the guitar in that frequency band, the difference 196 shown in FIG. 8 would cause an adverse masking effect and can be compensated for using the de-masker 180. For example, the vocals track 192 could be increased according to the detected difference 196 in order to apply a de-masking effect such as an equalization of levels or to have the vocals louder than the guitar in that frequency region.

[00136] As indicated above, the de-masking effects may also be used to maintain a particular difference between tracks. For example, the rule being applied may indicate that vocals should maintain a 5 dB difference 196 with the guitar so as to allow the vocals to dominate over the guitar. If the difference 196 goes below 5dB with a particular frequency band, the de-masker 180 can be used to maintain the difference 196, even if the vocal levels are relatively higher than the guitar through the entire spectrum, as illustrated in FIG. 9.

[00137] In FIG. 9, although the vocal track 192 is relatively higher than the guitar track 194 throughout the spectrum 190, if a particular difference 196 is desired, the one indicated in FIG. 9 would need to be compensated for in the appropriate frequency band. As shown in FIGS. 10A and 10B, by dividing the spectrum into a predetermined number of bands 200 (e.g. 10 as shown in FIGS. 10A and 10B), filters can be applied to specific frequency ranges, as identified by an analysis of the bands 200 and cross-adaptive processing performed, e.g. to compensate for an undesirable masking effect.

[00138] It can be appreciated that a predetermined number of filters may be used which is less than the total number of bands in the frequency spectrum in order to conserve power and reduce processing burdens. Also, although 10 bands are shown in this example, this is purely for illustrative purposes as more or fewer bands may be used. Furthermore, it can be appreciated that the analysis described herein can be performed in either the time domain or the frequency domain. In such a configuration, various rules or criteria may be used to determine how many filters are used and to which bands they are applied. For example, the bands may be given particular priorities based on the type of instrument wherein filtering is performed for only the highest priority bands (e.g. three) with the other bands being left alone.

[00139] In this example, a rule may dictate that frequency bands f_4, f_5, and f_6 are priorities or have been identified as the only bands which need filtering. A delta dB can be determined, e.g. as shown in FIG. 11 for f_5, and the delta used to determine what de-masking effect is required. In FIG. 11, an average vocals level 204 is determined (e.g., by applying an exponential moving average of the dB level of the band, for each instrument, updated over time) and compared to an average guitar level 206 to determine the spread 196 within band f_5. In one example, if a 5 dB spread should be maintained, and the delta is 2 dB, a de-masking of the guitar track 194 can be implemented by reducing the guitar level 206 in band f_5 by -3dB. Alternatively, +3dB could be applied to the vocals level 204 to achieve the same spread.

[00140] FIG. 12 illustrates example operations that may be performed by the de-masker 180 to de-mask a track. It can be appreciated that the diagram in FIG. 12 is based on operations performed on one track for ease of illustration and this diagram may be iterated and/or performed multiple times simultaneously in order to determine any relative masking between multiple combinations of tracks.

[00141] At 210, the multi-track audio input is obtained and the tracks being analyzed determined therefrom. The frequency spectrum of the respective signals are determined at 212 and the spectrum divided into bands at 214, while a track-type recognition is performed at 216 in order to determine which tracks are being compared. Based on the track-type recognition, an appropriate de-masking rule or rule set is determined at 218. For example, if vocals and guitar are being compared, a rule set dictated a relative spread in levels may be accessed to instruct offsets to be applied when filtering these tracks. The respective bands in the divided spectrum are compared in the subject tracks at 210 and the rule applied to determine at 220 if a de-masking operation should be performed. If not, the process ends at 222. If, for that band, for that track comparison, a de-masking operation should take place,

the necessary offset is determined at 224 and applied at 226 in order to equalize the tracks according to the appropriate rule.

[00142] A process for determining EQ filter values will now be described. A priority value is determined for each band of each track, depending on masking level, instrument specific rules for masking separation, and the rank of the band (the relative importance of the band for the particular track, found by listing each band in order of magnitude). The eventual priority value is the maximum found after all track comparisons, with the highest possible value equal to one. The eventual masking separation value is the value associated with this highest priority event.

[00143] All band masking values are weighted by dividing by their priority values, and then organised high to low. If a problematic band is within the range covered by the demuddier, i.e below its cut-off frequency, this is discounted.

[00144] If there are adjacent bands in this list, these are combined and the gain, q-factor and centre frequency values adjusted accordingly (see details below). For example, a list of bands for a particular track, from high to low priority, is {3,4,1,7,8,9,0,6,5,2}. 3 & 4 are grouped, as are 7, 8 & 9. The top 4 bands are then determined, to be the ones targeted by the 4 available parametric EQ filters: 3/4, 1, 7/8/9, and 0. Filter parameters are then adjusted accordingly: the centre frequency is the mean of the centre frequencies for the grouped bands, the gain is the interpolated centre gain of the grouped bands (with gain derived from the masking separation in dB), and the Q factor is proportional to the number of bands, so QFactor = QFactorMaster / numOfBands. So in the given example, Filter 1 (3/4) has a centre frequency and gain based on an interpolation of band 3.5, and a Qfactor divided by 2, Filter 2 uses the centre frequency and gain of band 1, with a QFactor = QFactorMaster, and so on. QFactorMaster is determined depending on the number of bands used for analysis.

[00145] Turning now to FIG. 13, example operations that may be performed by the de-muddier 182 are illustrated. As indicated above, the de-muddier 182 may be used to remove low frequency components from particular tracks, particular tracks that are predetermined to not require low frequency content. For example, in a rock set-up, it may be determined that only the bass guitar track and floor (kick) drum track require very low frequencies to be heard, and other tracks such as guitar and vocals can be filtered accordingly. It has been found that during the equalization processing, a rule related to low frequency content can be relied upon to perform high-pass filtering to remove low frequency components from at least one track. At 230, the track to be filtered is obtained and the

frequency spectrum for that track determined at 232. A track-type recognition may also be performed at 234 to enable a corresponding cut-off frequency (if any) to be determined at 236. The de-muddier 182 then determines at 238 whether or not a high pass filter should be applied, and what the cut-off frequency should be. If no filtering is to be applied, the de-muddier determines if additional tracks are to be analyzed at 242. If filtering is to be applied for that track, the appropriate high-pass filter is applied at 240 and once it is determined at 242 that no more tracks are to be analyzed, the process ends at 244.

[00146] FIG. 14 illustrates example operations that may be performed by the de-noiser 184. At 250, the track being analyzed is obtained and the de-noiser 184 determines at 252 whether or not the track includes noise. If not, the process ends at 254. If the track includes noise, in addition to equalizing the tracks using the de-masker 180 and de-muddier 182, the de-noiser 184 may be applied at 256 to reduce the level for that track to minimize the existence of such noise. The de-noiser 184 uses band analysis in the frequency domain, and an expansion technique to remove regular low level noise from the signal. A WOLA filterbank and short-time Fast Fourier Transform algorithm converts the signal into a sequence of band magnitudes in the frequency domain. Magnitude values for each band are passed through a noise gate controlled by hysteresis thresholds, and gain is applied in the frequency domain as a part of the expansion technique to bands that are determined to contain noise. The de-noised signal is then converted back into time domain samples using an inverse Fast Fourier transform.

[00147] The noise suppression stage calculates gains for each frequency band depending on the power in each band and applies these frequency band gains to the according bins.

[00148] The implemented version uses a hysteresis curve to switch between no suppression (gain = 1) and suppression (gain < 1). For high input levels no gain changes are applied (gain = 1), as the input power falls below a “Low” threshold a gain < 1 is applied. In between thresholds, the suppression on/off state remains at its current value. This hysteresis helps to prevent multiple switching if the input power is near the threshold.

[00149] To smoothen the achieved gain to prevent too abrupt gain changes, an attack-release is implemented. The attack time should be short, approx. 5-10 ms, while the release time should be reasonable long, e.g. 100-500 ms.In the current vst-implementation the attack- and release-times are set for all bands equal and the power thresholds “High” and “Low” and the gain of the suppression can be set bandwise.

[00150] After the gain is calculated for each frequency band, the complex output bins from the WOLA analysis are multiplied with these gains according to the bin-to-band rule from the power estimation stage. These gain-weighted bins are the input of the WOLA synthesis stage.

[00151] FIG. 15 illustrates example operations that may be performed by the de-esser 186. At 260, the track being analyzed is obtained and the de-esser 186 determines at 262 whether or not the track includes harsh sounds such as “s” and “f” sounds. If not, the process ends at 264. If the track includes harsh sounds, in addition to equalizing the tracks using the de-masker 180 and de-muddier 182 and reducing noise, the de-esser 186 may be applied at 266 to remove the harsh “s” and “f” sounds to minimize the undesirable effects from such sounds. A frequency analysis, in either the time domain (band pass filter), or frequency domain (FFT), can be used to extract and analyse the frequencies associated with the harsh vocal sounds. A combination of side-chained compression or equalisation as shown in FIG. 15B, in either the time or frequency domain, reduces these sounds at the instant they occur, as the detected noise value goes over a certain threshold. The de-esser has the option of only being enabled for vocal tracks, and for the band pass filter to be tailored for the vocal range, or microphone used, or other property of the signal chain – e.g a female vocal would typically have a higher range. It can be appreciated that the de-esser 186 may be utilized automatically, i.e. in addition to or rather than based on knowledge of which frequencies correspond with the harsh sounds (e.g. informed by semantic data). For example, detection of certain frequencies is informative of the existence of sibilant sounds, regardless of semantic information. It can also be appreciated that the configuration shown in FIG. 15B may also include a gain or voltage controlled amplifier rather than the parametric equalization.

[00152] *Master Equalization Processor:*

[00153] It has been recognized that following a chain or sub-chain, a mastering equalization process may be desirable to achieve a certain target spectrum for a particular track or tracks. In order to perform such a master equalization, an ideal curve representative of the target spectrum can be compared to the actual frequency spectrum detected, and appropriate equalization operations applied to the tracks to bring the respective spectrum closer to the target. It can be appreciated that the target spectrum can be determined in various ways, as will be discussed below. For example, popular songs may be analyzed in order to extract ideal frequency spectrums required to achieve a certain “sound”. The following also describes a method for converting the frequency spectrum of a particular song into a target profile indicating levels within certain frequency bands that should be achieved.

[00154] FIG. 16 shows a multi-track mixing processor 134 that is configured to extract frequency characteristics of the audio tracks to allow for a mastering equalization of multiple audio tracks. In the example shown in FIG. 16, the feature extraction at 270 corresponds to extraction of information from the signal used to perform an equalization, and the multi-track processing corresponds to at least one equalization optimization at 272. It can be appreciated that, similar to FIGS. 5 and 6, the optimization stage 272 generates controls 274 that are used by respective processors to modify the respective track at 276 and generate the output 102'. Also shown in FIG. 16 is a target profiles database 278, which may be accessed by the master equalization processor 134 to ascertain target levels to be achieved in certain frequency bands.

[00155] Similar to the multi-track equalization discussed above, master equalization may utilize a predetermined number of frequency bands to compare spectral characteristics of a track relative to a target, as shown in FIG. 17. In FIG. 17, a plot 280 is shown which includes a signal 282 for an arbitrary "Track X" and a target signal 284. It can be seen from FIG. 17, that Track X 282 differs substantially from the Target 284 through frequency bands f_1 to f_5 where the signals then become more similar.

[00156] FIG. 18 illustrates an average levels plot 288a including average levels for each of the frequency bands 286 for Track X 282, and FIG. 19 illustrates the average levels plot 288b including average levels for each of the frequency bands 286 for the Target 284. In this example, the levels are normalized such that a relative measure or "difference" between corresponding frequency bands can be determined. For example, in band f_1, the difference determined by comparing the levels in plots 288a and 288b may be 5 dB (2dB – (-3dB)) as shown in FIGS. 18 and 19, in which case, a 5dB offset will need to be applied to Track X 282 to become more like the Target 284. The levels in each band 286 for the target 284 may be obtained from a predetermined target profile in the profiles database 278.

[00157] It can be appreciated that normalization in this context (and elsewhere in this document) is the process of subtracting or dividing by a certain constant. This has different uses. It eases the relative comparison of different levels, for example the level of bands in the mastering equalizer, by converting to a scale between 1 and 0. This enables efficient and simple conversion to gain values. Another usage is in before-after comparison. With absolute measures it is sometimes hard to compare two tracks, as they may e.g. have the exact same relative energies per band, but one song has them 6 dB louder – normalizing (making sure the average of the band energies is e.g. 0dB by subtracting/adding a constant) would avoid this and ensure the level is the same for both tracks. In the master equalizer 134 case, the main conceptual advantage of normalizing is that the loudness before and

after is almost the same, since you're adding as much as you're subtracting (i.e. +3 dB in band 1, -3dB in band 2, instead of 6 dB in band 1 and 0 dB in band 2, which would make the signal louder). Another, practical advantage is that the total amount of filtering should be minimized, as the filtering is not perfect (adding 6 dB to a certain band increases the centre frequency by 6 dB, but other frequencies in this band are decreased a little less; it is conceivable that e.g. if all filters have a 3 dB gain, the spectrum will be less flat due to processing artefacts that wouldn't be there if all gains were 0 dB

[00158] FIG. 20 illustrates example operations that may be performed by the master equalization processor 134 to achieve a target spectrum. In this example, it is assumed that the mixed audio signal is being mastered on a frame-by-frame basis. At 300 the frame of the track being analyzed is obtained and the processor 134 determines at 302 the frequency spectrum for the signal. The processor 134 also determines at 304 the target to be compared to, e.g., by accessing the target profiles 278 and determining the target spectrum at 306. The frequency spectrum is divided into the predetermined bands at 308 and a representative value or "level" is calculated in each band as illustrated in FIG. 18 (e.g. as an average over the band as described below and shown in FIG. 26). Once the levels are determined at 310, the levels are compared to corresponding levels for a target spectrum at 312. Based on this comparison, one or more filters are applied at 314. The processor 134 may then determine at 316 if more frames are to be processed, in which case the process repeats from 300. Otherwise, the process ends at 318.

[00159] It can be appreciated that the operations performed at 314 may include a gain compensation procedure to account for offsets in one frequency band "bleeding" into a neighbouring band. FIG. 21 illustrates such an effect wherein for a particular band 286, if a +Y dB offset is to be applied to a track 282 in that band 286, the resultant effect would bleed into at least adjacent bands since the curve 320 does not typically approximate a "brick wall" filter that only affects the particular band 286. It can be appreciated that if one wants to approximate such a brick wall filter, that only affects a certain frequency region, by a certain amount, and the rest by another certain amount (e.g. 0 dB), with no transition between regions, various difficulties can be introduced, which may have poor side effects on the signal. The gain compensation approach described herein applies an iterative process that overcomes these drawbacks.

[00160] As shown in FIG. 21, an array 322 of offsets in this example may be applied assuming that only frequency band f_x requires an offset of Y. However, the corresponding effect show by the curve 320 adversely affects the bands in which a zero offset is required. Consequently, as shown in the example illustrated in FIG. 22, before outputting the

mastered track, the processor 134 can iterate through a gain compensation routine described below in which a first array 322a based on an initial analysis is applied to generate a first effect 324, in which further iterations are required due to the bleeding effect shown in first curve 320a. A second array 322b may then be generated in accordance with the gain compensation algorithm to perform additional offsets in neighbouring bands. This process may be repeated for additional iterations to generate a third array 322c and additional arrays if necessary. In the example shown in FIG. 22, an original offset of +6 to only band f_x is tweaked during subsequent iterations as discussed below and shown in FIG. 25, It can be appreciated that the gain compensation algorithm described below, although shown in the context of master equalization, should not be limited to such an application. For example, this technique could be applied to standard, static equalizers or any other application using similar filters, even outside of audio applications.

[00161] An example of a gain compensation algorithm will now be described, making reference to FIG. 25.

[00162] As mentioned above, when applying a series of peaking filters and/or shelving filters to increase/decrease the signal's energy by a certain amount at and around certain frequencies, the overlap/crossover of the filters causes different effective gains to be applied at the respective frequencies. For example, when applying a 6 dB gain at 1000 Hz and a - 3 dB gain at 1200 Hz, the effective gains will be slightly less than 6 dB (e.g. 5.5 dB) and slightly less than 3 dB (e.g. 2.5 dB), depending on the width of the respective filters.

[00163] The process illustrated in FIG. 25 calculates the gains to be applied in order to decrease this error and have effective gains as close as possible to the desired gains. In the following, the gains desired at the corresponding frequencies are called 'target gains', or G_t; the gains that are measured are called 'analysis gains', or G_a; and the gains that are returned by the algorithm and should instead be applied by the equaliser are called 'compensated gains', or G_c.

[00164] For peaking filters, the frequencies at which the filters are applied (the so-called 'centre frequencies', f_c) are the same frequencies at which the total filter response is analysed (the analysis frequencies or f_a). However, in the case of shelving filters, the characteristic frequency is the cutoff frequency (here also denoted as f_c as their function is similar in this context), but the gain applied is the gain that should be effectively added at frequencies beyond this point.

[00165] For this reason, in the case of shelving filters, the analysis points should be closer to the extremes of the spectrum (low and high frequencies) than the cutoff frequency

points. As an example, consider the case where a high shelving filter with 3 dB gain is applied: (well) beyond the cutoff frequency, the signal will be 3 dB louder, but at the cutoff frequency the effective gain is less (depending on the filter type). See further for an example of a system where the lowest and highest filter are shelving filters.

[00166] Consider now a system with a number of filters with fixed frequencies f_c (centre frequency for peaking filters, cutoff frequency for shelving filters) and quality factors Q (possibly but not necessarily the same for each filter). Alternatively, the filters could have a number of different parameters and be of virtually any type, but we assume for simplicity that the filter's parameters and types are not changing, whatever they may be, after they have been set to perform a certain spectral shaping. From this point onwards, the only filter parameter the algorithm manipulates is the applied filter gain G_c.

[00167] If the highest and/or lowest filters are shelving filters, the analysis frequencies can be calculated at 330 as follows to avoid the problem mentioned above:

[00168] If there is a low shelving filter rather than a peaking filter for the lowest frequency:

$$f_a(1) = \frac{f_{c1}^2}{f_{c2}} \qquad (1)$$

[00169] If there is a high shelving filter rather than a peaking filter for the highest frequency:

$$f_a(N) = \frac{f_{cN}^2}{f_{c(N-1)}} \qquad (2)$$

[00170] where N is the number of filters.

[00171] As a first step, the compensated filter gains (the gains that have to be applied in order to approach the desired gains at the analysis points) are set equal to the desired filter gains at 332, i.e.:

$$G_c = G_t \qquad (3)$$

[00172] From this point onwards, the compensated filter gains G_c are corrected during one or more iterations of the following.

[00173] The total filter response of the cascade of the N filters is calculated at 334, given the parameters above and the desired filter gains. Then, the gain this filter applies at the analysis frequencies is calculated at 336 and stored as G_a (the analysis gains). Based on these measured gains, the new compensated filter gains are calculated at 338 as follows:

$$G_c = G_c - (G_a - G_t) \quad (1)$$

[00174] In other words, the errors ($G_a - G_t$) (the difference between the desired gains and the measured gains at the analysis frequencies) are subtracted from the compensated filter gains. At this point, the system determines at 340 whether or not to iterate. Either the compensated gains are applied at 342, or there is another iteration (calculate filter response with newly calculated compensated gains G_c, analyse actual gains at analysis frequencies, and update compensated gains G_c again).

[00175] The algorithm could iterate until the error is smaller than a certain value, or it could iterate a fixed number of times. How fast the algorithm converges depends on the applied gains, the width and type of the filters, and the centre/analysis frequencies.

[00176] FIGS. 23A and 23B show a possible result of the algorithm. The example shown in FIG. 23A illustrates a low shelving filter, 8 peaking filters, high shelving filters, with a desired 6 dB boost at 1 kHz and no cut/boost at other frequencies. The example shown in FIG. 23B illustrates a low shelving filter, 8 peaking filters, high shelving filters, with a more complex desired EQ curve.

[00177] FIGS. 23A and 23B illustrate gain compensation algorithm at work, wherein these figures show the curve a standard equaliser could apply with the gains of the different filters shown as dots on the curve - note that the actual gain the equaliser applies is significantly different due to overlap. They also show the calculated compensated gains G_c after 5 iterations of the above algorithm, and the resulting response curve of the equaliser when these gains are applied. Note that in this case, the curve shows an almost exactly right amount of gain applied at the respective frequencies. The phase plots suggest the algorithm introduces little to no real phase problems. The different curves or points can be identified as follows. The compensated gain curve (the frequency-dependent gain that would be applied using the algorithm) goes through the points that show the target values (the values one wishes to apply at the analysis frequencies). The other points are the compensated gains (they are often slightly higher or lower than the desired gain at the corresponding frequency, because the algorithm compensates for overlap); the other curve is the frequency-dependent gain the equalizer (or filter array of any kind) would apply without the

compensation algorithm. It should be noted that the compensated curve almost flawlessly applies the desired target gains at the relevant frequencies, whereas the uncompensated curve is often quite a bit off. It may be noted that for the shelving filters, the target gains are plotted at the analysis frequencies (i.e. a bit further upwards for high shelving or a bit lower for low shelving), as the desired gain, e.g. +3 dB, is not fully applied around the cutoff frequency f_c yet, but a bit higher than this (and all the way down to zero or down to infinity, in principle). However, the compensated gains (the gains one actually applies after running the algorithm) are shown where they are actually applied, i.e. at the cutoff frequency for the shelving filters.

[00178] *Automatic centre frequency and Q:*

[00179] In any equaliser design where a constant relative width or a constant Q (see formulas below) are desired, along with fixed centre frequencies that are at equal geometric distance from each other (i.e. equal on a log scale, e.g. 1000 Hz, 2000 Hz, 4000 Hz, 8000 Hz), the following method can be used to determine the centre frequencies and Q factor. Using this method, only a lower and upper frequency limit (for audio this could e.g. be f_l =20 Hz and f_u =20 kHz) and a number of bands N are required. Background formulas:

$$f_l = f_c\left(\sqrt{1+\frac{1}{4Q^2}}-\frac{1}{2Q}\right) \tag{5a}$$

$$f_u = f_c\left(\sqrt{1+\frac{1}{4Q^2}}+\frac{1}{2Q}\right) \tag{5b}$$

$$Q = \frac{f_c}{f_u - f_l} = \frac{f_c}{\Delta f} \tag{5c}$$

$$\text{(5a) \& (5b)}: f_l \cdot f_u = f_c^2\left(\sqrt{1+\frac{1}{4Q^2}}-\frac{1}{2Q}\right)\left(\sqrt{1+\frac{1}{4Q^2}}+\frac{1}{2Q}\right) \tag{5d}$$

$$= f_c^2\left(1+\frac{1}{4Q^2}-\frac{1}{4Q^2}\right)$$

$$= f_c^2$$

$$\text{(5a) \& (5b)}: f_c = \left(\sqrt{1+\frac{1}{4Q^2}}+\frac{1}{2Q}-\sqrt{1+\frac{1}{4Q^2}}+\frac{1}{2Q}\right) \tag{5e}$$

$$= \frac{f_c}{Q} \tag{5f}$$

$$= \Delta f \tag{5g}$$

$$\tag{5h}$$

[00180] Equation (5e) should be true because the centre frequency is exactly in the centre of the lower and upper cutoff frequency f_l and f_u on a logarithmic scale, i.e.

$$\log f_c = \frac{\log f_u - \log f_l}{2} \tag{6a}$$

$$\Updownarrow$$

$$\log f_u - \log f_c = \log f_c - \log f_l \tag{6b}$$

$$\Updownarrow$$

$$2\log f_c = \log f_u + \log f_l \tag{6c}$$

$$\Updownarrow$$

$$f_c = \sqrt{f_u f_l} \tag{6d}$$

[00181] This means the centre frequency is the geometric mean of the lower and upper 3 dB cutoff frequency.

[00182] *Formula automatic centre frequency:*

$$f_n = f_l \cdot \left(\frac{f_u}{f_l}\right)^{\frac{n-0.5}{N}} \tag{7}$$

[00183] where f_n is the center frequency of band n ($1 \le n \le N$), f_l the lower frequency bound (here 20 Hz), f_u the upper frequency bound (here 20 kHz), and N the number of bands. The –0.5 in the exponent is there to assure the bandwidth of the outer filters (lowest and highest) extend to the lower resp. upper bound. This way, the center frequencies are equidistant from each other on a log scale, and the bands extend to the edges of the frequency range we're interested in (e.g. 20 Hz - 20 kHz) and not (much) further. This should correspond closely with the fixed frequencies for the 10 band EQ, provided the N/2 + 1th (even) or (N + 1)/2th (odd) frequency is forced to 1kHz, and the other frequencies shifted by the same amount. The same is true for the 5 band EQ, although there the middle frequency is hard-coded to be different from 1 kHz.

[00184] The quality factor (Q) has the following definition:

$$Q = \frac{\Delta f}{f} \tag{8}$$

[00185] with f the center frequency of the filter, and "f its 3 dB bandwidth. Hence, assuming the 3 dB bandwidth should extend from $f_{l,n}$ to $f_{u,n}$, with $f_{l,n}$ the geometric mean between $f_{c,n-1}$ and f_n, and $f_{u,n}$ the geometric mean between f_n and $f_{c,n+1}$, the static Q factor for every filter of an N band Master EQ becomes (with f_n as in Equation (9e) below).

[00186] *Formula for automatic Q factor:*

$$Q = \frac{f_{ci}}{f_{hi} - f_{li}} \tag{9a}$$

$$= \frac{f_{ci}}{\sqrt{f_{hi-1} f_{hi}} - \sqrt{f_{li} f_{li-1}}} \tag{9b}$$

$$= \frac{\sqrt{f_{li}}}{\sqrt{f_{hi-1}} - \sqrt{f_{li-1}}} \tag{9c}$$

$$= \frac{\sqrt{f_l \left(\frac{f_h}{f_l}^{\frac{n-i}{N}}\right)}}{\sqrt{f_l \left(\frac{f_h}{f_l}^{\frac{n-i+1}{N}}\right)} - \sqrt{f_l \left(\frac{f_h}{f_l}^{\frac{n-i-1}{N}}\right)}} \tag{9d}$$

$$= \frac{1}{\left(\frac{f_h}{f_l}\right)^{\frac{1}{2N}} - \left(\frac{f_h}{f_l}\right)^{-\frac{1}{2N}}} \tag{9e}$$

[00187] In the 5-band MasterEQ, the (hard-coded) Q factor is 1.0. In the 10-band MasterEQ, it is 1.4. The above formula yields Q = 0.67 and Q = 1.41, respectively. FIG. 24 shows the centre frequencies and the Q factors for a 5-band EQ, a 10-band EQ, a 10-band EQ with one of the center frequencies forced to 1 kHz, and a 31-band (graphic) EQ (see for example the dbx 3231L Graphic Equalizer). Note the hard-coded center frequencies are spaced almost the same amount. Conversely, the Q factor and centre frequencies (hard-coded or obtained automatically), the lower and upper cutoff frequencies are obtained as follows:

$$f_l = f_c \left(\sqrt{1 + \frac{1}{4Q^2}} - \frac{1}{2Q}\right) \tag{10a}$$

$$f_h = f_c \left(\sqrt{1 + \frac{1}{4Q^2}} + \frac{1}{2Q}\right) \tag{10b}$$

[00188] FIG. 24 illustrates centre frequencies and Q factors, wherein the lines in the middle of the black, dashed lines (lower and upper frequencies) are the centre frequencies following equation (9e). The hard-coded centre frequencies in the 5- and 10- band MasterEQ are also shown. It may be noted that the black lines show the intervals according to:

$$\left[f - \frac{\Delta f}{2}, f + \frac{\Delta f}{2}\right] = \left[f - \frac{1}{2Q}, f + \frac{1}{2Q}\right]$$

[00189] As discussed above, pre-existing songs can be used to generate target frequency spectrums to be stored as target profiles 278. In order to generate such a target profile 278, the frequency spectrum for the target track is first determined, and then the corresponding "levels" described above generated from the frequency spectrum, using an averaging process illustrated in FIG. 26.

[00190] In the context of the master equalization process discussed above, among other applications and contexts, one may be interested in the energy per band of a song (or more generally an audio file), or the energy per band of an average of a number of songs. In the latter case, these songs may share one or more characteristics, such as being songs of the same artist, genre, mixing style, or success (e.g. based on their top position in the charts). It may be noted that the frequency bands can be defined as one sees fit.

[00191] Based on these energies calculated per band (called 'target energies' in this text), the master equalizer 134 can measure the energies per band of the incoming signal, compare them to these 'target energies', and hence derive the appropriate gain to be applied per band.

[00192] The following text outlines the used algorithm to derive the target energies (although other methods may be used to this end). At first, the magnitude spectrum of each considered audio file is calculated at 344 using the Fast Fourier Transform (FFT).

[00193] If the audio x is stereo ($x = [x_L, x_R]$), it is converted to a mono file at 348 by averaging:

$$x = \frac{x_L + x_R}{2} \qquad (11)$$

[00194] Then, the FFT is calculated for each window of each mono audio file x_i at 350, where the window length is N_{win} samples and every window starts at a multiple of N_{hop} samples. Each window is first multiplied with an appropriate window (e.g. Hanning, Hamming, etc,) before calculating its FFT. The jth FFT of the ith song is then stored in Xi,j . The average magnitude spectrum is then obtained at 352 as follows:

$$X_i = \sum_{j=1}^{N} \frac{2\,|X_{i,j}|}{L/2} \qquad (12)$$

[00195] with L the FFT length and N the number of FFTs to be measured (i.e. the number of windows). The system may then determine at 354 if other songs are to be included in an average spectra at 354. If not, the magnitude spectrum computed at 352 is used to generate a target profile 278.

[00196] If more than one song is to be averaged, the magnitude spectra for the different songs are then averaged at 356, i.e. the total 'target' magnitude spectrum is $X = \frac{\sum_{i=1}^{S} X_i}{S}$ if there are S songs. The target profile 278 of the total target spectrum would then be generated at 358.

[00197] Alternatively, each song can be converted to a cumulative distribution between 0 and 1 (as a normalisation measure, and to avoid excessive variations in standard deviation and confidence intervals), and only then averaged.

[00198] With f_n the upper frequency limit of band n, the target energy values for every band are calculated as follows. All bins of magnitude spectrum X corresponding with frequencies below f_1 (the first band) are summed as value E_1 (total energy of the first band). Then, for every next band, the magnitude spectrum values corresponding with bins between f_{n-1} and f_n are summed to obtain the band's energy value E_n.

[00199] At any appropriate point in this algorithm, normalisation can take place. For example, the audio x or magnitude spectra X_i of the different songs can be normalised (multiplied by a constant gain value to achieve an equal peak level, RMS level, or loudness measure), or the resulting band energy values En can be multiplied by the same factor to achieve a peak value or an average equal to 1 (0 dB) - or any other value for that matter.

[00200] It can be appreciated that a high-resolution, FFT-style response for every preset can be stored, so that one can 'downsample' it to any number of presets using the above method. Alternatively, a lesser-resolution (e.g. 10-point) preset can be stored that can be 'upsampled' to a higher number of bands if we need a higher resolution for the master equalizer 134.

[00201] It can also be appreciated that the above-described techniques used in the multi-track and/or master equalization processes should not be limited to such applications, or audio. Such techniques may be applied to any similar filtering processes and/or filter band design.

[00202] It may be appreciated that the concepts related to multi-track equalization may be equally applied to master equalization and vice versa. For example, the comparison of

parameters and features within a plurality of frequency bands within the frequency spectrum may be programmed to be used by both processes in the overall system 10.

[00203] *Stereo Pan Positioning Processor:*

[00204] FIG. 27 shows a multi-track mixing processor 134 that is configured to extract features of the audio tracks to allow for a pan positioning optimization of multiple audio tracks. In the example shown in FIG. 27, the feature extraction at 370 corresponds to extraction of information from the signal used to perform panning operations, and the cross adaptive processing corresponds to at least one pan positioning optimization at 372. It can be appreciated that, similar to FIGS. 5 and 6, the optimization stage 372 generates controls 374 that are used by respective processors to modify the respective track at 376 and generate the output 102'.

[00205] The stereo positioning functionality of the multi-track audio processing system 10 employs the same loudness and noise gate processing used for determining active tracks so that inactive tracks bypass the processing and the track's parameters therefore remain constant. The stereo pan processing may operate as follows.

[00206] Spectral centroid information is extracted for determination of the stereo positioning of a signal. This spectral centroid defines a panning factor associated with an audio signal, set by a ratio against the maximum spectral centroid found over all channels, with the result that higher frequencies are progressively panned further from the centre of the panning space. The spectral centroid of each active track is calculated using a Fast Fourier Transform (FFT). Initially, as a track enters for the first time, a determination is made of which of the other tracks has the closest spectral centroid. Then each couple of tracks related by the closeness of their spectral centroid are then set to opposing left or right channels of the stereo output.

[00207] Hence, this operation prevents tracks from crossing over the 0.5 centre point. If this process results in a poor distribution of tracks over the left and right channels then the system will perform this operation again until an acceptable distribution is provided. As source distribution problems typically occur on the rare occasion that multiple tracks enter the mix at exactly the same time, repeating the operation after the tracks have entered ensures a better source distribution. As a fail-safe measure the program will only allow a small number of attempts before the distribution is fixed.

[00208] The system 10 is also arranged to determine a maximum spectral centroid range found over both channels of all tracks. A maximum spectral centroid parameter indicative of this range is then stored in a memory of or associated with the system 1000 and updated as

and when this range changes. The panning factors of each audio track can then be adjusted to further spread the tracks across the left and right channels so that the full panning space is utilised even when the full frequency spectrum is not. The extent of the panning width can therefore be controlled by the user or automatically in order to maximise the use of the available spatial audio width.

[00209] Spectral and spatial balance are also analysed at the output of the stereo mix in order improve the quality of the audio output.

[00210] For spectral balance, FFTs of the left and right output channels are split into five frequency bands, and the magnitude of each band's frequency bins are cumulatively summed per channel. The arctangent of the ratio between the left and right summation produces a spectral balance angle for each band. If the spectral balance angle of a mix is below 0.45 or above 0.55, then pan locations of each track are adjusted to push the spectral balance angle back towards the 0.5 centre point. The adjustment of pan locations is performed with a ratio so that tracks with spectral centroid locations closest to the centre frequency of each band are moved the furthest, and those tracks outside a certain bandwidth (determined by a fixed Q-factor of 0.3 from the centre frequency) are not moved at all.

[00211] Spatial balance is improved by analyzing the ratio of the peak magnitude of output left and right channels, and moving all sources indiscriminately by a small factor, provisionally 0.02, if the ratio of the peak magnitude of output left and right channels is outside the same allowed tolerance as described for the spectral balance above (below 0.45 or above 0.55). Typically, however, the adjustment of pan locations for spectral balance ensures the overall spatial balance is kept within the allowed tolerance.

[00212] Combined, the system 10 continuously adapts to produce a balanced mix, with the intent to maximize panning as far as possible up to the limits determined by each track's spectral centroid. All parameters, including the final pan controls are passed through EMA filters to ensure that they vary smoothly. Lead track(s), typically vocals, can be selected to bypass the panning algorithm and be fixed in the centre of the mix.

[00213] Turning now to FIGS. 28 to 35, an exemplary pan positioning process is illustrated.

[00214] In the configuration shown in FIGS. 28 to 35, a real-time system for automating stereo panning positions for a multi-track mix is presented. Real-time feature extraction of loudness and frequency content, constrained rules and cross-adaptive processing are used to emulate the decisions of a sound engineer, and pan positions are updated continuously to

provide spectral and spatial balance with changes in the active tracks. As such, the system is designed to be highly versatile and suitable for a wide number of applications, including both live sound and post-production. A real-time, multi-track C++ VST plug-in version is also shown. A detailed evaluation of the system is given, where formal listening tests compare the system against professional mixes from a variety of genres.

[00215] Stereo positioning is the process of changing the apparent location of a sound source in a binaural audio mix. Most commonly, this is achieved by feeding left and right channels with the same sound source and adjusting the relative amplitude of the channels. This is referred to as the interaural level difference (ILD), and is traditionally adjusted by the pan pots on a mixing desk.

[00216] Localisation of sound sources is also aided by temporal differences between the ear channels, known as the interaural time difference (ITD), for frequencies lower than 1.5kHz. This accounts for the extra time required for longer-wavelength sounds to reach the ear, and in sound production is achieved by introducing an appropriate delay between the channels, and additionally EQ to approximate a high-frequency roll-off due to the acoustic effects of the head. ITD delay is in the region of 1-2ms, while Haas panning (without the use of ILD) is around 20ms. Delay based stereo positioning techniques can be very effective, but equally can introduce issues when listening using loudspeakers e.g. comb-filtering and the requirement for the listener to be in a central location between the two speakers. Typically, the technique is used sparingly and only in post-production. The configuration shown in FIGS. 28 to 35 addresses the stereo placement of sources using ILD, although the use of ITD is considered and cab be built into a plug-in as an additional option.

[00217] The following presents a configuration of pan positioning processing to automate the task of panning a mix. The motivation behind the adopted approach is to determine general rules and constraints which can be adopted to emulate the performance of a sound engineer in a real-time environment. This requires the extraction of features and cross-adaptive processing to analyse all incoming tracks and reach appropriate decisions for adjusting the mixing controls.

[00218] The following example provides a wide array of improvements over the original proof-of-concept solutions, such as being intended for both live and post-production use, an arbitrary number of tracks, fully autonomous, use of spectral centroid from an FFT instead of a filter bank, better use of panning space and spectral/spatial balancing. The configuration also takes influence from real-time processing challenges. With the addition of techniques

such as vocal detection, currently in development, the proposal is for a fully autonomous panning tool with minimal or no human interaction required.

[00219] The frequency content of each track and the relationship between the tracks is used to determine panning position. However, the main objective when panning is to retain balance in the stereo domain. An important aspect of the algorithm can therefore be the steps taken to monitor different measures of balance, and to perform adjustments where necessary.

[00220] An analysis of mixing practice shows sources with higher frequency content are progressively panned further towards the extremes. A typical drum kit, for example, places the kick drum in the centre, the toms and snare close on either side, with the high frequency cymbals furthest left and right. Furthermore, high frequency sounds diffract less as they bend around the head, and so the panning effect needs to be greater to represent this. For these reasons an expanding panning width is needed to push higher frequency sources wider in the stereo field.

[00221] In addition to expanding the panning width with frequency, sources with frequency content below a certain threshold should be fixed in the centre of the mix. Having low frequency sources off centre can provide an uneven power distribution, and furthermore, due to the longer wavelength there is little or no directional information below 200Hz.

[00222] Panning techniques dictate that sources with similar spectral content be placed apart in the stereo field to minimise spectral masking. As a result the tracks can be more easily distinguished in the mix.

[00223] Spatial balance is the comparison of signal level between left and right channels and is the most important consideration when mixing, where the aim is for both to be approximately equal [6,7]. As the activity and intensity of sources can change during a song the source placement must be able to adapt to provide a balanced mix.

[00224] Spectral balance is the ratio of intensity of frequency content between left and right, so that there is an equal spread of frequencies across the mix.

[00225] Audio signals are typically divided into well-defined frequency bands: lows, low-mids, high-mids and highs, and each band should have approximately equal content in left and right channels. Where there is a single source dominating a band, typically the source will be moved towards the middle of the mix, or the source may be duplicated in the opposite channel and a stereo effect (phase, delay, reverb etc.) applied.

[00226] Stereo spread is a measure of how the whole panning space has been filled. For a full stereo image there should be an even distribution of sources to avoid gaps in the stereo field.

[00227] An additional consideration is the overall weighting on the panning width. Generally speaking hard panning of sources is unnecessary, but an overall weighting on the panning width allows the extent of the utilised panning space to be controlled.

[00228] In popular music, a lead vocal track is likely to be the focal point of a song. To provide balance and a natural listening experience it is most common for the track to be placed in the centre of the panning space. In this situation, user interaction to designate lead vocal tracks is desirable. For a fully autonomous system, however, vocal detection techniques can be used to automate the process. With the understanding that vocal detection is unlikely to be 100% accurate, the weighting on the decision needs to be in preference of false positives which may place more tracks than desirable in the centre and produce a sub-optimal mix, rather than false negatives that may place a lead vocal in a non-central position and be most likely to produce a poor mix.

[00229] Fixed pan positions are unlikely to remain optimal for the entirety of a track. Sound engineers will typically adjust pan positions over time or record automation curves in Digital Audio Workstation (DAW) software to make alterations to the mix. The algorithm therefore should continually tweak the pan positions to optimise the mix.

[00230] As previously mentioned, techniques exist other than amplitude based panning for changing the stereo image, particularly in post-production, and their use in the algorithm should be considered.

[00231] *Example Algorithm:*

[00232] A block diagram of a configuration for a pan positioning processor 128' is shown in FIG. 28.

[00233] To deal with the nature of short-frame real-time processing, the algorithm uses exponential moving average (EMA) filters extensively to provide smoothly varying data variables. The algorithm makes use of a technique for determining the current active/inactive status of each track for every frame. Whilst a simpler signal level gate could be employed in this case, the intention is for the program to make use of the original calculation when the programs are combined. The EMA filter is a 1st order IIR filter, with the following difference equation, where α is a value between 0 and 1:

$$y[n] = (1 - a) \cdot x[n] + y[n - 1] \qquad (1)$$

The value of α is adjusted according to the sample rate and frame size for a fixed filter response.

[00234] A loudness value per frame is calculated using the EBU R-128 standard, which is an energy measurement on a signal processed by two biquadratic IIR filters. Filter coefficients adapt depending on sample rate to ensure a constant frequency response. A loudness measurement is calculated per frame and processed using an exponential moving average filter, to provide a smoothly varying loudness measurement for each incoming track.

[00235] A noise gate, with two loudness thresholds at -25 and -30LUFS (Loudness Units to digital Full Scale) and a hysteresis loop, provides a binary indication for each frame of whether a track is active or not. The hysteresis loop prevents excessive switching of state when the loudness level fluctuates above and below one threshold. Feature extraction and the control of the exponential moving average smoothing filters are determined by the noise gate, including commencing smoothing when a track first becomes active.

[00236] The spectral centroid is used to determine the 'centre of mass' of a spectrum, and provides a time-varying frequency value in Hertz (Hz) for each source every frame. The spectrum is calculated with a Fast Fourier Transform. Because real signals are being analysed only the first half of the spectrum need be calculated, due to the duplication above the Nyquist frequency. Spectral centroid is calculated as:

$$SC_m = \frac{\sum_{n=0}^{N/2-1} |X_m[n]| \cdot f[n]}{\sum_{n=0}^{N/2-1} |X_m[n]|} \qquad (2)$$

Where X_m represents the discrete Fourier transform of the *mth* signal in the multi-track set, and [ ] is the frequency represented in bin *n*.

[00237] The exponential moving average of the spectral centroid SCema_m is updated only when the noise gate determines the track to be active, preventing erroneous spectral centroid values being used.

[00238] The size of the FFT should be considered to obtain a sufficient frequency resolution to detect low or close frequency content. For a real-time plug-in implementation where the incoming frame size is controlled externally, a buffer accumulates a sufficient number of samples before calculating the FFT. The buffer size is chosen by considering bin width:

$$binWidth = \frac{f_s}{N} \qquad (3)$$

where f_s is the sampling frequency in Hz and N is the FFT size. Assuming a maximum of 192kHz, a frame size of 2048 or above (providing a maximum spacing of 62.5Hz) is considered sufficient.

[00239] As a track enters the mix for the first time a decision is made as to whether the source should be dominant in the left or the right channel. This is then fixed to ensure a source cannot cross over from one channel to the other:

$$P_m[n] = 0 \; or \; 1 \qquad (4)$$

The decision made is to use the opposite polarity to the channel with the closest spectral centroid, provided it hasn't been selected as a lead channel. This way, the distance between sources with similar spectral content is maximised and spectral masking is reduced.

[00240] The mean left/right weighting of all panning positions is checked after the addition of each new track. As a safety measure, in the event of a poor distribution (determined as when the mean strays beyond a tolerance of 0.2 from the centre), the system can be configured to automatically reset.

[00241] With the dominant channel decided the degree of panning applied to each source is scaled according to its spectral centroid, the maximum spectral centroid value of all sources and the overall panning width, to produce a panning factor for each track. A custom exponential curve, shown in FIG. 39, determines the source distribution.

[00242] As a method to prevent stereo spread imbalance, the maximum spectral centroid value of all tracks is stored and updated over time. This allows the panning ratio to adapt according to the frequency range, and allows the full panning space to be utilized when the full frequency spectrum is not (shown in FIG. 30).

[00243] The panning width is a user-controlled value between 0 and 10 to extend or restrict the width of all sources, set to 5 by default for fully autonomous use. It works by moving the maximum spectral centroid value using a weighting of one third of the *SCmax* value, to adjust the angle of each track appropriately, as shown in FIG. 30. In FIG. 30, panning positions on the left graph show consistent spread for different values of maximum spectral centroid, and the right graph illustrates the effect of the panning width control.

[00244] The final panning factor is defined as:

$$Pf_m[n] = \left(\frac{\log_{10}(SC_m[n])}{\log_{10}\left(SCmax + \left((10 - PW) \times \frac{SCmax}{3}\right)\right)} \right)^4 \qquad (5)$$

where $SC_m[n]$ is the spectral centroid of each track, SCmax is the maximum spectral centroid calculated from all tracks, and PW is the panning width factor between 0 and 10. This is applied to the *P*[*m*] starting value of 0 or 1 to give a panning position centered on 0.5:

$$P_m[n] = (Pf_m \cdot (2 \cdot P_m[n] - 1) + 1)/2 \quad (6)$$

[00245] It can be appreciated that exempt from these general panning rules are designated lead tracks and tracks with a spectral centroid below the low frequency cut-off point, set nominally to 200Hz. In these cases the pan position is fixed at 0.5 in the centre of the panning space.

[00246] The processes detailed above give appropriate positions for the different sources within the mix, which will remain approximately static throughout assuming reasonably constant spectral centroid readings. As such, the mix should be reasonably balanced already, and require only optional minor tweaks to pan positions. However, balancing takes on particular importance as the dynamics of the mix change over time; when tracks drop out or come in, for example.

[00247] At all times the mix is tending to the maximum possible use of the panning space, up to the limits set by the panning factor. For this reason, balancing will only involve pulling sources inwards towards the centre and not pushing them further towards the extremes. Once balance has been achieved the mix will attempt to move the sources back to their original static positions.

[00248] The aim of the spectral balancing is to maintain the left/right balance across the entire frequency spectrum.

[00249] A 5-band approach is used, where the FFT of the left and right panned master channels are calculated, and the complex magnitude taken. Centre frequencies of 750, 1650, 3650, 7750 and 16000 Hz are used to cover the audible frequency spectrum. The spectral balance angle per band is calculated as the inverse tangent of the sum of magnitudes:

$$SpecB_b = \tan^{-1}\left(\sum_{k=K_b}^{K_{b+1}-1}|L[k]|/\sum_{k=K_b}^{K_{b+1}-1}|R[k]|\right) \quad (7)$$

where *L* and *R* are the FFT data from the left and right channels, *b* is the band between 1 and 5, *k* is the FFT bin and is the starting bin number for each band.

[00250] The aim is to converge each of the spectral balance values to 0.5, i.e. the centre of the mix. A tolerance of 0.05 is allowed, meaning balancing will only occur on a band where $0.45 < SpecB_b < 0.55$.

[00251] For each band requiring balancing, sources are ordered by their distance from the centre frequency. Only sources on the higher-weighted channel within a certain bandwidth from the band centre frequency (using a Q-factor of 0.5) are moved. The ordering of the sources affects to what extent they are moved, with the closest sources moved the most using the following factor:

$$P_m[n] = P_m[n] + \left(dir \times G_{SB} \times \left(M_A - \frac{index_m}{M_A} \right) \right) \quad (8)$$

where *MA* is the number of tracks which have become active, *index* is the source's place in the distance array and ranges from 0 (closest source) to *MA, dir* is set to either 1 or -1 to ensure the sources move in the desired inward direction, and is the movement factor and is fixed by default to 0.3.

[00252] Spatial balance is defined as the inverse tangent of the peak signal level ratio from the left and the right channels.

$$SpatB_k = \tan^{-1}\left({|y_r|}/{|y_l|} \right) \quad (9)$$

[00253] Similarly to the spectral balance, the aim is to converge at the 0.5 centre position, when $0.45 < SpatB_b < 0.55$. In that case, all active sources (with the exception of designated lead tracks or sources with a spectral centroid below the low frequency threshold) on the channel with the higher weighting are moved inwards by the same small factor.

[00254] It was shown experimentally that a by-product of spectral balancing is the balancing of the mix spatially as well, and so the process is often obsolete.

[00255] The illustrated configuration, as thus described, is for the placement of monaural (mono) sources. To this end, sources with existing stereo information can be mixed down to mono for replacement in the stereo field. However, this may not always be desirable. Stereo sources, recorded from coincident microphones or a stereo instrument like a piano, can contain useful stereo information that should be maintained. In this situation, the pan pots become a tool for weighting the mix towards left or right, known as 'balancing'.

[00256] In this implementation stereo information is maintained by default, with the width adjusted by the pan pot weighting, applied according to the same spatial and spectral balancing rules as for mono sources.

[00257] Delay-based stereo placement can be used instead of, or in conjunction with, amplitude panning. As the algorithm is based on a traditional pan-pot approach, only the slight ITD delay is used to emphasise the existing source placement, typically between 1 and 2ms. Delay is chosen depending on the pan pot position, with a linear relationship up to 2ms from mono to hard-panning, as described by the following equation:

$$\tau_m[n] = Pf_m[n] \cdot 2 \times 10^{-3} \quad (10)$$

Where $\tau_m[n]$ is the time delay in seconds applied to the mth track.

[00258] Automating the use of time delays in the algorithm is still a work in progress, and whilst it is built into the software, by default the option is not currently used.

[00259] In addition to the amplitude and delay-based approaches there are numerous stereo effects which can be applied to both mono and stereo inputs. These include applications of reverb and chorus to provide depth, and width adjustment techniques, for example hard-panned double-tracked delayed sources. While these can provide interesting and effective additions to the stereo mix, they are largely used for artistic decisions and their use presents additional complexities to an autonomous algorithm.

[00260] Further research is required to establish rules for the use of stereo effects. A basic implementation of the double-tracked delay technique has been built into the software, which from preliminary testing has provided interesting additions to the final mix. As above, however, the option will not be in use until automation rules have been determined.

[00261] There are numerous panning laws determining the ratio of spreading signal power between left and right channels. The most common is the sine/cosine -3dB pan law, which has the property of equal power from left to right (see FIG. 31).

[00262] This law is used to place the sources in the stereo mix, using the following equations:

$$y_L[n] = \cos\left(P_m[n] \cdot \pi/2\right)$$

$$y_R[n] = \sin\left(P_m[n] \cdot \pi/2\right) \quad (11)$$

[00263] The above described algorithm may be built into a multi-track VST plug-in as shown in FIG. 32. Additional routines can be added to the algorithm for real-time use,

including an expandable track count by monitoring input activity, adjustment of parameters, and reset and on/off toggling capability.

[00264] The user interface, shown in FIG. 32, includes switches and controls for pan width and switching on/off, and visualisations to represent spectral and spatial balance, pan positions, and a goniometer to provide real-time feedback of the stereo activity. The goniometer coordinates are determined as shown in Equations 12 and 13, where *n* is the sample number of a circular buffer of stored output samples.

$$x_{coord}[n] = y_r[n] - y_l[n] \quad (12)$$

$$y_{coord}[n] = y_r[n] + y_l[n] \quad (13)$$

[00265] A listening test was conducted to evaluate the system against professional mixes and across a variety of genres.

[00266] A multiple stimulus with hidden anchor listening test was used for the subjective evaluation of the system. Similar to the MUSHRA framework for perceptual audio evaluation, this allows audio content to be rated to an individual's preference against a specific criterion. In this test an automatic mix was compared against three professional mixes. There was no reference included as there was no ideal mix; however a monophonic mix was used as a hidden anchor.

[00267] Three sound engineers with experience in studio and live applications were asked to create mixes for the test-audio: one semi-professional with 5+ years' experience ('Eng. 1') and two professionals with 15+ years' experience ('Eng. 2' and 'Eng. 3').

[00268] In these mixes, the only parameter that was modified was panning position. The multi-tracks were raw but were loudness balanced appropriately so the engineers could focus solely on panning location. The engineers were asked to use a Digital Audio Workstation with a -3dB paw law, to correspond with the method used in the automatic pan system. However, 'Eng. 3' used a -2.5dB pan law.

Gender	Male	10
	Female	1
Audio Production experience descriptors and number of years of experience.	**Beginner**	**2**
	<5 years	3
	Competent	**3**
	>5 years	4
	Proficient	**2**
	>10 years	3
	Expert	**4**
	>15 years	1
Critical listening skills and details of experience.	**Beginner**	**1**
	Competent	**2**
	Proficient	**7**
	Expert	**1**
	Musician	4
	Music related training	3
	PhD related subject	4
Hearing Impairments	Yes (slight tinnitus)	1
	No	10

Table 1: Results of preliminary questions to test subjects

[00269] As the system performs time-varying pan positioning, the engineers were informed to use automation where they thought appropriate. 'Eng. 1' and 'Eng. 3' created studio mixes where they were able to listen to and make changes any number of times to their preference, 'Eng. 2' performed a live mix, where the mixes could only be listened through to once or twice before real-time decisions had to be made. This was done to explore the different approaches. All mixes were created in an appropriate studio environment and the engineers provided detail of location, software and hardware used.

[00270] There were six multi-tracks chosen with varying genres including: 'Funk/Rock', 'Reggae', 'Jazz/Folk', 'Opera', 'Alt. Pop' and 'Gothic Electro'. The multi-tracks were taken from the Sound on Sound 'Free Multi-track Download Library'. Overall, the test-audio consisted of twenty-second excerpts of each song including three professional mixes ('Eng. 1', 'Eng. 2' and 'Eng. 3'), one auto-pan mix ('Auto') and a mono mix ('Mono').

[00271] There were 11 participants in total for the audio evaluation. Table 1 above shows the results of the preliminary test questions of participant's audio production and critical listening skills. All tests were conducted in an isolated listening room, with identical headphones, and a constant listening level.

[00272] For the first test participants were asked to rate the sound mixes in terms of their preference. The results are therefore entirely subjective. FIG. 33 shows the mean with error bars displaying the 85% confidence intervals using the T-distribution.

[00273] The professional mixes rate consistently high throughout with the 'Auto' mix scoring similarly or just below. However the 'Auto' mix out-performs the professional mixes on the 'Reggae' track. 'Eng. 2' and 'Eng. 3" in particular perform consistently well, with 'Eng. 1' performing well throughout and even outperforming all professional mixes in the 'Jazz/Folk' and 'Opera' tracks but being least consistent overall. In 'Alt. Pop', 'Eng. 1' rates low due to a corrupt audio file that had not been identified.

[00274] It can also be seen that the 'Mono' mixes are consistently rated lowest in all of the genres, with an exception for the 'Alt. Pop' song where it rates fairly high. This indicates a preference for a narrower stereo image for this song. The professional mixes that were rated highly had an audibly narrower stereo image compare to the 'Auto' mix, which was rated poorly. However this should be considered reflective of the individual song and not of the entire genre.

[00275] In a next test, the participants were asked to "Rate the appropriate use of stereo mixing considering: placement and balance of sources, placement of frequency content in the mix between left and right channels, and balance of overall content in the mix between left and right channels." Results are shown in FIG. 34.

[00276] This question was designed to make the participants focus on how well the mixes met the panning constraints, particularly in terms of the balance of left/right content. The 'Mono' mix was used as a hidden reference to expel unreliable results.

[00277] Similar to the results in Question 1 the 'Mono' mixes were rated consistently poorly except for the 'Alt. Pop' song. Overall, the professional mixes are consistently rated highly with the 'Auto' mix just below. Averaged mean and median results for all songs are displayed in FIG. 35 for each mix type, and for both tests. These give a clearer depiction of the overall performance of each mix type. It can be seen that 'Eng. 2' performs best in Q1 for the mean and median, and in Q2 for the median. 'Eng. 3' performs best in Q2 for the mean.

[00278] The averages differ because the mean is more affected by outliers, as shown in 'Eng. 1', as there is one very low score and generally much more varied results throughout. The median however takes the middle value and so is less affected by extremes and more by consistency, such as seen in 'Eng. 2' throughout Q1 and Q2.

[00279] In regards to the live and studio approaches to mixing, the live 'Eng. 2' mix performs most consistently overall, with the exception of the 'Eng. 3' studio mix for the mean of Q2. This was unexpected as 'Eng. 2' had less opportunity to modify decisions. However it

could be due to personal experience as a professional live engineer and technical consistency.

[00280] Overall, the results show that the auto mix performs consistently in Q1 and Q2 across genres. Generally it rates just below the professional mixes, except in the Reggae track which it out-performs, and the 'Alt. Pop' song where it performs badly. This indicates the generally successful application of the system across genres, rating closely to professional engineered mixes.

[00281] The results also indicate that the defined panning constraints are correct, due to the correlation between the results of Q1 and Q2. This shows that the system is closely following the approach that professionals take, which was a major objective at the start.

[00282] Generally the results were as expected. It was assumed that the professional mixes would out-perform the 'Auto', with the more experienced engineers 'Eng.2' and 'Eng. 3' performing above and 'Eng. 1' performing below or similarly. The professional mixes out-perform the 'Auto' apart from the mean in Q1. However, the most important result to highlight is the ability of the auto-pan to consistently work across multiple genres, falling only just below the standard of professionally engineered mixes.

[00283] A surprising result was that the mono tracks on occasion rated quite highly. It seems apparent that certain genres may benefit from a narrower stereo image, such as the 'Alt. Pop' and 'Reggae' tracks.

[00284] Accordingly, the above describes a comprehensive new approach to the autonomous stereo positioning for music production. This approach has applications in both live and off-line applications, and scored highly in a listening test in comparison with three professional mixes.

[00285] *Compression Processor:*

[00286] FIG. 36 shows a multi-track mixing processor 124 that is configured to extract features of the audio tracks to allow for a compression of multiple audio tracks. In the example shown in FIG. 36, the feature extraction at 470 corresponds to extraction of information from the signal used to perform compression, and the cross adaptive processing corresponds to at least one compression optimization at 472. It can be appreciated that, similar to FIGS. 5 and 6, the optimization stage 472 generates controls 474 that are used by respective processors to modify the respective track at 476 and generate the output 102'.

[00287] *Example 1:*

[00288] Dynamic range compression is a nonlinear audio effect that reduces the dynamic range of a signal and is frequently used as part of the process of mixing multi-track audio recordings. A configuration for automatically setting the parameters of multiple dynamic range compressors (one acting on each track of the multi-track mix) will now be described making reference to FIGS. 37 to 46. The perceptual signal features loudness and loudness range are used to cross-adaptively control each compressor 124. The compressor configuration is fully autonomous and includes six different modes of operation. These were compared and evaluated against a mix in which compressor settings were chosen by an expert audio mix engineer. Preferences were established for the different modes of operation, and it was found that the autonomous configuration was capable of producing audio mixes of approximately the same subjective quality as those produced by the expert engineer.

[00289] As discussed above, multi-track audio mixing is the production of a coherent sound mixture from multiple, individual audio sources, and is usually performed by skilled and experienced audio mix engineers. The signal processing operations routinely used in audio mixing include level balancing, spectral equalisation, spatial positioning and dynamic range compression (DRC). These and other processes are applied to individual tracks, or sub-groups of tracks, in order to produce an audio mixture in which the constituent sound sources are appropriately balanced, which sounds subjectively pleasing and which achieves a certain artistic intention.

[00290] The majority of digital audio effects (DAFX) are designed for single channel applications and process an audio input based on parameters which are specifically chosen by the user. Advances in digital signal processing have lead to the investigation of adaptive DAFX in which the effect parameters are set automatically based on signal features with little or no user interaction. It has been recognized that very few of the DAFX currently available on the market are designed to automate any of the mixing process. The potential advantages of such DAFX are significant – from allowing on-experts to produce quality mixes with little or no prior experience, to speeding up the workflow of professional mix engineers.

[00291] Individual sound sources within a multi-track mix are processed not in isolation but with respect to all other sound sources, or a subset thereof. Intelligent multi-track DAFX are, therefore, cross-adaptive inasmuch as the automatic control of one channel in the mix depends on features derived from other channels.

[00292] DRC has many applications when it comes to mixing multi-track audio. Typical examples include controlling the transient attack of percussive instruments such as drums, raising the overall loudness of a sound source by applying compression with make-up gain and providing a more consistent signal level. In this example, a system is described and evaluated which, using high level, perceptual audio features, automatically and cross-adaptively applies DRC to individual tracks within a multi-track mix.

[00293] DRC is a nonlinear audio effect that narrows the dynamic range (i.e. the difference between the loudest and quietest parts) of a signal. This is achieved by applying an attenuation to the signal whenever its level exceeds a given value. The set of parameters that can be used to describe a generic dynamic range compressor are threshold, ratio, knee, attack time, release time and make-up gain.

[00294] Threshold is the level above which the input signal is attenuated. Ratio is the amount of attenuation that is applied (for example, a ratio of 3:1 would result in a 1 dB increase in output signal level for every 3 dB increase in input signal level above the threshold). Knee is the dynamic range over which the ratio increases to its specified value (low values result in so-called 'hard' knees and high values result in 'soft' knees). Attack time is (approximately) the time it takes for the compressor 124 to reach the desired attenuation ratio once the signal overshoots the threshold (or enters the knee region), and release time is (approximately) the time it takes for the compressor 124 to return to a state of no attenuation once the signal returns back to a level below the threshold. Make-up gain is applied uniformly to the whole output signal after attenuation.

[00295] The basic goal when mixing multi-track audio is to ensure that the individual sound sources are blended together into a coherent-sounding whole, and modest amounts of DRC are particularly suitable for this task. There are a number of different possible compressor design choices one can make when implementing DRC. In this example, a digital implementation of a feed-forward monaural compressor 124 with a smoothed de-coupled peak detector is illustrated. It should be noted, however, that the intelligent, multi-track dynamic range compression method described herein is independent of the compressor model.

[00296] Two important signal features, loudness and loudness range, are used to control the application of DRC in this example. Although loudness is a subjective quality, objective measures have been recommended which approximate the characteristics of the human hearing system. In this example, the standard developed by the International

Telecommunication Union, as described in ITU-RBS, 1770-2, is used. Specifically, for a monaural signal, the loudness is defined as:

$$\text{Loudness} = -0.691 + 10\log_{10}\left(\frac{1}{N}\sum_{i=1}^{N} y^2[i]\right) \quad (1)$$

where $y[i]$ is the input signal after it is passed through a head-related transfer function filter and then a high pass filter. The unit of this loudness measurement is the LU (Loudness Unit), which is similar to the dB.

[00297] In general, short-term loudness measurements of a signal will vary over time. Loudness Range (LRA) quantifies the amount of this variation. It can be thought of as the perceptual equivalent of dynamic range, and is therefore of interest when considering intelligent DRC systems. A technical definition of LRA is given by the European Broadcasting Union (EBU) in Tech-3342. This specifies that loudness measurements are taken in sliding analysis windows of length 3 seconds with at least 66% overlap between consecutive windows. The resultant vector of short-term loudness measurements is then processed using a two-stage cascaded gating scheme. The first stage is an absolute gate set to -70referenced relative to the maximum possible loudness of a digital signal (0 LU Full Scale). The second stage is a relative gate set to20 LU below the integrated loudness of the absolute gated signal. 'Integrated loudness' is a measure specified in EBU Tech-3341[11] and is intended to give a single overall loudness measurement for an entire piece of audio. LRA is then defined as the difference between the 10th and 95th percentiles of the twice-gated loudness measurements.

[00298] LRA is a natural signal feature to consider when looking at DRC but the 3 second time scale over which the EBU definition measures loudness may not be appropriate. DRC is often used to reduce the dynamic range of a signal over much shorter timescales, for example to reduce transient attack over time scales less than 50 ms.. In this example, LRA is calculated using loudness measured in 400 ms sliding windows at a rate of 7.5 Hz (i.e. 67% overlap of consecutive windows). Windows of length 10 ms and 3 seconds were also considered. However, it was found that when using 10 ms windows LRA is almost always high, since at this time scale signal peaks are captured, and it is more a measure of instantaneous amplitude variation than loudness range. Conversely, when using 3 second windows, LRA tends to be low since typical musical audio signals are often relatively uniform at this time scale. 400 ms was therefore considered to be a good intermediate window length, since it was usually found to result in large variations in LRA across different tracks of any given multi-track audio recording.

[00299] The configuration for the compressor 124 in this example was designed to produce a monaural output mix by automatically applying intelligent dynamic range compression with make-up gain to each individual track of a multi-track audio recording. The overall signal flow diagram is shown in FIG. 37.

[00300] The primary component of the configuration in FIG. 37 is the signal dependent and cross-adaptive automation of DRC. However, automation of the basic gain of each track in the mix is also included since this is perhaps the most fundamental task of any mixing process and is vital if the output is to sound at all reasonable. The aim of this automation is to ensure that all tracks have equal loudness within the mix. Pre-gain is applied to individual tracks, before they enter the compressor, so that the integrated loudness of each is equal to the maximum integrated loudness of the tracks before adding the gain. Adding a gain of G dB to a digital signal is achieved by multiplying all samples by a factor of $10^{G/20}$. Occasionally, this will result in one or more samples with an absolute value greater than 1, i.e. the signal will be 'clipped'. If such clipping is detected on any of the tracks after this gain stage, then, to avoid distortion, cross-adaptive normalisation is applied by multiplying all tracks by $1/\max\{x\text{clip}[n]\}$, where $x\text{clip}[n]$ is the post-gain signal with the highest clipping level. This process ensures that clipping is avoided but equal loudness between all tracks is maintained.

[00301] In order to minimize undesirable DRC artefacts (such as 'breathing', 'pumping' and 'drop outs'), attack and release times are automated using the spectral flux of the signal. Two separate modes of operation are defined to automate the compressor's threshold, ratio and knee parameters.

[00302] In 'threshold' mode, the compressor ratio is fixed at ∞ : 1 and the knee is set to the absolute value of the threshold. The amount of DRC is then controlled with the threshold parameter alone. As the threshold is lowered, the compressor is triggered more frequently and the knee becomes wider (or 'softer'). FIG. 38A provides an illustration of the input-output characteristics of a compressor in 'threshold' mode.

[00303] In 'ratio' mode, the amount of DRC is controlled via the ratio parameter alone. A fixed moderately hard knee of 3 dB can be chosen for this mode. In general, a signal with a higher root mean square (RMS) level requires a higher threshold in order to have the same amount of DRC applied as a signal with a lower RMS. Therefore, in 'ratio' mode, the compressor threshold is fixed relative to the RMS of the signal. After some investigation, 12 dB below the RMS was found to be a suitable level. The input-output characteristics of a compressor in this mode are shown in FIG. 38B.

[00304] Make-up gain is included in most compressor designs to allow the overall loudness of the output and input signals to be balanced. In this system, it is automated such that the integrated loudness of the compressor output is equal to the integrated loudness of the input and cross-adaptive normalisation is used to avoid clipping, as described above.

[00305] The appropriate amount of compression for each individual track is automatically and cross-adaptively determined by analyzing the LRA of all tracks in the mix. Two basic hypotheses were formulated *a priori* in order to design the cross-adaptive algorithm: 1) that DRC reduces LRA in a roughly monotonic way, and 2) that successful multi-track DRC helps to produce a coherent mix of sound sources by compressing tracks with higher LRA more than those with lower LRA, such that the difference between the highest and lowest track LRA is reduced.

[00306] Experiments have been carried out to quantify the effect of DRC on LRA in each of the two compressor modes ('threshold' and 'ratio'). Seven multi-track audio recordings, covering a variety of different genres, were used. These were the same recordings used later for the subjective listening tests (see discussion below). Each recording had up to 7 individual tracks, giving a total of 41 individual signals for testing. For each signal, LRA was calculated before and after applying DRC. In 'ratio' mode, the compressor ratio was varied from 1:1 to 10:1. In 'threshold' mode, the compressor threshold was varied between –25 dB and +25 dB relative to the signal's RMS.

[00307] The cascaded gate used when calculating LRA was designed to discount the noise floor of the signal and sections of silence. Applying make-up gain, particularly when the amount of DRC is high, can cause the noise floor to be amplified to the extent that it passes through the gate and is included in the LRA calculation. This is undesirable and can lead to anomalous LRA measurements. Therefore, the active sections of a signal were defined as those which contribute to the pre-DRC LRA measurement. Post-DRC LRA was then calculated based on the active sections only.

[00308] FIG. 39 shows how different amounts of DRC affected the LRA measurements. The results in FIG. 39A were obtained using an auto-compressor in 'threshold' mode, so that a lower threshold (moving left to right on the graph) corresponds to an increased amount of DRC. Averaged over all 41 test signals, the change in LRA was found to vary smoothly with threshold and, as expected, LRA decreased as the compressor threshold decreased. However, there was wide variation across the test signals. This comes from the fact that the maximum achievable amount of absolute LRA reduction is dependent on the LRA of the pre-DRC signal itself. It also appeared that, for many signals, there was a lower limit below

which it was not possible to reduce LRA via DRC alone. Similar results were observed using an auto- compressor in 'ratio' mode (see FIG. 39B).

[00309] It was verified that, for most signals, DRC can be expected to reduce LRA in a monotonic fashion, i.e. LRA decreases as the amount of DRC increases. However, it was found that this is certainly not true in all cases and there does not seem to be an obvious way of predicting, even approximately, to what extent LRA of a given signal will change after applying DRC.

[00310] Next, define the LRA range (denoted ΔLRA) of a multi-track recording as the difference between the highest and lowest LRA of individual tracks. Using the hypothesis that DRC improves the quality of multi-track mixes by reducing ΔLRA, a cross-adaptive algorithm was developed to automate the remaining single control parameter of the compressor (ratio if in 'ratio' mode, or threshold if in 'threshold' mode). The overall amount of DRC that is appropriate for a multi-track audio mix is largely a matter of personal taste and, therefore, three different levels of 'touch' were defined for the system: a 'light' touch results in an overall reduction in ΔLRA of 3 LU, a 'medium' touch reduces ΔLRA by 6 LU and a 'heavy' touch reduces ΔLRA by 9 LU in this example.

[00311] The cross-adaptive algorithm is as follows: no DRC is applied to the track with the lowest LRA; the largest reduction of LRA is sought for the track with the highest pre-DRC LRA; and the LRAs of the remaining tracks are reduced proportionally.

[00312] Specifically, for each track in the mix, a target LRA (i.e. theideal post-DRC LRA) is defined as:

$$LRA_T(i) = LRA(i) - \Delta LRA_{red}\left(\frac{LRA(i) - LRA(i_{min})}{LRA(i_{max}) - LRA(i_{min})}\right) \quad (2)$$

where i is the index number of the track, $LRA(i)$ is the pre-DRC loudness range of track i, i_{min} = argmin$i$($LRA(i)$), i_{max} = argmax$i$($LRA(i)$) and ΔLRA_{red} is the amount by which ΔLRA is to be reduced (dependent on the touch parameter).

[00313] Since it is typically not possible to know in advance precisely which compressor settings will result in the required LRA reduction for each track, these can be found by iteration. Starting values are based on the empirical data presented in FIG. 39 (using the average reduction in LRA for a given ratio or threshold). In 'ratio' mode, an optimal ratio is found to the nearest 0.1 dB. In 'threshold' mode, an optimal threshold is found to the nearest 0.5 dB. FIG. 40 shows an example of the target and achieved LRA reductions for tracks in a multi-track mix using Eq. 2, with different compressor modes and touches.

[00314] The configuration described above has six different forms of operation defined by the choice of compressor mode ('threshold' or 'ratio') and touch (light, medium or heavy). Ultimately, it may be desirable to have a user-defined 'touch' parameter within the system, so that the overall amount of DRC can be broadly controlled manually. However, it may be less desirable to retain an option regarding the mode of the auto-compressor since the difference between 'threshold' and 'ratio' modes would not be at all obvious to the average user. For this reason, and to investigate the performance of the automatic system compared with a manual application of multi-track DRC, subjective evaluations were carried out. The six different forms of automatic operation were compared together with a mix in which DRC was applied manually and a mix in which no DRC was applied.

[00315] Seven multi-track audio recordings were used to test and evaluate the system. They covered a range a musical genres: four Rock/Indie songs, two instrumental Jazz pieces and one Acoustic Folk song. For each recording, a 20 second excerpt was chosen manually based on the following criteria:

[00316] 1. The excerpt should be representative of the whole recording: it could be a section of a verse, a chorus or a transition, but all tracks should be active (i.e. all instruments included in the recording should be playing) for most of the excerpt.

[00317] 2. It must be possible to reduce the ΔLRA of the excerpt by at least 9 LU using our approach, i.e. the pre-DRC ΔLRA must be at least 9 LU and all of the target LRA reductions must be successfully achieved in both compressor modes for all touch settings.

[00318] 3. Different automatic mixes (using different compressor modes and touch settings) should sound audibly different, to extent that a non-expert listener would, with careful listening, be able to differentiate between them.

[00319] The test recordings were obtained from an online resource [18] and were downloaded as completely raw stems, i.e. no mixing, effects or post-recording processing had been applied already. The recorded instruments included: electric/acoustic guitar, electric bass guitar, double bass, piano, violin, acoustic drum kit and vocals. The drum kit tracks were sub-mixed before processing; one sub-mix for all the close drum microphones and one for overhead, room and cymbal microphones. Similarly, if a particular instrument had been 'double tracked', i.e. the same part recorded twice, then these were sub-mixed to a single track. This reduced the total number of tracks per recording to between three and seven.

[00320] The level of each track was set using an equal loudness algorithm, as described in section 4.1. However, five of the seven test recordings contained lead vocal tracks. It is

common for such tracks to be slightly louder in the mix than others [19], and so each lead vocal was boosted by 3–6 dB relative to its equal loudness level. The precise amount of boost was determined manually for each recording and was the same for all mixes.

[00321] A subjective listening test was designed to evaluate the quality of the different mixes with intelligent compression in relation to each other, an automatic mix with no DRC (i.e. with only gain automated) and a mix in which DRC settings were chosen manually by an audio engineer with extensive experience of multi-track audio mixing in both studio and live sound environments. To allow the manual mix to be completed, a graphical user interface was developed following the conventions of a standard software Digital Audio Workstation, e.g. vertical track layout, waveform views and soloing capability (see FIG. 41).

[00322] The same compressor implementation was used as for the automatic mixes. Attack, release, threshold, knee and ratio could be set manually for a compressor on each track, but the gain was automated in the same way as for the fully automatic mix. This may be be referred to herein as the 'expert manual' mix.

[00323] A comparison of the LRA reductions per track achieved by the expert manual mix settings compared to the target LRA reductions of the automatic system with a medium touch is shown in FIG. 42. It can be seen that the LRA reductions were, in general, quite similar. However, compared with the automatic system, the expert mixer applied more compression to the tracks with the lowest pre-compression LRA, and less to the tracks with the highest pre-compression LRA.

[00324] All of the expert manual mixes except one resulted in a reduction in ΔLRA (see Table 1). This provides some evidence in support of the hypothesis used to design the cross-adaptive control algorithm described. The average reduction in ΔLRA achieved by the manual mixes was 2.3 LU, just below that which is achieved by the automatic system with a light touch (3 LU).

Table 1: *Change in ΔLRA after expert manual application of multi-track DRC.*

Song number (name)	Pre-DRC ΔLRA	Post-DRC ΔLRA	Change in ΔLRA
1 (Rock/Indie 1)	11.1 LU	5.8 LU	-5.2 LU
2 (Rock/Indie 2)	11.2 LU	11.1 LU	-0.1 LU
3 (Rock/Indie 3)	15.2 LU	14.9 LU	-0.3 LU
4 (Rock/Indie 4)	17.3 LU	7.3 LU	-9.9 LU
5 (Instrumental Jazz 1)	12.5 LU	11.1 LU	-1.4 LU
6 (Instrumental Jazz 2)	12.0 LU	10.8 LU	-1.2 LU
7 (Acoustic Folk)	14.6 LU	16.5 LU	+1.8 LU
Average	—	—	**-2.3 LU**

[00325] A design similar to the 'Multiple Stimuli with Hidden Reference and Anchor' (MUSHRA) test was used. Participants were asked to rate each mix on a scale of 0 (very bad) to 100 (excellent), based on specific criteria. There were three tests in total, each with a different criterion for evaluation. The MUSHRAM Matlab interface was used to administer the tests (see FIG. 43) but was altered to allow playback of each mix to be interrupted. The MUSHRA test design was originally developed primarily for the evaluation of audio codec quality and specifies the inclusion of an objectively high quality 'reference' and an objectively low quality 'anchor'. However, since there is no obvious way of defining an objectively bad application of DRC within a multi-track mix, no anchor was used in these listening tests. Tests 1 and 2 used a reference sample, but test 3 did not. All mixes (both automatic and manual) were normalised to overall equal integrated loudness in order to avoid bias related to subjective preference for louder or quieter signals. For each test, four multi-track recordings were used from a range of genres and, for each of these, the order of presentation of the mixes was randomised. 15 participants were recruited (13 male, 2 female) and pre-screened to ensure that they all had no hearing impairments, were familiar with the concept and typical sound of DRC and had experience of analysing and listening critically to audio. The tests were administered under controlled conditions in a good listening environment. Post-screening of participants was conducted by analyzing the correlation between each participant's scores and the median of the scores from all participants. Pearson's correlation and Spearman's rank correlation were calculated to identify potential outliers. Then, after manual inspection of each participant's ability to award consistent grades, one participant was excluded from the test 1 results and two were excluded from the test 2 results. No participants were excluded from the results of test 3.

[00326] Test 1 was designed to evaluate subjective preference for the overall amount of DRC that was applied. Different levels of the touch parameter were used with the compressor mode fixed to 'threshold'. Two Rock/Indie recordings, one Jazz and one Acoustic Folk recording were used as the test material. The 'no DRC' mix was used as the (hidden) reference – i.e. it was labeled as the 'reference' but also included without a label amongst the other test mixes for evaluation. The 'expert manual' mix was also included. Participants were instructed to "*rate the following according to the appropriateness of the relative amounts of dynamic range compression applied to each individual sound source in the mix*" and were required to score at least one mix in the 'bad' category.

[00327] The results of this test are shown in FIG. 44. Relative scores for the different mixes were fairly consistent across all four recordings. It was only the threshold–light automatic mix that scored higher overall than the no DRC mix, indicating that participants did not think that very much DRC was required. The threshold–heavy mix scored significantly lower than all other mixes and was the only one to be rated bad overall. The fact that the no DRC mix scored relatively highly is also perhaps indicative of the difficulty in choosing appropriate DRC settings, and the extent to which badly chosen parameters can seriously degrade the quality of a mix.

[00328] Test 2 was designed to evaluate subjective preference for the two different compressor modes. The compressor mode was therefore varied but, since the amount of DRC was not being evaluated, the touch parameter was fixed; a heavy touch was chosen to provide the most audible differences between mixes. A different set of songs, but from the same genres as test 1, were used as the test material. Once again, the 'no DRC' (hidden) reference mix and the 'expert manual' mix were also included. Participants were asked to "*rate the following according to the sound quality of the dynamic range compression applied to each individual sound source in the mix*".

[00329] The results of this test are shown in FIG. 45. Overall, the expert manual mix scored the highest and both automatic mixes were rated poor. The sound quality of the ratio–heavy mix was preferred only slightly to the threshold–heavy mix. It is clear from tests 1 and 2 that the heavy touch automatic mixes were not favoured by the participants, most likely due to the audio quality being severely compromised by excessive amounts of DRC.

[00330] Finally, test 3 was designed to be an overall general evaluation of all six different automatic mixes against each other, a mix with no added DRC and the 'expert manual' mix. The test material consisted of four recordings from the Rock/Indie genre. The 'no DRC' mix

was included, but not as an explicit reference. Participants were asked to "*rate the following according to the overall quality of the mix.*"

[00331] The results of this test are shown in FIG. 46 and are fairly consistent with those from the previous two tests. The heavy touch automatic mixes had the lowest scores and, overall, the two light touch automatic mixes and the expert manual mix scored highest. The expert manual and threshold–light mixes were virtually tied with the top scores, but the ratio–light mix scored only slightly higher than the mix with no DRC.

[00332] In this test, as in test 2 in which ratings were based on the "sound quality" of DRC (FIG. 45), the expert manual mix of the 'Rock/Indie 4' song scored particularly highly compared with the automatic mixes. An examination of the manual mix compressor settings showed that they were largely similar to the automatic settings on all tracks, except one of the vocals. The expert had heavily compressed this track and reduced its LRA by 13.5 LU (by comparison, the heavy touch automatic mixes reduced its LRA by just 9 LU). This was clearly a successful strategy and is an example of the difficulties of designing an automatic system that is capable of consistently out-performing or matching the expertise and listening experience of a human engineer.

[00333] The configuration described in this example uses the perceptual audio features loudness and loudness range to automatically and cross-adaptively control multi-track DRC. The control strategy is based on the *a priori* hypothesis that the fundamental role of DRC in multi-track audio mixes is to reduce the difference between the highest and lowest individual track LRA, and that sound sources with higher LRAs require greater amounts of DRC. This hypothesis was substantiated empirically by examining the post-DRC changes in LRA achieved when an experienced mix engineer chose the compressor settings manually.

[00334] A number of different modes of automatic operation were designed and evaluated using a subjective listening test. Automatic mixes using a compressor in 'ratio' mode were not consistently rated higher than the 'threshold' mode mixes, but when using a heavy touch there was some evidence that 'ratio' mode was preferred. A light touch was found to be the most subjectively appropriate; both the ratio–light and threshold–light mixes performed very well when compared to the expert manual mixes. Indeed, the average reduction in ΔLRA achieved by the manual mixes was very similar to that achieved by the light touch automatic mixes. The best performing automatic mode of operation was threshold–light, and there was good evidence to suggest that, in this mode, the system is capable of automatically applying multi-track DRC to at least the same subjective standard as would be achieved if settings were chosen manually by an experienced mix engineer.

[00335] Two observations of the listening test results were the generally low scores (no mix was rated above 'fair' on average) and the consistency with which the 'no DRC' mix was rated as one of the best. There are a number of possible explanations for this. For example, the differences between the mixes may not have been clearly audible and the overall sound quality of the recordings themselves may not have been considered high. Other choices of multi-track recordings may, therefore, have yielded more conclusive results, or higher ratings overall.

[00336] The fundamental assumptions used in this system were that a reduction in ΔLRA would be preferred by listeners, and that the best way to achieve this is by applying DRC to each track in proportion to its pre-DRC LRA. The target LRA reductions are then defined by equation (2). However, there are many other possible approaches one could use. For example, equation (2) could be modified so that DRC is applied to all tracks in the mix, rather than all but one. Or the target LRA reduction could depend on additional signal features, or equal LRA across all tracks could be sought (i.e. ΔLRA = 0).

[00337] *Example 2:*

[00338] As indicated above, dynamic range compression is a nonlinear, time dependent audio effect. As such, preferred parameter settings are difficult to achieve even when there is advance knowledge of the input signal and the desired perceptual characteristics of the output. In the example illustrated in FIGS. 47 to 58,, an automated approach to dynamic range compression is provided where the parameters are configured automatically based on real-time, side-chain feature extraction from the input signal. Parameters are all dynamically varied depending on extracted features, leaving only the threshold as a user controlled parameter to set the preferred amount of compression. A series of automation techniques are analyzed in this example, including comparison of methods based on different signal characteristics. Subjective evaluation was performed with amateur and professional sound engineers, which established preference for dynamic range compressor parameters when applied to musical signals, and allowed a comparison of performance of the approaches discussed herein against manual parameter settings.

[00339] It has been recognized that a superfluously used dynamic range compressor 124 suppresses musical dynamics and may produce lifeless or even boring recordings deprived of their natural sound and character. Mastering dynamic range compression and refraining from overusing it is not an easy task even for professional engineers, due to the versatility of the effect, together with the large number of choices regarding its use. Being a nonlinear effect, if used carelessly it may alter a signal in unpredictable and undesired ways. Setting

up the compressor parameters in a sensible way is nontrivial because the effects of each parameter are not often apparent and there is typically a high degree of correlation between the different parameters.

[00340] Automating the compressor parameters and in general, the parameters of any audio effect, can provide evident advantages to the user. Although not intended to replicate artistic choices, when the compressor 124 is used to decrease dynamic range, automation will save users the trouble of properly setting the effect to avoid sound artifacts and in many cases it will give better results. In addition to this, for a highly diverse signal there might not be a static set of parameters that would be optimal. An automated compressor with parameters that dynamically adapt to the signal's characteristics may give better results than a set of static human preferences.

[00341] Compressors with partly automated parameters (such as 'autorelease,' for instance) have already found their way to production both as analogue and digital designs. In some existing designs, the automation of the time constants is performed by observing the difference between the peak and RMS levels of the signal fed in the side-chain. In at least one previous system, an RMS measurement was used to scale the release time constant. The RMS measurement, however, is an absolute one and dependent on overall signal level. It does not directly take into account the transient nature of the signal.

[00342] The concept of replacing a user-controlled ratio and knee width with an infinite ratio and single, user controlled knee width has been used previously in both analogue and digital compressors, albeit with a static knee width. Similarly, automatic make-up gain can be found in some compressor designs, but only as signal independent static compensations that do not take into account loudness, even though the main purpose of make-up gain is to achieve the same loudness between the uncompressed and compressed signals.

[00343] Related work has been concerned with reverse engineering a compressor, based on analysis of the signal before and after compression. However, these prior attempts do not offer full automation of compressor parameters based on the input signal characteristics. Furthermore, previous listening tests or user studies have not been used to explore the effectiveness of a compressor automation approach, even for automating a single parameter within the design.

[00344] In this example, a configuration is presented and evaluated in which a set of methods is used to automatically determine appropriate values for the standard compressor parameters in an intelligent way, depending on the input signal's properties and statistics, and the intended use of each parameter. This way, the required user interaction is reduced

to a minimum, ultimately to a single setting of how much compression is desired. First, a description of the parameters that are automated and the compression model used in described. Next, a series of methods are provided to effectively reduce the number of user-adjustable parameters. Next, a subjective evaluation of the various automation methods is presented, with both amateur and professional sound engineers. Finally, the results are summarized.

[00345] This example provides a configuration to automate a dynamic range compressor 124, such that it can be operated or set with a single parameter. All other parameters are dependent on the input signal characteristics. Thus, the set of parameters of a typical compressor is first described, followed by a compressor design that was used in this example.

[00346] FIG. 47 depicts a block diagram of an example compressor configuration that can be used in this example. The compressor parameters define the behavior of the side-chain which determine the instantaneous compressor gain $c[n]$. As described above, the most commonly used compressor parameters may be defined as follows.

[00347] *Threshold* (denoted T) defines the level above which compression starts. Any signal overshooting the threshold will be reduced in level.

[00348] *Ratio* (R) controls the input/output ratio for signals overshooting the threshold level. It determines the amount of compression applied. As shown in Figure 2, the ratio sets the slope of the static compression characteristic when the input level exceeds the threshold.

[00349] *Attack and release times* (τA and τR) provide a degree of control over how quickly a compressor acts. They are also known as time constants. Instantaneous compressor response is not sought because it introduces distortion on the signal. The attack time defines the time it takes the compressor to decrease the gain to the level determined by the ratio once the signal overshoots the threshold. The release time defines the time it takes to bring the gain back up to the normal level once the signal has fallen below the threshold.

[00350] A *Make-Up Gain* control is usually provided at the compressor output. The compressor reduces the level (gain) of the signal, so that feeding back a make-up gain to the signal allows for matching the input and output loudness level.

[00351] The *Knee Width* (W) option controls whether the bend in the static compression characteristic, depicted in FIG. 48, for input levels near the threshold has a sharp angle or has a rounded edge. A sharp transition is called a Hard Knee and provides a more noticeable compression. A softer transition where the ratio gradually grows from 1:1 to a set

value in a transition region on both sides of the threshold is called a Soft Knee. It makes the compression effect less perceptible.

[00352] Optionally, a compressor 124 may also have *look-ahead*, given in milliseconds. With look-ahead, the side-chain determines the amount of compression based on the current signal level, but the control is applied to a delayed version of the input signal. However, this introduces latency, and was not analyzed in this example. It may be noted that such an embodiment may be adapted to be used in live performances and broadcasts. Also, use of a peak detector (as opposed to RMS), was found to provide a quick response to changes in signal characteristics, hence lessening the need for look-ahead.

[00353] A compressor 124 also has a set of additional controls which are sometimes found in modern designs. These include a Hold parameter, Side-Chain filtering, and many more.

[00354] The compressor model used in this example is a feed-forward compressor with a smoothed decoupled peak-detector, whose output is given as the input signal times a control value determined by signal level estimation in a side-chain configuration:

$$y[n] = c[n] \cdot x[n] \qquad (1)$$

where $x[n]$ denotes the input signal, $y[n]$ the output signal, $c[n]$ the control voltage. The control voltage is calculated from a copy of the input signal that passes through the side-chain, as seen in FIG. 47. The side-chain first includes a peak detector to provide an instantaneous estimate of signal level:

$$x_G[n] = 20\log 10\,|x[n]| \qquad (2)$$

[00355] The gain computer implements a static compression curve with input $xG[n]$ and output $yG[n]$, and is given by Eq. (3), where the sample number $[n]$ has been omitted for readability:

$$y_G = \begin{cases} x_G & 2(x_G - T) < -W \\ x_G + \dfrac{(1/R - 1)(x_G - T - W/2)^2}{/(2W)} & 2\,|(x_G - T)| \le W \\ T + (x_G - T)/R & 2(x_G - T) > W \end{cases} \qquad (3)$$

where T, R and W are the *threshold*, *ratio* and *knee width* parameters.

[00356] FIG. 48 presents Eq. (3) for both hard and soft knee, and a make-up gain giving displacement from the diagonal. The (static) amount of compression is thus

$$x_L[n] = x_G[n] - y_G[n] \quad (4)$$

[00357] Smoothing is performed by a gain smoothing (also known as ballistics) stage,

$$y_1[n] = \max(x_L[n], \alpha_R y_1[n-1] + (1-\alpha_R)x_L[n])$$
$$y_L[n] = \alpha_A y_L[n-1] + (1-\alpha_A)y_1[n] \quad (5)$$

where $\alpha_A = e^{-1/(\tau_A f_s)}$ and $\alpha_R = e^{-1/(\tau_R f_s)}$ are filter coefficients derived from the compressor's attack and release times, τ*A* and τ*R*, and *fs* is the sampling frequency. The control voltage is thus found by adding the make-up gain, *M*, and then converting this back from decibel to linear scale:

$$c[n] = 10^{\wedge}(M - yL[n])/20 \quad (6).$$

[00358] Such a compressor design yields smooth and relatively artifact-free performance for a wide variety of signals, when compared with alternative designs. However, due to the influence of the attack envelope on the release trajectory in Eq. (5), the measured release time is approximately τ*A* + τ*R*.

[00359] Since the compressor parameters are used in different stages of the compressor design their automation methods can be independent of each other, even though they might be based on the same signal statistics.

[00360] Very short attack and release times should be avoided because they introduce a number of unpleasant artefacts such as pumping, breathing, low frequency distortion and other artifacts. Very long attack and release times are also rarely beneficial. The longer the attack the less responsive the compressor is to the signal. Likewise, a long release time may cause perceived dropouts after short transient sounds or reshape the decay part of notes and modify the sound of instruments.

[00361] Since most signals are time varying, dynamically varying time constants are preferred, so that they can adapt to the nature of the transient, steady state and decay components in the signal. To minimise artifacts, a suitable auto-attack and release mechanism would choose shorter time constants when the input signal is highly transient or percussive, and longer time constants if it is a more steady state signal. In this example two methods are discussed, a time domain approach and a time-frequency processing approach that give greater control and flexibility over the selection of suitable time constants.

[00362] The crest factor is defined as the ratio of peak signal level to root mean squared (RMS) signal level over a given duration. In order to measure the short-term crest factor of a

signal, without introducing any latency, one can combine a peak detector and an RMS detector, as given in Eq. (7),

$$\begin{aligned} y_{Peak}^2[n] &= \max(x^2[n], \alpha y_{Peak}^2[n-1] - (1-\alpha)\,|x^2[n]|) \\ y_{RMS}^2[n] &= \alpha y_{RMS}^2[n-1] + (1-\alpha)x^2[n] \\ y_C[n] &= y_{Peak}[n] / y_{RMS}[n] \end{aligned} \quad (7)$$

where forgetting factor $\alpha = e^{-1/(\tau fs)}$ is calculated from time constant τ and sampling frequency *fs*. The RMS detector is a 1-pole smoothing filter applied to the square of the input signal, also known as an exponential moving average filter. The peak detector above has instantaneous attack and a smooth release trajectory. If one chooses the peak detector and RMS detector time constants to be identical, one ensures that the release envelopes of both detectors are the same, and that the peak detector's output cannot be less than the detected RMS output. The crest factor is independent of overall signal scaling and is therefore compatible with the design goal of level-independence.

[00363] The time constant τ for the two detectors determines the integration time of the crest factor measurement, and was set at 200ms based on informal testing. Though the crest factor of a steady state signal is fairly low, it increases once the signal contains transients. Transients show high amplitude, but are of short duration (typically less than 10ms) in relation to the 200ms integration time. Thus their contribution to the RMS value is typically much less than their contribution the peak value. Thus, the crest factor can be used to locate transient parts in the signal, like note onsets.

[00364] The maximum attack time can be set to τ_{Amax}=80ms and the maximum release time to τ_{Rmax}=1s. One can find RMS detectors in some compressors with time constants on this scale to prevent low frequency distortion and other artifacts. In order to avoid dropouts and pumping, the effect of a high crest factor on the release time should be significant. For this reason we divide each maximum time constant by the square of the crest factor. The crest factor for a pure sine wave is $\sqrt{2}$, so we then multiply by 2 to ensure that the maximum time constant is reached for sinusoidal input signals. Finally to compensate for the influence of the attack time on the measured release time in this example, as discussed above, one can subtract the attack time from the release time. Thus, the time varying automation of attack and release times is given by Eq. (8).

$$\begin{aligned} \tau_A[n] &= 2\tau_{A\max} / y_C^2[n] \\ \tau_R[n] &= 2\tau_{R\max} / y_C^2[n] - \tau_A[n] \end{aligned}. \quad (8)$$

[00365] The short time Fourier transform (STFT) of an input signal *x* is defined as

$$X(n,k)=\sum_{m=-N/2}^{N/2-1} x(nh+m)\omega(m)e^{-j2\pi mk/N} \tag{9}$$

where $X(n, k)$ represents the kth frequency bin of the nth frame, $\omega(m)$ is an N-point Hamming window and h is the hop size between adjacent windows.

[00366] The spectral flux (*SF*) measures how quickly the power spectrum of a signal changes and offers detection based on amplitude or energy information of the signal. It is calculated from the change in magnitude of the STFT over two successive frames, and it is restricted to count only those frequency bins where the energy is increasing. The normalized spectral flux is then defined as:

$$SF(n)=\frac{\sum_{k=-N/2}^{N/2-1} H(|X(n,k)|-|X(n-1,k)|)}{\sum_{k=-N/2}^{N/2-1}|X(n,k)|} \tag{10}$$

where $H(x)=(x+|x|)/2$ is the half-wave rectifier function.

[00367] Though spectral flux is typically used as an onset detection function, it can readily be used for transient detection purposes. The more transient a signal is, the higher its spectral flux value will be and the shorter the time constants that are needed to achieve proper compression. The spectral flux is more sensitive than the crest factor and this enables it to detect more subtle changes to the signal. For the spectral flux calculation we use a window $N = 1024$ points for the Fourier transform, with a hop size between adjacent windows $h = 512$, i.e., 50% overlap between windows. These settings were chosen because they produce narrow peaks for the spectral flux function, at the same time instances as the crest factor. However, as shown in FIG. 49, the spectral flux is also able to catch the change in frequency of the sine wave.

[00368] In order to overcome the problem of associating a spectral flux value to a maximum time constant, the normalized spectral flux can be scaled to the range of values of the crest factor. The crest factor is usually highest for the very first sample, which can be easily shown if we take equation (7) and set

$$\gamma_C[1]=\frac{1}{\sqrt{(1-\alpha)}}\geq\frac{x[n]}{\sqrt{\alpha\gamma_{RMS}^2[n-1]+(1-\alpha)x^2[n]}}\geq\gamma_C[n] \tag{11}$$

[00369] This value is used as the high boundary for the spectral flux function and scales all other spectral flux values accordingly.

[00370] The attack and release time constants play an important role close to the onsets of notes, since onsets will probably cross the threshold level and trigger the compression. Therefore, the time constants can be correlated to the peaks of the spectral flux, which in turn are closely related to note onsets. Because spectral flux peak values are a lot higher compared to their corresponding crest factor values for a crest factor time constant of 200ms, the square of these values does not need to be used in order to achieve short enough times after transients. Instead we use an instantaneous attack peak detector with a release time of 2 ms for calculating attack times and 9ms for the release times to smooth the spectral flux curve.

$$\begin{aligned}\tau_A[n] &= 2\tau_{A\max} / SF_{smooth}[n] \\ \tau_R[n] &= 2\tau_{R\max} / SF_{smooth}'[n] - \tau_A[n] \\ SF_{smooth}[n] &= \max(x[n], \alpha SF_{smooth}[n-1] + (1-\alpha)SF[n])\end{aligned} \quad (12)$$

[00371] The maximum attack and release time constants are the same as those used for the crest factor method. The parameter γ was set to 0.8 to provide a less intense change of release times as opposed to attack times.

[00372] Nevertheless, the difference between γ=0.8 and γ=1 is small and for the sake of simplicity it can be dropped in a basic implementation of the method.

[00373] FIG. 50 is an example of the crest factor and the smoothed spectral flux methods used to obtain the attack and release times.

[00374] Both threshold and ratio parameters relate to the static compression characteristics. In an automated compressor it may be desired that the user to only have to adjust one parameter that will define the desired compression amount they want to apply to the audio signal. One may then let the threshold be manually chosen, set the ratio parameter to infinity and use a soft knee with an automated knee width that will vary with time depending on the compression of the signal. This method is based on the idea that a very soft knee can also be seen as an automatic ratio.

[00375] By using a soft knee in the gain computer stage and setting the ratio to ∞:1, the slope of the static compression curve of Eq. (3) becomes

$$\frac{dy_G}{dx} = \begin{cases} 1 & 2(x_G - T) < -W \\ 1 - (x_G - T + W/2)/W & 2|(x_G - T)| \le W \\ 0 & 2(x_G - T) > W \end{cases} \quad (13)$$

[00376] Hence, the signal will be perfectly limited once it exceeds $T+W/2$. Below that point the slope will gradually increase, reaching ½ (equivalent to a ratio of 2:1) exactly at T, and it will keep decreasing until $T-W/2$, where it will become 0 (no compression at all, equivalent to a ratio of1). So by setting the ratio to infinity and varying the knee-width one can access the whole range of compression ratios. FIG. 51 presents the static compression behavior for various knee widths and a set threshold.

[00377] If the compression applied is for short periods of time, so only a few peaks are trimmed and the average gain reduction is small, then one might want the compressor toast as a hard limiter. On the other hand, if the signal is heavily compressed and the average gain reduction is high, one might want a smoother and less obvious compression effect.

[00378] For the automatic knee mechanism we propose an adaptive method based on the average gain reduction of the compressor, given by the following equation:

$$\begin{aligned} c_{Dev}[n] &= \alpha c_{Dev}[n-1] + (1-\alpha)(c[n] - c_{Est}) \\ W[n] &= 2.5(c_{Dev}[n] + c_{Est}) \end{aligned} \quad (14)$$

cDev provides a smooth estimate of how much the control deviates from an estimated value based on the parameter settings of the compressor. A control voltage estimate, *cEst*, is used to bias the averaging filter, by subtracting the estimate before the filtering and adding it back in afterwards. A reasonable setting for the estimated value is given from the Threshold and Ratio settings, $cEst=T(1-1/R)/2$. This, initializes the average gain reduction at a value close to its intended values and allows the control voltage estimate to quickly adapt to changes in parameter settings during real-time operation.

[00379] The averaging time constant of the filter, found from $1/(\ln)\ s\ \tau = -f\ \alpha$, needs to be carefully chosen. A time constant that is too short will follow the compressor's gain reduction too quickly and a long time constant will be too slow in following the gain reduction curve and reaching the intended values. τ =2s was chosen, so that use of average gain reduction for make-up gain described below does not interfere with the release envelope.

[00380] The scale factor 2.5 was derived empirically from informal listening tests. The result is a knee width that is slowly and smoothly varied with time.

[00381] The following method suggests a way to optimize the knee width using information on the input signal extracted with the normalized spectral flux. Signals with extensive transient content will have their spectral flux values above a certain level, considerably higher compared to that of a signal with fewer transients, since in every frame step there is typically significant transient content captured by the SF. For example, in FIG. 52, the spectral flux values of the drums sample are constantly above 0.1 and around 0.2 while the spectral flux values of the bass sample reach a minimum level of 0.05.

[00382] First, the minimum levels of the spectral flux are calculated using a modified version of an instantaneous attack decoupled peak detector and then a low-pass filter used to find the moving average of these values. The method can be summarized as:

$$SF_{\min}[n] = \min(|SF[n]|, \alpha SF_{\min}[n-1] + (1-\alpha)SF[n])$$
$$SF_{\min.avg}[n] = (1-\alpha_2)SF_{\min}[n] - \alpha_2 SF_{\min.avg}[n-1] \quad (15)$$

where the coefficients α, α_2 were based on time constants τ = 2ms and τ_2 = 1ms respectively, to obtained a desired performance.

[00383] The evaluation results for the knee width automation showed that the relationship between the average gain reduction and the preferred knee width is nonlinear and instrument independent. Therefore, a polynomial of order *k* was used to describe the following relationship:

$$W[n] = 2.5c_{Avg}{}^{k}[n]$$

[00384] A steady state signal should result in a roughly constant knee width, to prevent unnecessary modulation of compressor parameters. On the other hand, signals that are very transient in nature should produce a compressor whose knee width varies more with gain reduction, so that it can both act like a limiter for high amplitude signals, and provide a smooth transition to no compression on low amplitude signals.

[00385] As shown in FIG. 52, a highly transient signal, such as a percussive drums sample, will never present very low values for minima since there will always be transient activity captured by spectral flux. A steady-state signal will have spectral flux minima reaching lower values since initial transients of the attack part of the notes will quickly fade while the steady-state part will remain longer. Based on the average of the spectral flux minima we set *k* to values that, as seen in section 3, produce the desired behavior for the knee width.

$$k = \begin{cases} 0.6 & SF_{\text{min,avg}} > 0.1 \\ 0.05 & SF_{\text{min,avg}} \leq 0.1 \end{cases} \qquad (17)$$

[00386] The aim of a make-up gain function is to achieve equal loudness between the compressor input and output signals (though it may also be used to maximise loudness of compressed recordings, contributing to the 'loudness war'). A first approach is to estimate the make-up gain based on the average amount of applied compression. From Eq. (14):

$$c_{\text{make-up}}[n] = -(c_{\text{Dyn}}[n] + c_{\text{Lim}}) \qquad (18)$$

[00387] A second approach uses a loudness function to compare perceived loudness before and after compression. The EBU standard for loudness, based on a thresholded implementation of the ITU 1170 standard, may be used to compare the loudness of tracks in multitrack audio, in order to automate time varying fader controls. Here, the EBU standard is used to measure loudness of the uncompressed and the compressed signal. This enables us to extract the loudness difference between the two signals and use it to calculate the make-up gain needed for the compressed signal. Even with the application of loudness-based make-up gain, the compressor is still able to significantly reduce the loudness range of the signal.

[00388] Subjective evaluation was performed with two groups of subjects: 9 expert mixing engineers (Professional group) and 7 amateurs who had experience with dynamic range compression (Amateur group). A 'method of adjustment' style test was performed to obtain quantitative data on how humans set up and use a dynamic range compressor in their environment with their own equipment. Each test subject was provided with a VST plug-in of the compressor, test instructions and 4 short audio tracks of drums, bass played in "slap" style, soft vocals and acoustic guitar. The instructions included a series of listening tests in which the users had to tune individual parameters to their preferred setting while keeping all other parameters fixed at predefined settings. The predefined values were usually such that they would generate obvious amounts of compression and make any compression artifacts easily spotted by the listener.

[00389] The results were compared with what the automation method had chosen as preferred automated parameter settings. While the preferred human choices for each setting were single, static values, the compressor's automation method is an adaptive process, producing dynamically varied values. Therefore, some form of grouping or averaging of the dynamic values had to be performed.

[00390] For the evaluation of attack and release times, other parameters were predefined as follows: threshold at -30dB, ratio at ∞:1 and knee width at 0dB (hard knee). For auto-attack time we calculated the average out of all attack time values that fall in a time period equal to the maximum attack time after every possible onset (peak of the spectral flux) of the signal. For the auto-release time, since we cannot predict exactly when release times will be used we simply found the mean out of all the values.

[00391] FIG. 53 and FIG. 54 present box plots for the preferred attack and release time respectively. The box in each column indicates the interquartile range. The bottom of the box corresponds to the lower quartile (25^{th} percentile) and the top of the box to the upper quartile (75th percentile). The dash within the box shows the median value of the data set, and the vertical black line shows the sample range from the minimum to the maximum sample value.

[00392] The small interquartile ranges for bass sample show that most of the testers agree that it requires a very fast attack time in order to prevent the initial transient of each note from slipping through. To avoid a drop-out after the very hard initial transient, most also set the release time to a very fast value (median of 26 ms for the professionals).

[00393] Test subjects preferred a longer attack and release time for the guitar compared to the bass. This resulted in the spectral flux providing good results for the auto-attack, and the crest factor providing good results for the auto-release. For vocals, the automatic release is quite slow, especially for the crest factor approach. The median for both amateurs and professionals suggests a much faster release time constant (of 100 to 150 ms).

[00394] For the drums sample, which is similar to the bass in terms of richness in transients, the choices of the professionals are highly diverse. If one concentrates at the median time constant, it is much longer this time (19 ms), which indicates that it might be desirable to preserve the initial transient of each drum hit.

[00395] Due to the lack of transients in the soft vocals sample, the automatic compressor chooses a slow attack time in order to prevent it from being distorted. Although the interquartile range for the professionals is high, the median for both professionals and amateurs suggests an attack time of only approx. 6 ms.

[00396] Generally, the spectral flux automation method performs better than the crest factor method. This is mainly due to the fact that for the crest factor method we were depending on the long time constant to produce a smooth result while for spectral flux we used a subsequent smoothing filter to achieve this.

[00397] The parameters for knee width evaluation were ratio at ∞:1, attack time at 0.5 ms and release time at 100 ms. Test subjects were asked to choose their preferred knee width for threshold values -18dB, -25dB and -40dB. For comparison of automation with preferred user knee width setting we calculated average gain reduction for each threshold value and from that found the corresponding average knee width that the automation methods used. The test concentrated on the two more percussive signals, drums and slap bass, since the influence of the knee is more pronounced on such content. FIG. 55 and FIG. 56 present the results from this test.

[00398] The results on the drums sample confirm that testers prefer a softer knee for heavier compression, i.e., lower threshold. The trend is more clearly seen in the professional results than in the amateur ones. It seems that using average gain reduction as a means of adjusting knee width was successful. For the slap bass sample, the median for the professionals suggests a fairly constant knee width regardless of threshold, and amateurs prefer a softer knee at a threshold of -18 dB than at -25 dB. Therefore, using the same automation as that for the drums sample gives poor results. The spectral flux modification corrects this shortcoming and the results are more consistent with what users would use.

[00399] FIG. 57 shows the choices of the professionals for the drums sample as a function of threshold, with the choices of the gain-reduction dependent automation method indicated by a thick line. Almost all results show an upward trend (broader knee width for lower threshold), although the trajectories themselves differ regarding scaling and offset. This justifies a choice of using the information from the average gain reduction to adjust the width of the knee. Furthermore, including information from the spectral flux helped fine-tune the method to better fit the preferred user choices.

[00400] As shown in FIG. 57,. Individual choices (in dB) for the drums sample knee-width experiment for professionals. Thick black line represents the median.

[00401] The parameters for the make-up gain evaluation test were threshold at -30dB, ratio at ∞:1, attack time at 0.5 ms and release time at 100 ms. These settings (low threshold and very short time constants) were chosen to guarantee that all 4 test signals would be heavily compressed. Test subjects were asked to manually vary the make-up gain, until the compressed signal has the same loudness as the uncompressed signal. Results are presented in FIG. 58.

[00402] In FIG. 58, ox plots for the Make-up gain evaluation are shown, with results in dB with median value (dash), average control voltage automation (dot) and loudness-based automation (cross).

[00403] The full range of results for this experiment varied significantly (e.g. one professional tester applied 24 dB of make-up gain to the bass sample, which is more than 10 dB above the median value). A few testers reported that they found it difficult to judge whether the two signals appear equally loud when their dynamic range is so different and came to different results, whether they concentrated on the attack (transients) or the sustain (steady-state) part of the sound. However, the interquartile range is quite small, within only 3 dB for all of the make-up gain experiments. This means that most testers agreed on a make-up gain for a given sample.

[00404] Comparing the results to the automation, the average control-voltage dependent make-up gain is quite accurate for the guitar and the vocal samples. In both cases most professionals would apply slightly more and most amateurs would apply slightly less make-up gain but may not provide the desired gain for the slap bass and drums audio samples. Both signals were characterized by short-lived high peaks with high transient content located mainly in their loud onsets. These loud transients contained in the original signals are quite significant for perception of the signal's overall loudness, so when those transients are suppressed by the compressor, we require more make-up gain than the actual average gain reduction in order to achieve equivalent loudness.

[00405] The loudness-based make-up gain comes a lot closer to the median value of the users' experiment and can be characterized as accurate for all cases apart from the drums where the make-up gain is about 3 dB more than what the testers believe it should be. That can be explained due to the transient nature of drums, which makes loudness measurement difficult.

[00406] In this second example, the example proposes a series of methods to automate most of the parameters of a digital dynamic range compressor 124. These methods are independent of one another for each parameter and can be used together or separately in different compressor models. The performance of these methods were studied and compared against the choices of human operators as discussed above.

[00407] This example therefore presents a unique subjective evaluation of preference for dynamic range compressor parameters when applied to musical signals. However, a key purpose of the evaluation was to understand how the proposed automation methods perform in comparison to user preferences.

[00408] It may be noted that an alternative approach to the automation of the static compression characteristic would be to automate the threshold to follow the RMS of the signal and let the user adjust the ratio based on what they prefer. This would avoid keeping

the ratio fixed at infinity, which confines the compressor's operation to be close to a limiter. For the attack and release times we proposed the use of Spectral flux as a method for transient/onset detection.

[00409] When using the crest factor to obtain the time constants one can achieve smoother operation by increasing the time constant used for the peak and the RMS detector in the crest factor calculation. A similar approach could also be followed for the spectral flux. The use of a detector with long attack and release times could smooth the spectral flux curves. This is an alternative to the approach we used to calculate the attack and release times with the spectral flux.

[00410] The evaluation of the automatic make-up gain suggests that the use of EBUR-128 and ITU-R BS.1770-2 is effective and shows general agreement with user preference.

[00411] In general, creating an automatic compressor could be simplified by knowing what type of signal it will be applied. For example, an auto-compressor that only has to work on drums for instance can make many more assumptions about its input signal than a compressor that is expected to sound well on an arbitrary tracks. The auto-release mechanism, for instance, could potentially benefit from some form of tempo-dependence, at least for very rhythmic signals.

[00412] Finally the system could also be configured to allow the user to control how the automation behaves by being able to set the meta parameters controlling the release time. The compressor behaviour would still adapt to the signal, but allow the user to maintain control over compression characteristics.

[00413] *Example 3:*

[00414] A further example for a compressor processor 124 will now be described, making reference to FIGS. 59 to 66.

[00415] As noted above, DRC is commonly used in audio production, noise management, broadcasting, and live performance applications. To a large extent, DRC defines much of the sound of contemporary mixes. DRC can make sounds louder, punchier, richer, and more powerful. Yet if used incorrectly or superfluously, dynamic range compressor suppresses musical dynamics, producing lifeless recordings deprived of their natural sound character.

[00416] It has been observed that there is a high degree of correlation between the different compressor parameters. In order to draw ultimate results from a compressor, one should understand the function of each control and how these affect the dynamic behaviour

of each treated instrument. Inappropriate setting up of the compressor parameters produces artifacts like pumping and breathing.

[00417] Parameter automation of a dynamic range compressor 124 using computerised signal analysis can provide advantages to audio amateurs or musicians who lack of expert knowledge in sound acoustics, signal processing. Mixing production is no doubt an artistic and technical task, however, much of the initial work follows established rules and best practices. Automating the compressor 124 speeds up the routine work and the trail-and-error process of setting the effect to avoid sound artifacts. In addition to this, conventional use of a static set of compressor parameters might not be optimal when the signal is highly diverse. An intelligent compressor 124 with parameters that dynamically adapt to the signals' characteristics would result a better mix than a set of static human preferences.

[00418] Automatic dynamic range compressor 124 solutions have been considered, however, in prior systems, threshold is still manually chosen, ratio is set to infinity and a soft knee with an automated knee operating on spectral flux is used to decide the amount of compression. A cross-adaptive method for automating multi-track DRC using perceptual audio features loudness and loudness range has been explored, wherein the control strategy is based on the *a priori* that the fundamental role of DRC in multi-track audio mixes is to reduce the difference between the highest and lowest individual track loudness range, and that sound sources with higher loudness range require greater amounts of DRC.

[00419] In this example, a configuration is described and evaluated for a new cross-adaptive approach of intelligent multi-track compression. By 'intelligent', the multi-track compressor 124 is able to analyse the signals' content, dynamically adapt to audio inputs, automatically configure parameter settings, and exploit best practices in sound engineering to modify the signals appropriately. The configuration in this example is built upon a cross-adaptive audio effect processing architecture, where the signal processing of an individual source is the result of the relationships between all involved sources. In addition to the fully automated mode, a new single user-control (instead of the conventional 6-dimension control space of standard compressor) to determine how much compression is desired is also introduced to provide an optimal, minimal user interaction.

[00420] First, brief descriptions of generic compressor parameters and the digital compressor model used in this example are provided. Next, assumptions to instruct multichannel compression automation that extracted from best practices in mixing engineering are described. Next, a rule-based, feature-driven, automatic multichannel compression algorithm is described. A series of subjective evaluations are then provided.

[00421] As explained above, a typical set of parameters of a single-track compressor can be defined as follows.

[00422] *Threshold* determines the *extent* of the compression.

[00423] The threshold defines the level above which gain reduction starts. Any signal exceeding the threshold is known to be an overshooting signal and would normally be reduced in level. The threshold is most often calibrated in dB in digital full scale (Izhaki 2008).

[00424] *Ratio* determines the *degree* of the compression.

[00425] The ratio defines the input/output ratio for signals overshooting the threshold level, i.e. if a signal is above the threshold by 10dB with a 2:1 ratio, the signal is attenuated by 5dB.

[00426] Threshold and ratio define the basic transfer characteristic of a compressor 124, as shown in FIG. 59.

[00427] *Attack and Release* determines the degree of how fast a compressor acts. The attack and release are also known as time constants or response times. The attack determines how quickly gain reduction can rise once the signal is over the threshold, while release determines how quickly gain reduction can fall once the signal is below the threshold. Digital compressors can respond to sudden level changes instantly. However, quick response sometime introduces distortion or artifacts on the signal. The attack and release phases in a compressor are shown in FIG. 60.

[00428] *Knee Width* controls whether the threshold-determined point in the transfer characteristics of a compressor has a sharp or smoothed *transition*. A sharp transition is called *hard-knee*, which brings more intrusive compression that draws more distinctive effects. A smoothed transition is called *soft-knee* where the ratio gradually increases from unity to the set ratio that spreads to both sides of the threshold. It makes compression effect more transparent. Hard knee and soft knee compression are illustrated in FIG. 61.

[00429] *Make-up gain*

[00430] Because the compressor is reducing the level of the signal, the ability to add a fixed amount of make-up gain at the output is usually provided so that a post-compression optimum level can be obtained.

[00431] A compressor 124 might also have some additional controls such as hold, side-chain filtering, look-ahead, etc.

[00432] A digital compressor model design which is a feed-forward compressor with a smoothed branching peak detector shown in FIG. 62, may be employed in this example of a compressor 124.

[00433] The output of the proposed compressor model can be simply described as Eq. (1). Where [n] is the input signal, $y[n]$ is the output signal and $c\ n$ denotes the control voltage that determined by the side-chain processing.

$$[n] = c[n] \cdot x[n] \quad (1)$$

[00434] The side-chain first includes a peak detector to provide an instantaneous estimate of the input signal level.

$$[n] = 20 \log_{10} x[n] \qquad (2)$$

[00435] Then the gain computer implements a static compression curve with input [n] and output y[n], and is given by Eq. (3), where the sample number n has been omitted for readability. *T*, *R* and *W* are the threshold, ratio and knee width parameters.

$$y_G = \begin{cases} x_G, & 2(x_G - t) < -W \\ x_G + \dfrac{\left(\frac{1}{R} - 1\right)\left(x_G - T + \frac{W}{2}\right)^2}{2}, & 2|x_G - T| \le -W \\ T + \dfrac{x_G - T}{R}, & 2(x_G - t) > W \end{cases} \qquad (3)$$

[00436] The amount of compression is thus:

$$x_L[n] = x_G[n] - y_G[n] \quad (4)$$

[00437] Smoothing is then performed at a level detection stage using Exponential Moving Average (EMA):

$$y_L|n| = \begin{cases} \alpha_A y_L|n-1| + (1-\alpha_A)x_L|n|, & x_L|n| > y_L|n-1| \\ \alpha_R y_L|n-1| + (1-\alpha_R)x_L[n], & x_L|n| \le y_L[n-1] \end{cases} \quad (5)$$

Where $\alpha_A = e^{-1/(\tau_A f_S)}$ and $\alpha_R = e^{-1/(\tau_R f_S)}$ are filter coefficients derived from the compressor's attack and release times, τ_A and τ_R, and f_s is the sampling frequency.

[00438] The control voltage is thus found by adding the make-up gain M, and then converting this back from decibel to linear scale:

$$c[n] = 10^{\frac{(M - y_L|n|)}{20}} \quad (6)$$

[00439] Control rules/Assumptions to automate the multichannel compressor 124 are now described.

[00440] Assumption 1: A source track with a higher degree of level fluctuations should have more compression.

[00441] It is generally believed that one of the main purposes of applying compression is to balance level fluctuations. In the early stage, professional mixing engineers often compress instruments that have high string-to-string level variations, such as vocals, drum tracks so that their relative levels are roughly consistent. The degree of level fluctuations of an audio source can be referred to its dynamic range. However, signal measures do not correlate well with perception. Thus loudness range measurement might be more applicable when it comes to mixing decisions. This principle has been further developed into an automatic approach of reducing the loudness range difference between multichannel inputs signals. Alternatively, crest factor calculated from the peak amplitude of an audio waveform divided by its RMS value can also be a coarse measurement of dynamic range. The crest factor is independent of overall signal scaling and therefore level independent. The crest factor of a steady state signal is fairly low, but it increases once the signal contains transients, which have high amplitude in a short time period (typically less than 50 ms). Thus transients' contribution to the RMS value is much less than their contribution to the peak value. So crest factor is often used as transient indicator.

[00442] *Assumption 2: A source track with more low frequency content is more prone to compression.*

[00443] It has been recognized that although mixing engineers might not realize it, many of them did admit to being prone to low-end compression. The more uniform the low-end frequency response is, the better it will translate between speakers. This un-conventional idea that the amount of compression somehow depends on the frequency response, especially the low-end. It has been found that there is a clear trend towards larger crest factors/loudness range in higher octaves, which is indeed in the agreement with the assumption. FIG. 63 illustrates average crest factor (left graph) and average loudness range (EBU 2010) per octave. FIG. 64 provides a comparison of the commercial mix dataset rules (dark) to the uncompressed equal-loudness mix results (light). Crest level per octave is shown on the left graph, and the loudness range per octave is shown on the right graph.

[00444] A rough comparison of the previous results against those of non-compressed mixes (simply by summing all tracks with equal loudness) is shown in FIG. 64.

[00445] The validity of this assumption was further evaluated by the follow-up listening tests, where eight different scenarios for compression were set up for subjective evaluation. Subjects are asked to evaluate on clarity and sound quality. Results indicated that low-frequency response is the preferred feature for compression amount, as both conditions are defined to be frequency dependent, with emphasis on low-end sensitivity.

[00446] *Assumption 3: Compressor attack and release should be set up depend on the transient nature of the signal.*

[00447] Attack times usually span between 5 ms and 250 ms and release times are often within the 5–3000 ms range (Izhaki 2008). Most values in the literature point to a usage that lies in the middle. It's generally accepted that attack and release time parameters is employed to catch the transient nature of the sound. Some commercial compressors offer a switchable auto- attack or auto-release. Mostly this is achieved by the compressor observing the difference between the peak and RMS levels of the side-chain signal. Several approaches to auto-adaptive implementations of compressor time-constants have been tested, for example to propose to scale the release time based on RMS values, and scale attack and release times based on the crest factor. More recently, it has been found that the subject and scale are parameters based on either modified crest factor or modified spectral flux.

[00448] *Assumption 4: Compressor knee width should be set up based on the amount of compression.*

[00449] As mentioned above, a soft-knee enables smoother transition between non-compressed and compressed parts of the signal and thus a more transparent compression effect. Generally speaking, the larger the amount of estimated compression applied on the signal, the more obvious the effect would be. In order to produce a natural compression effect in the automatic mixing system, the compressor knee width therefore should be adaptively configured based on the estimated amount of compression applied on the signal. And the amount of compression applied largely depends on the relationship between threshold and ratio. Threshold defines the extent of compression while ratio defines the degree of compression. In previous auto-mixing applications the auto-adaptive nature of the compressor 124 was modified so that only one parameter was needed to control the unit. This can be said to be an approach that converges towards the assumption that there is a maximum/optimum amount of compression.

[00450] *Assumption 5: Compressor make-up gain should be set so that output loudness equals input loudness.*

[00451] Compressor make-up gain has been seen to be an additional feature that allows an extra (static) gain stage, and is indistinguishable from post-compressor fader movement in most implementations. In automatic mixing, it could be stated that loudness setting is done post-compression. Although this is not as simple as it may seem, as an adaptive time-changing make-up gain will act as an auto-leveler and the compressor properties will be lost, so realtime implementation implies a very delicate choice of time constants for the adaptation process, and a convergence strategy using cumulative values similar to the one that has been proposed for several instances of auto- mixing. Offline implementation is simpler, as the make-up gain is simply given by the difference between the EBU R128 weighted input loudness and the output loudness, and this can even be calculated for the whole duration of a file. In previous automatic mixing implementations there is an EBU R128 weighted post-compression make-up gain to compensate the perceptual loudness loss due to compression effect. Listening tests that have been done show that the EBU loudness method indeed produced a good approximation of how professional mixing engineers would set the make-up gain. This is one of the assumptions that should be taken as self-evident for some auto-mixing applications.

[00452] *Assumption 6: There is a maximum and optimal amount of compression that depends on sound source features.*

[00453] Quantitative descriptions have been made about the amount of compression that should be applied on different instruments. Some suggestion are summarized as below:

- For vocals, gain reductions that fall between 3 and 6 dB will often sit well in a mix (although some rock vocalists will want greater compression).
- Bass accepts ranges from 5 to 10 dB.
- Acoustic guitar from 3 to 8 dB.

[00454] The auto-adaptive nature of the compressor 124 has been modified so that only one parameter was needed to control the unit. This can be said to be an approach that converges towards the assumption that there is a maximum/optimum amount of compression. A discrete separation between transient like signals, and steady state ones, can allow the former to have larger variability for W in order to accommodate for the transient peaks. In another proposal, a more diversified, high-level semantic based feature set is suggested as a basis for dynamic range compression mapping. Other types of mappings include having song section define the amount of DRC applied. Also, percussive tracks can be manually separated from sustained sounds, as they are assumed to need a different treatment.

[00455] The algorithm utilized in this example will now be described, making reference to FIG. 65.

[00456] The intelligent multichannel compressor adapts the cross-adaptive audio effect processing architecture, where the effect exploits the interdependence of the input audio features in order to output the appropriate amount of compression, and incorporates best practices as constrained control rules – see above. The block diagram of this configuration is shown in FIG. 65:

[00457] First, a copy of the multichannel input signals is fed to a side-chain processor where audio features of each individual track are extracted for further cross-adaptive analysis.

[00458] The relationship between the RMS value and the peak magnitude of the signal has always been an important indicator for automating compressor parameters.

[00459] Initially, computation of RMS level of a signal *x* is given by:

$$x_{RMS}^2[n] = \frac{1}{m}\sum_{M=0}^{M-1} x^2|n-m| \qquad (7)$$

[00460] When applying a sliding window, Eq.7 reduces to:

$$x_{RMS}^2[n+1] = \frac{x^2|n+1| - x^2|n+1-M|}{M} + x_{RMS}^2[n]$$

[00461] Or an EMA filter (also know as low-pass one pole filter):

$$x_{RMS}^2[n+1] = (1-\alpha_{RMS})x^2|n+1| + \alpha_{RMS}x_{RMS}^2[n] \qquad (9)$$

Where $\alpha_{RMS} = e^{-1/(\tau_{RMS}fs)}$ and fs is the sampling frequency. Time constant τ_{RMS} determines the integration time of the RMS detector.

[00462] Peak magnitude of the signal is calculated as:

$$x_{Peak}^2|n| = \max(x^2|n|, (1-\alpha_{Peak})x^2|n+1| + \alpha_{Peak}x_{Peak}^2|n|) \quad (10)$$

[00463] If the peak detector's and RMS detector's time constants α_{peak} α_{RMS} are set to be identical, the release envelopes of both detectors are guaranteed to be the same, and the peak detector's output is no less than the detected RMS output. The crest factor, $C\ n$ defined

as the ratio of the peak magnitude to the RMS value of the signal over a certain integration time is thus:

$$C[n] = \frac{|x_{Peak}[n]|}{x_{RMS}[n]} \quad (11)$$

[00464] A spectral feature, low-frequency percentage defined as the ratio of the sum of the frequency content up to 1KHz to the sum of the whole frequency content, is proposed to instruct the intelligent compressor. A standard Fast Fourier Transformation (FFT) with Hanning windows is performed on the each frame of the signals to obtain their spectral distribution. The low frequency percentage, $LF(m)$ of track m is then calculated from Eq. (12):

$$LFP(m) = \frac{\sum_{k=0}^{1000\,(Hz)} X(m,k)}{\sum_{k=0}^{f_s/2} X(m,k)}, k \in [0, f_s/2) \quad (12)$$

where k represents the k^{th} frequency bin of the spectral distribution of track m, (m, k), the output of the FFT.

[00465] Audio features extracted from each individual signal are fed to the cross-adaptive feature analysis block, where the relationship between all the sources involved in the audio mix being taken into account, producing the final instructions for compressor parameter automation. Two cross-adaptive features, percussivity weighting factor and frequency weighting factor, are introduced based on *Assumptions 1 and 2* respectively.

[00466] As assumption 1 suggests, crest factor can be used as a rough measurement of level fluctuation, or in other word, the degree of percussivity of the source. A cross-adaptive feature called percussivity weighting factor, is thus introduced to describe the relationship between all the input sources in terms of the degree of level fluctuation or percussivity. It may be calculated based on the crest factors of input signals.

[00467] First, the average value of the crest factors of all the sources are calculated:

$$C_{avg} = \frac{1}{M}\sum_{m=1}^{M} C(m) \quad (14)$$

Where m is the index of the track number and M is the total number of the input tracks. The average crest factors value is then used as an adaptive threshold or reference for percussivity weighting factor, PW . The mapping between PW m and (m) is constructed

using a modified Gaussian distribution centred around C_{avg}. Recalling a generic Gaussian function is:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad (15)$$

where b is the position of the center of the peak, and c, the standard deviation, controls the width of the "bell". $PW\ m$ is then formulated as follow empirically:

$$PW(m) = \begin{cases} e^{-\frac{(C(m)-C_{avg})^2}{2c^2}}, & C(m) \leq C_{avg} \\ 2 - e^{-\frac{(C(m)-C_{avg})^2}{2c^2}}, & C(m) > C_{avg} \end{cases} \quad (16)$$

[00468] c is set to 2 based on informal testing. As Eq. (16) suggests, $PW\ m \in 0,2$ meaning the larger the $PW\ m$ value, the more percussive the m track is. A typical shape of $PW\ m$ as a function of (m) is shown as FIG. 66. Equation 16 guarantees that most values of $PW\ m$ are centred on the adaptive reference Cavg.

[00469] Frequency weighting factor is introduced to describe the relationship between all input sources in terms of low-frequency content based on the extracted low-frequency percentage feature. The frequency weighting factor of m^{th} track, $F(m)$ is defined as the ratio of the low frequency percentage to the average low-frequency percentage of all sources, formulated as Eq. 17:

$$FW(m) = \frac{LFW(m)}{\frac{1}{M}\sum_{m=1}^{M} LFW(m)} \quad (17)$$

[00470] Ratio automation is achieved by mapping the values of the cross-adaptive feature descriptors into the actual ratio values of each compressor. We use the percussivity weighting factor as the descriptor for the degree of level fluctuation and frequency weighting factor as the descriptor for the amount of low frequency content. And the mapping rules are derived from the constrained controls provided above. Since ratio parameter determines the degree of compression. We can go ahead and interpret assumption 1 & 2 as follows:

[00471] 1. A source track with a higher degree of level fluctuation acquires larger ratio values.

[00472] 2. A source track with more low frequency acquires larger ratio values.

[00473] It may be observed that there is a monotonic relationship between the ratio value and the values of the two descriptors. Therefore one can formulate the automatic ratio as a function of the descriptors $PW\ m$ and $FW\ m$ as below:

$$ratio(m) = 1.25 \cdot PW(m) + 1.25 \cdot FW(m) \qquad (18)$$

where m is the index of track number.

[00474] The scale factor 1.25 is derived empirically from informal listening tests. Since ratio values are calculated per frame, a one-pole EMA filter is employed to smooth the variation between neighbouring frames:

$$ratio'(m, n+1) = \alpha_{ratio}\, ratio'(m, n) + (1 - \alpha_{ratio}) ratio(m, n) \qquad (19)$$

[00475] Notice that denotation changes from m to $m, n + 1$ to reflect the frame-by-frame processing.

$$\alpha_{ratio} = e^{-1/(\tau_{ratio} f_s)} \qquad (20)$$

[00476] Smoothing factor τ_{ratio} set to 1000 ms derived empirically from informal listening tests. Time-varying ratio values are thresholded so that no ratio values are less than 1 at the end of calculation.

[00477] RMS value of the signal offers good starting point of setting threshold. Thus the threshold is initially formulated as below:

$$threshold(m) = a \cdot 20 \cdot \log_{10} x_{RMS}(m) \qquad (21)$$

[00478] The scaling factor a = 1.1 is derived empirically informal listening.

[00479] The threshold automation is further improved by taking the degree of the percussivity of the input signals into account. There is a practical mixing rule saying “the lower the threshold the lower the ratio”. The idea behind is that once the rough amount of compression is achieved, lowering the threshold (more compression) and then lowering the ratio (less compression) will result in roughly the same amount of compression, but one with a slightly different character. Put another way, lower threshold means more is affected, while higher ratio means more effect on the affected. Professional mixing engineers generally prefer setting a higher threshold with a bigger ratio to setting a lower threshold with a smaller ratio when employing compressor to treat a track that is more percussive or contains more transients. Therefore, we add a small offset determined by the percussivity weighting factor of the signal to the threshold automation and it becomes:

$$threshold(m) = 1.1 \cdot 20 \cdot \log_{10} x_{RMS}(m) + 1.5 \cdot PW(m)$$

[00480] The scaling factor of 1.5 in this example may be chosen empirically from informal listening tests. Since PW ranges from 0 to 2, so that it will only result in a small positive offset (0 dB to 3 dB). The threshold automation obeys the mixing rule that a lower threshold for a more percussive tracks.

[00481] Follow assumption 4, the knee width is set to half of the absolute value of threshold.

$$knee(m) = \frac{|threshold(m)|}{2} \quad (23)$$

[00482] This equation indicates that the lower the threshold, the wider the knee width.

[00483] For attack and release automation, previous algorithms may be used, additionally following assumption 3 such that compressor attack and release should be set up depend on the transient nature of the signal.

$$\tau_A[n] = \frac{2\tau_{A-MAX}}{C[n]} \quad (24)$$

$$\tau_R[n] = \frac{2\tau_{R-MAX}}{C[n]} \quad (25)$$

where the maximum attack time $\tau_{A\text{-}MAX}$ is set to 80 ms and the maximum attack time $\tau_{R\text{-}MAX}$is set to 1000 ms in this example.

[00484] The purpose of automatic make-up gain in this example configuration is to achieve equal loudness between the compressor input and output signals according to Assumption 5. We employ the EBU Loudness standard, a thresholded implementation of the ITU 1170 standard (ITU, 2001) (EBU, 2010), for loudness measurement of the signal. Modification for a better loudness estimation for multichannel audio content may also be considered. Thus, the make-up gain can be derived from the loudness difference of the signals, before and after the compression:

$$G(m,n) = 10^{\frac{l_{in}(m,n)-l_{out}(m,n)}{20}} \quad (26)$$

where $G\ m, n$ is the make-up gain of the m^{th} track at n^{th} frame, and l_{in}, l_{out} are pre-compression and post-compression loudness of the signal. EMA filter is also applied to smooth the variation of make-up gains between adjacent frames:

$$G'(m,n+1) = \alpha_{gain} \cdot G'(m,n) + (1 - \alpha_{gain}) \cdot G(m,n+1) \quad (27)$$

where $\alpha_{gain} = e^{-\frac{1}{\tau_{gain} f_s}}$.

[00485] A shorter time constant, τ_{gain} offers more accurate estimation of loudness variation due to compression. However, it generates high degree of make-up gain fluctuation between frames. As a result, the automatic make-up gain might act like anti-compression instead. Through informal listening test, τ_{gain} is set to 2.5s to provide a relatively steady gain meanwhile reasonable loudness estimation is maintained.

[00486] It can be appreciated that full automation of the compressor 124 can be implemented, with partial/experimental settings, to allow full user control. Also, in a variation of the plug-in shown herein, an additional parameter that affects the ratio and threshold can be implemented using, for example, a slider that increases or decreases the ratio/threshold to allow the user to selectively choose “more” or “less” compression. Moreover, it can be appreciated that after automation and before application of the compression, personal preferences can be taken into account (e.g. for certain mixes, sub-groups, etc.) by applying a weighting to the ratio and/or threshold.

[00487] It should also be noted that the compression techniques described herein are particularly suitable for cross-adaptive processing, rather than applying compression based only on the track being analyzed. In other words, the configurations described herein enable the system to look at other tracks to determine dynamic or spectral features, e.g. a moving average over time. For example, more or less percussivity can be applied based on what other tracks are exhibiting.

[00488] It will be appreciated that any module or component exemplified herein that executes instructions may include or otherwise have access to computer readable media such as storage media, computer storage media, or data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Computer storage media may include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. Examples of computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be

accessed by an application, module, or both. Any such computer storage media may be part of the production system 10, mixing engine 104, etc.; any component of or related thereto, or accessible or connectable thereto. Any application or module herein described may be implemented using computer readable/executable instructions that may be stored or otherwise held by such computer readable media.

[00489] The steps or operations in the flow charts and diagrams described herein are just for example. There may be many variations to these steps or operations without departing from the principles discussed above. For instance, the steps may be performed in a differing order, or steps may be added, deleted, or modified.

[00490] Although the above principles have been described with reference to certain specific examples, various modifications thereof will be apparent to those skilled in the art as outlined in the appended claims.

Claims:

1. A method of performing automatic equalization in an automatic multi-track audio mixing system, the method comprising:

dividing a frequency spectrum for an audio track into a plurality of bands;

determining a value representative of a spectral characteristic in each band;

comparing the value in each band to a corresponding band in another signal; and

applying a filter to at least one band in either the audio track or the other signal based on the comparison.

2. The method of claim 1, wherein the other signal is another track in the audio signal, and the filter performs an offset to de-mask one of the tracks relative to the other.

3. The method of claim 1 or claim 2, further comprising applying at least one filter in at least one band to remove low-frequency content in either the audio track or the other signal.

4. The method of any one of claims 1 to 3, further comprising applying at least one filter or compression to at least one band to remove noise.

5. The method of any one of claims 1 to 4, further comprising applying at least one filter or compression to remove unwanted vocal noise.

6. The method of claim 1, wherein the other signal corresponds to a target spectrum, and the filter performs an equalization of the audio signal to the target spectrum.

7. The method of claim 6, wherein the target spectrum is accessed from a predetermined target profile.

8. The method of any one of claims 1 to 7, being repeated on a frame-by-frame basis.

9. The method of any one of claims 1 to 8, further comprising:

determining a first filter response to be applied to a plurality of frequency bands in a signal;

computing a first gain to be applied at each frequency band;

computing a second filter response using the first gains to be applied at each band and corresponding target gains; and

applying the second filter response.

10. The method of claim 10, further comprising:

analyzing the second filter response to compute a second gain to be applied at each frequency band;

computing a third filter response using the second gains and the target gains; and

applying the third filter response.

11. The method of claim 9 or claim 10, wherein the method is performed in a plurality of iterations, the filter response being applied after the plurality of iterations.

12. The method of any one of claims 9 to 11, further comprising computing analysis frequencies for a signal being analyzed.

13. The method of claim 7, wherein the target profile is generated by:

determining a magnitude spectrum for a target audio file;

determining an average magnitude spectrum in each of the bands; and

generating the target spectrum and storing as the predetermined target profile.

14. The method of claim 2, wherein a plurality of audio tracks are obtained; and

for at least one track:

calculating a prominence of masking between tracks in frequency bands;

assigning priority based rules to determine a masking to target; and

controlling a plurality of equalisation filters per track.

15. The method of claim 14, wherein the equalization filters comprise at least one of: gain, Q factor, and centre/cut-off frequency.

16. The method of claim 2, wherein a plurality of audio tracks are obtained; and

for at least one track:

determining a cut-off frequency for setting a high pass filter; and

applying the high pass filter to the at least one track to remove low frequency content in the track.

17. The method of claim 9, wherein if the audio file is in stereo, the method further comprises converting to mono for analysis.

18. A method of performing automatic mastering equalization of audio tracks, the method comprising:

dividing a frequency spectrum for an audio signal into a plurality of bands;

determining a value representative of each band for the audio signal;

comparing the value in each band of the audio signal with corresponding bands in a target spectrum; and

applying at least one filter to the audio signal to perform an equalization towards the target spectrum.

19. The method of claim 18 being repeated on a frame-by-frame basis.

20. The method of claim 18 or claim 19, wherein the target spectrum is accessed from a predetermined target profile.

21. A method of performing gain compensation, the method comprising:

determining a first filter response to be applied to a plurality of frequency bands in a signal;

computing a first gain to be applied at each frequency band;

computing a second filter response using the first gains to be applied at each band and corresponding target gains; and

applying the second filter response.

22. The method of claim 21, further comprising:

analyzing the second filter response to compute a second gain to be applied at each frequency band;

computing a third filter response using the second gains and the target gains; and

applying the third filter response.

23. The method of claim 21 or claim 22, wherein the method is performed in a plurality of iterations, the filter response being applied after the plurality of iterations.

24. The method of any one of claims 21 to 23, further comprising computing analysis frequencies for a signal being analyzed.

25. A method of generating a target spectrum for equalization of audio content, the method comprising:

determining a magnitude spectrum for a target audio file;

determining an average magnitude spectrum in each of the bands; and

generating the target spectrum and storing as a predetermined target profile.

26. The method of claim 25, wherein the target spectrum comprises an average of the average magnitude spectrum and one or more additional magnitude spectra.

27. The method of claim 25 or claim 26, wherein if the audio file is in stereo, the method further comprising converting to mono.

28. A method of performing equalization of audio tracks, the method comprising:

obtaining a plurality of audio tracks;

for at least one pair of tracks:

comparing each of a plurality of frequency bands in a frequency spectrum for one track to corresponding frequency bands in another track;

applying at least one rule according to which tracks are being compared, the at least one rule being indicative of a type of filtering to be performed; and

determining at least one filter to be applied to at least one of the pair of tracks; and

applying the at least one filter.

29. A method of performing equalization of audio tracks, the method comprising:

obtaining a plurality of audio tracks; and

for at least one track:

determining a cut-off frequency for setting a high pass filter; and

applying the high pass filter to the at least one track to remove low frequency content in the track.

30. A computer readable medium comprising computer executable instructions that when executed by a processor perform the method of any one of claims 1 to 29.

31. A system comprising at least one processor, and a memory comprising computer executable instructions that when executed by the processor, configure the system to perform the method of any one of claims 1 to 29.

32. An automatic multi-track audio mixing system comprising a processor and memory, the memory comprising computer executable instructions for:

performing a single track equalization;

performing a multi-track equalization;

obtaining user input on semantic rules for at least one of genre and style; and

enabling an interaction between equalization units in a mixing system signal chain.

33. A method of performing automatic multi-track equalization of audio tracks, the method comprising:

obtaining a plurality of audio tracks;

for at least one pair of tracks:

comparing each of a plurality of frequency bands in a frequency spectrum for one track to corresponding frequency bands in another track;

applying at least one equalisation filter to de-mask overlapping frequency content;

applying at least one equalisation filter to remove unwanted low-frequency content;

applying at least one filtering and compression operation to remove unwanted background noise;

applying at least one filtering and compression stage for removing unwanted vocal noise.

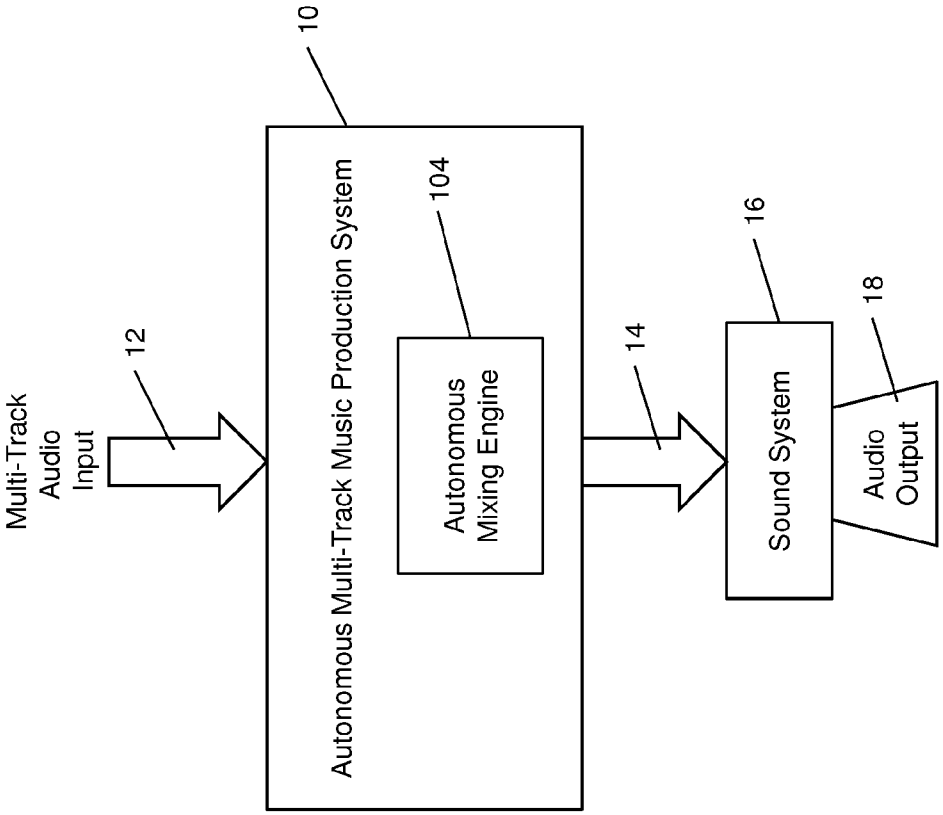

Multi-Track
Audio
Input
12
10
Autonomous Multi-Track Music Production System
104
Autonomous
Mixing Engine
14
16
Sound System
18
Audio
Output

FIG. 1

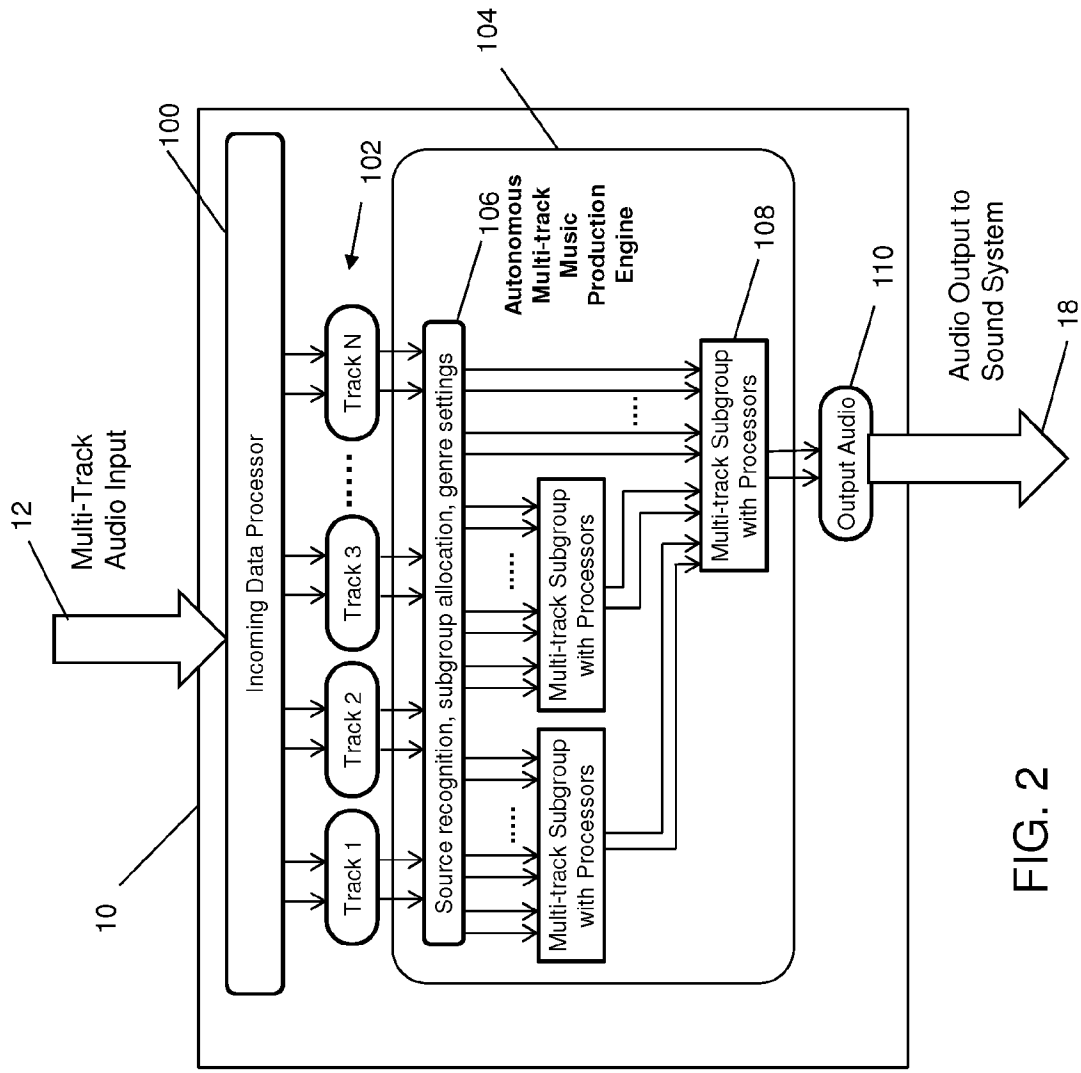

12
Multi-Track
Audio Input
10
100
Incoming Data Processor
Track 1
Track 2
Track 3
Track N
102
104
Source recognition, subgroup allocation, genre settings
106
Autonomous
Multi-track
Music
Production
Engine
Multi-track Subgroup
with Processors
Multi-track Subgroup
with Processors
Multi-track Subgroup
with Processors
108
Output Audio
110
Audio Output to
Sound System
18

FIG. 2

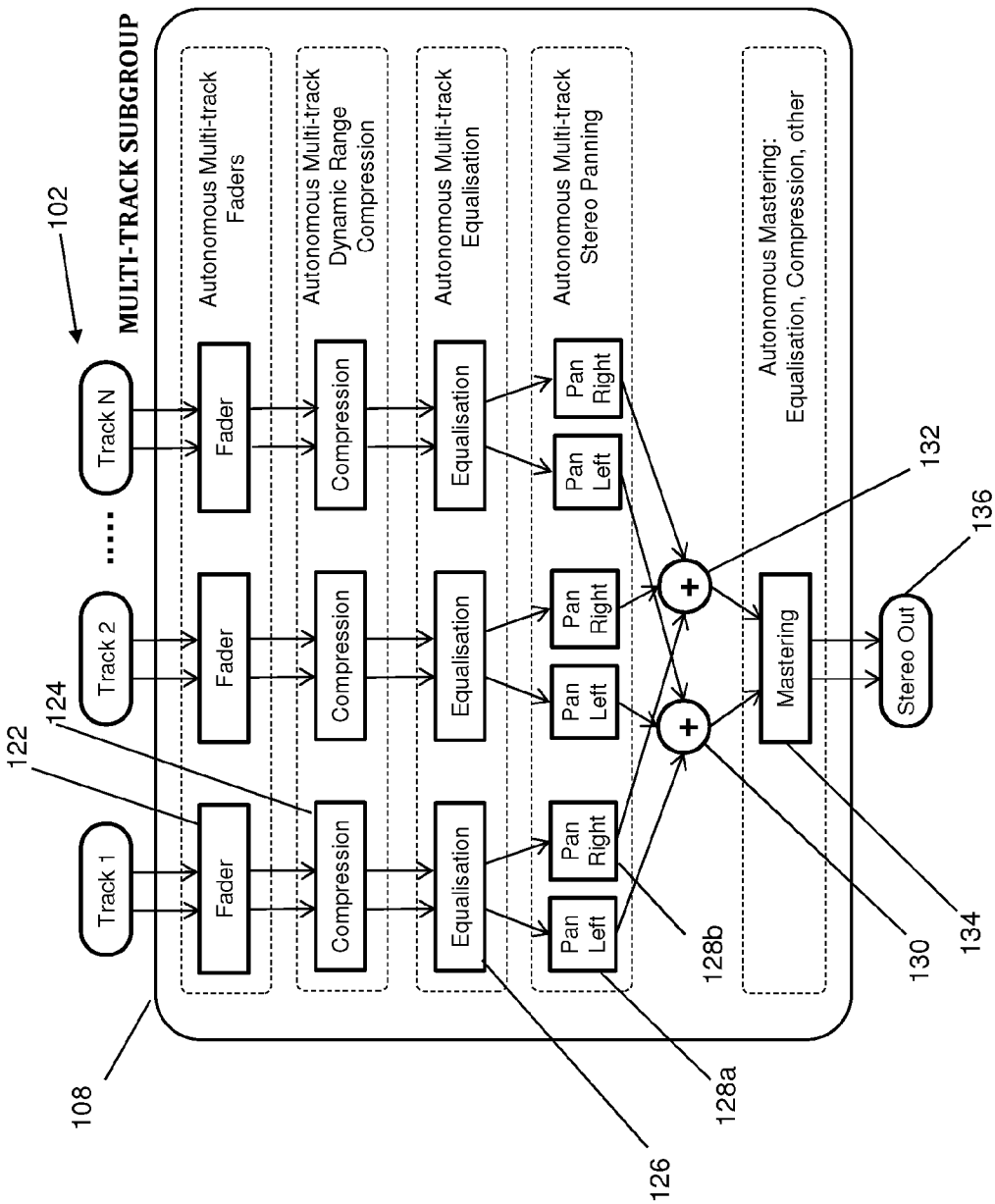

MULTI-TRACK SUBGROUP
102
108
Track 1
Track 2
Track N
Fader
Compression
Equalisation
Pan Left
Pan Right
Autonomous Multi-track Faders
Autonomous Multi-track Dynamic Range Compression
Autonomous Multi-track Equalisation
Autonomous Multi-track Stereo Panning
Autonomous Mastering: Equalisation, Compression, other
Mastering
Stereo Out
122
124
126
128a
128b
130
132
134
136

FIG. 3

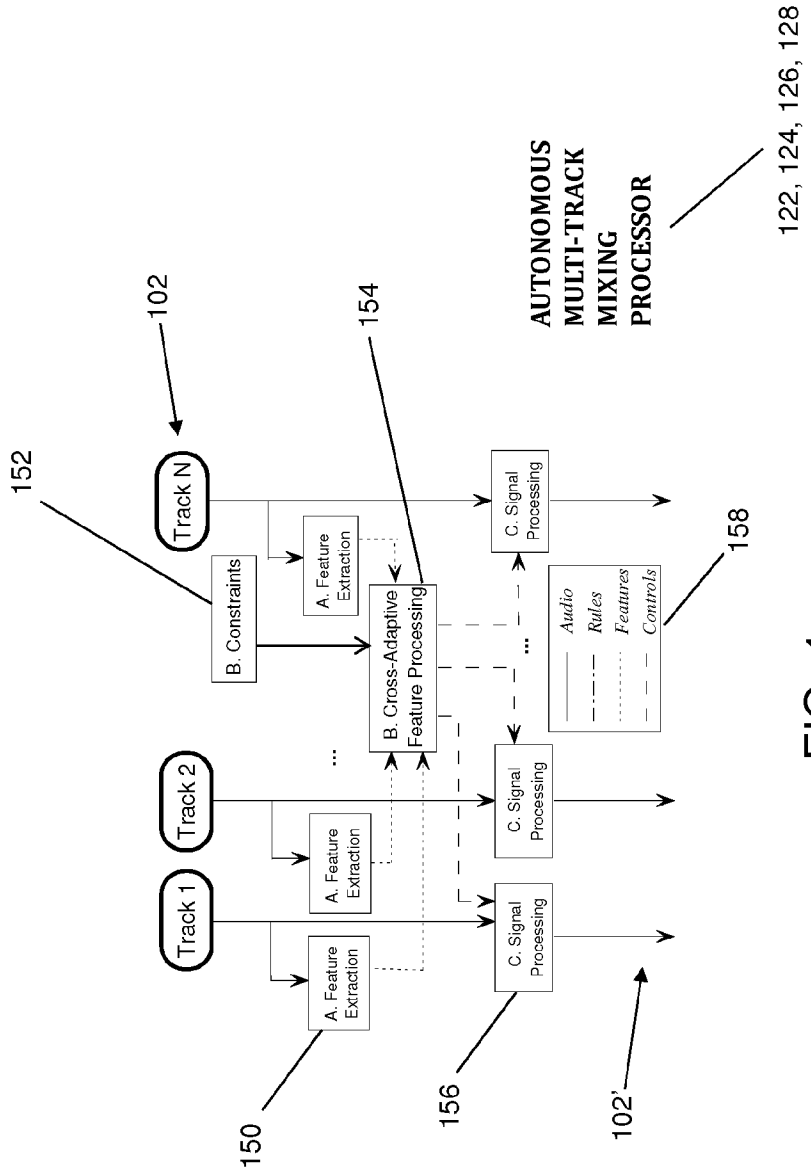

AUTONOMOUS
MULTI-TRACK
MIXING
PROCESSOR
122, 124, 126, 128
102
154
152
Track N
Track 2
Track 1
B. Constraints
A. Feature Extraction
A. Feature Extraction
A. Feature Extraction
B. Cross-Adaptive Feature Processing
C. Signal Processing
C. Signal Processing
C. Signal Processing
Audio
Rules
Features
Controls
158
150
156
102'

FIG. 4

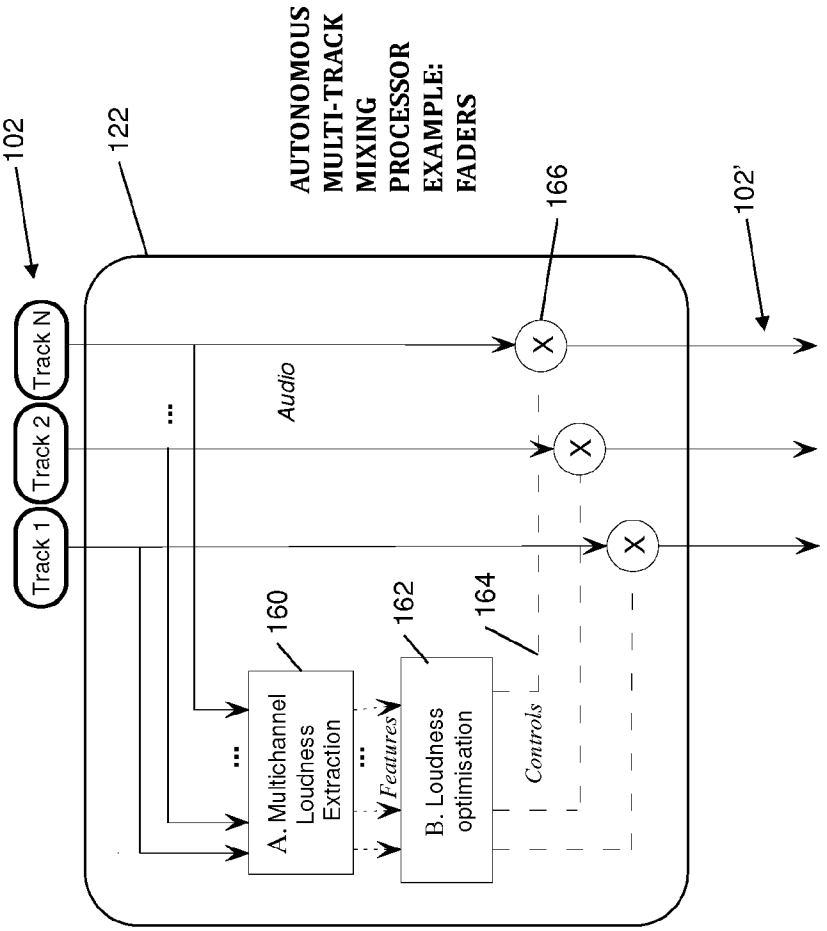

AUTONOMOUS MULTI-TRACK MIXING PROCESSOR EXAMPLE: FADERS
102
122
166
102'
Track 1
Track 2
Track N
Audio
A. Multichannel Loudness Extraction
160
Features
162
B. Loudness optimisation
164
Controls

FIG. 5

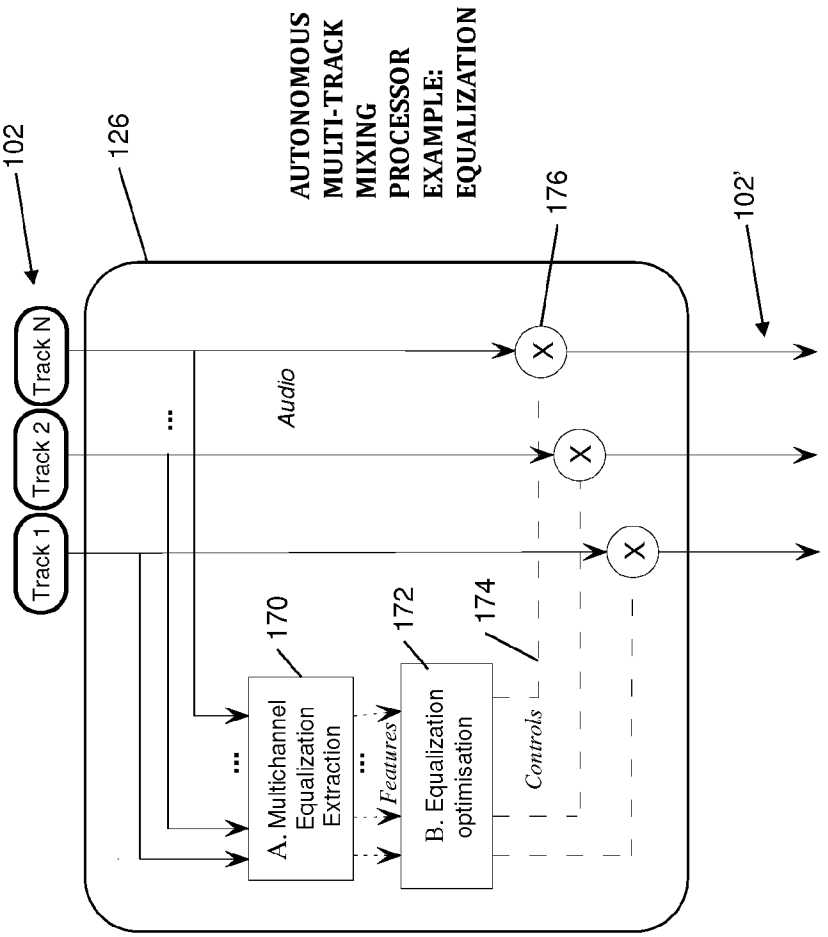

AUTONOMOUS MULTI-TRACK MIXING PROCESSOR EXAMPLE: EQUALIZATION
102
126
176
102'
Track 1
Track 2
Track N
Audio
A. Multichannel Equalization Extraction
170
Features
B. Equalization optimisation
172
174
Controls

FIG. 6

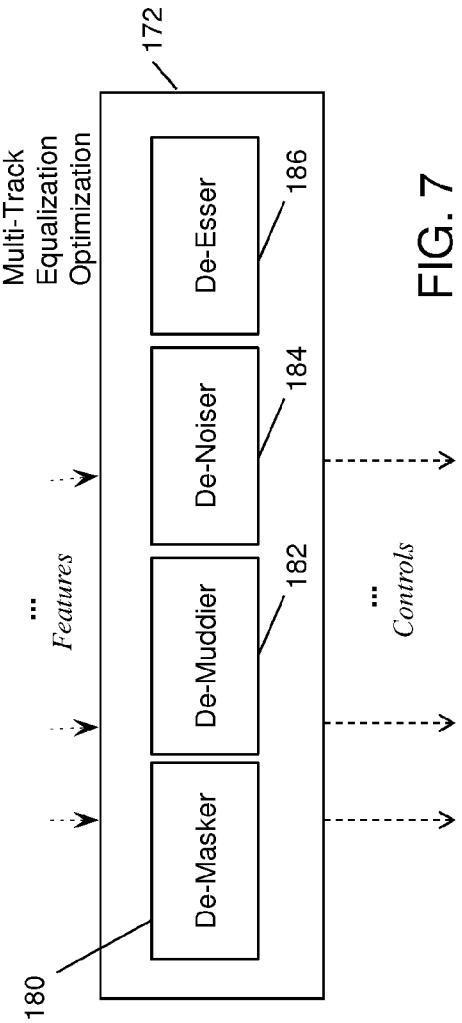

Multi-Track
Equalization
Optimization
172
De-Masker
De-Muddier
De-Noiser
De-Esser
180
182
184
186
...
Features
...
Controls

FIG. 7

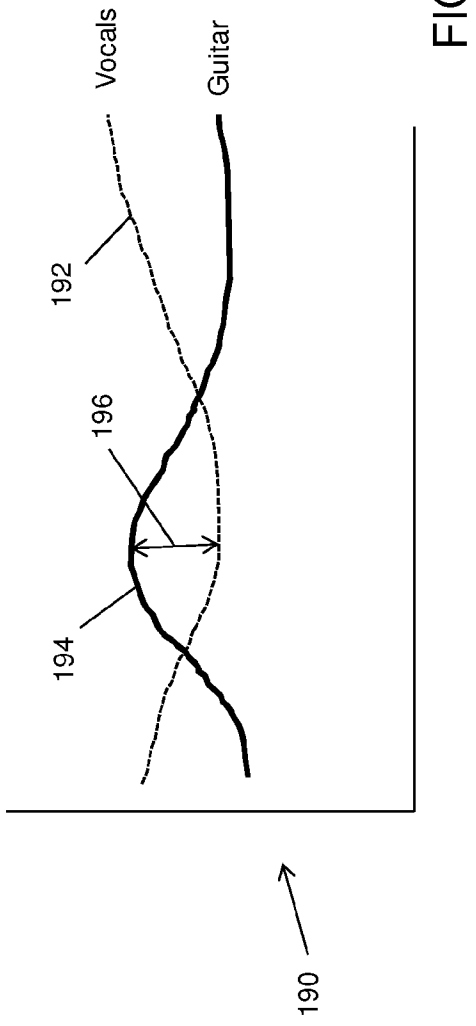

Vocals
Guitar
192
196
194
190

FIG. 8

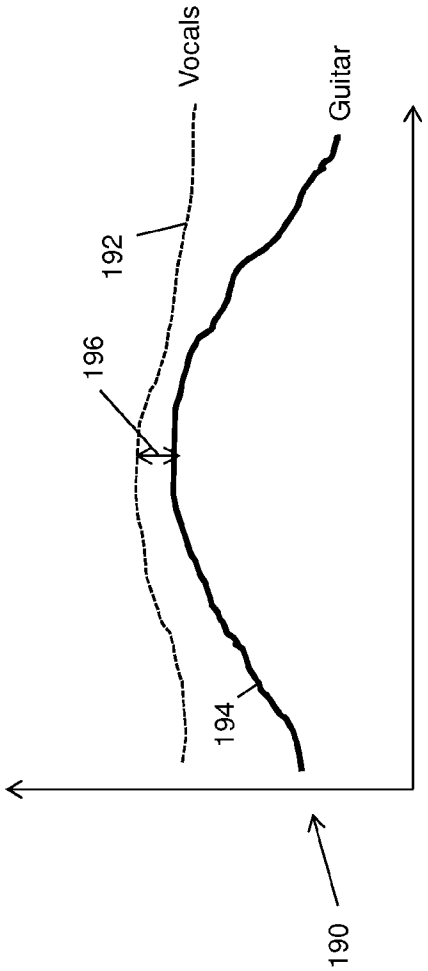


FIG. 9

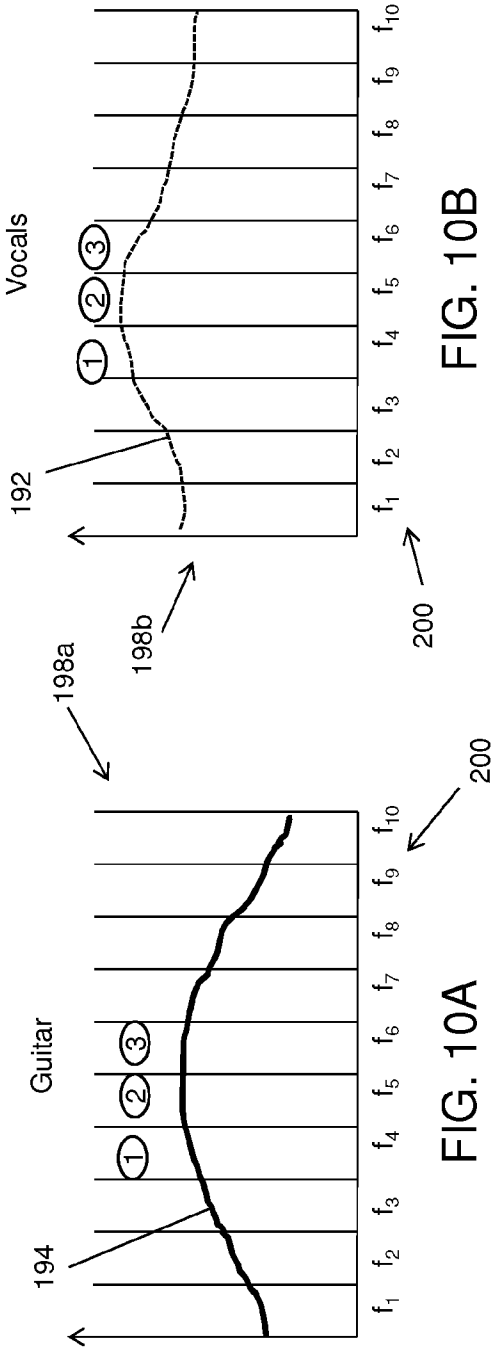


FIG. 10B

FIG. 10A

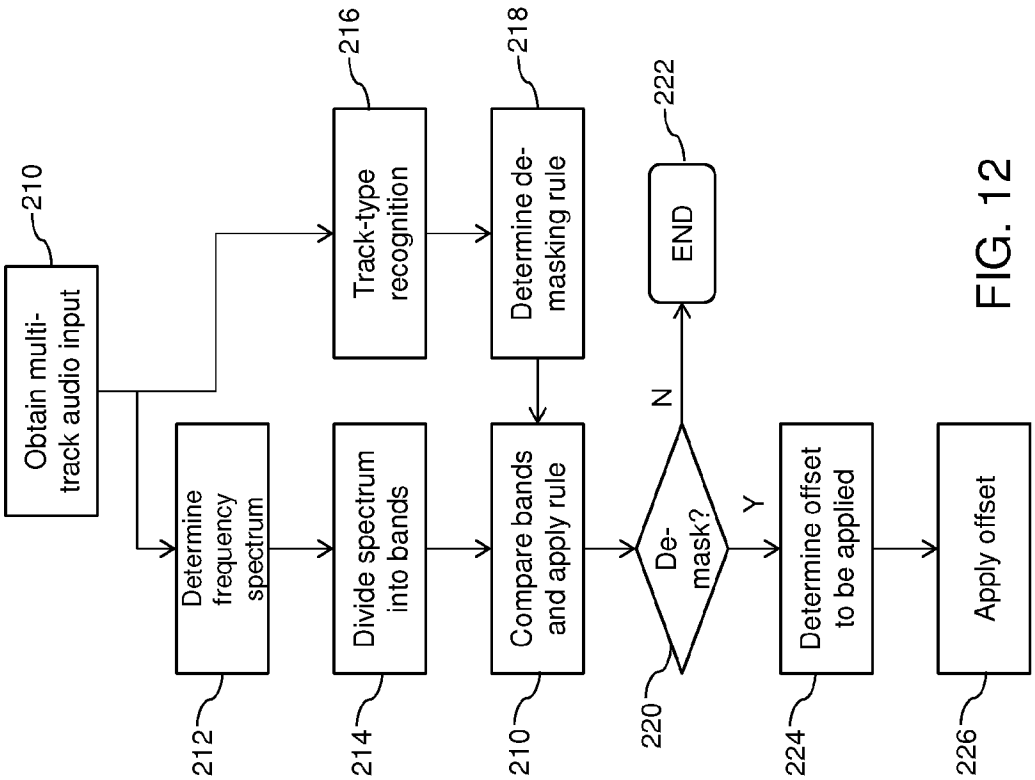

Obtain multi-track audio input
210
Determine frequency spectrum
212
Track-type recognition
216
Divide spectrum into bands
214
Determine de-masking rule
218
Compare bands and apply rule
210
De-mask?
220
N
END
222
Y
Determine offset to be applied
224
Apply offset
226

FIG. 12

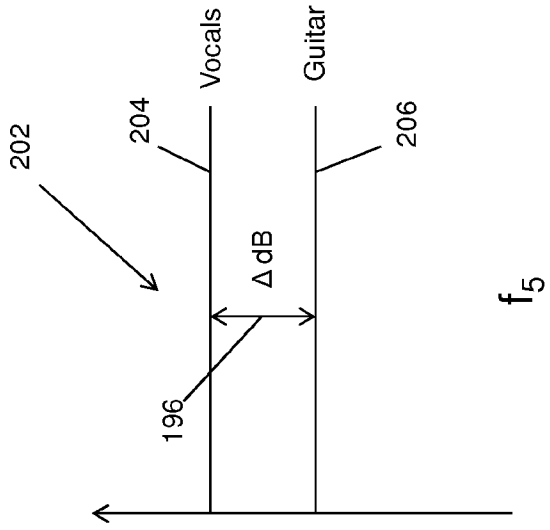

202
204
Vocals
206
Guitar
Δ dB
196
f_5

FIG. 11

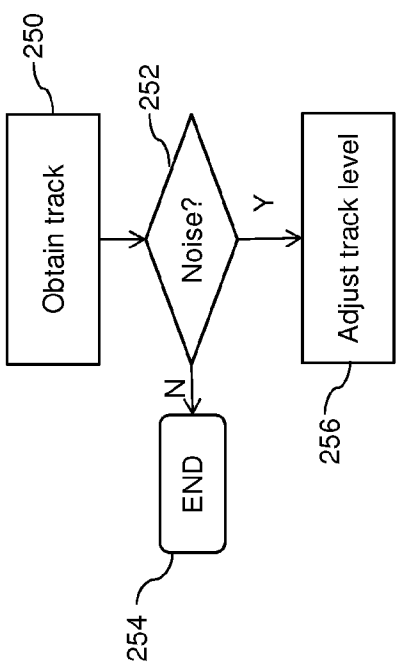


FIG. 14

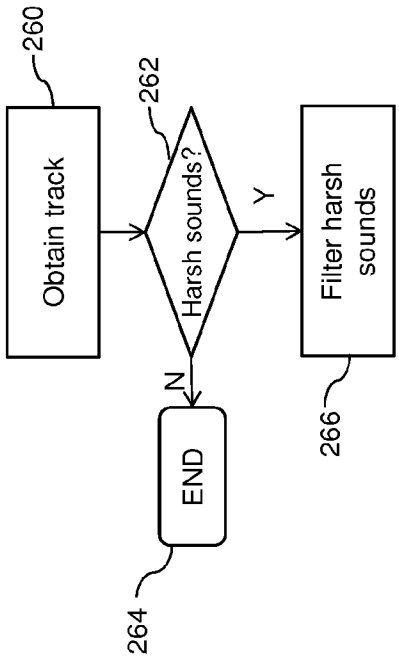


FIG. 15A

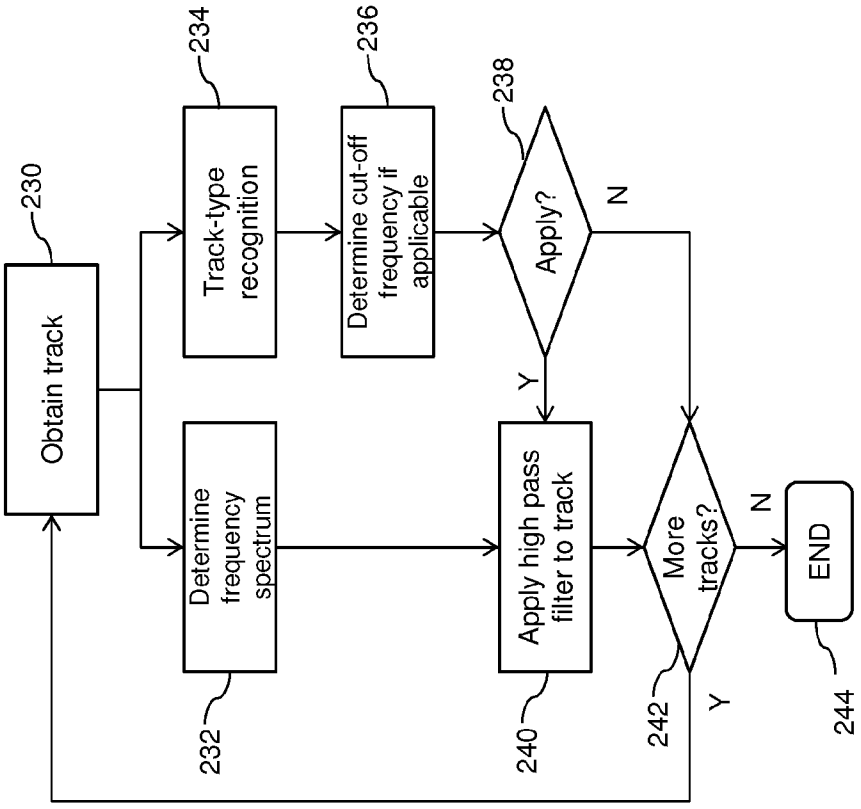


FIG. 13

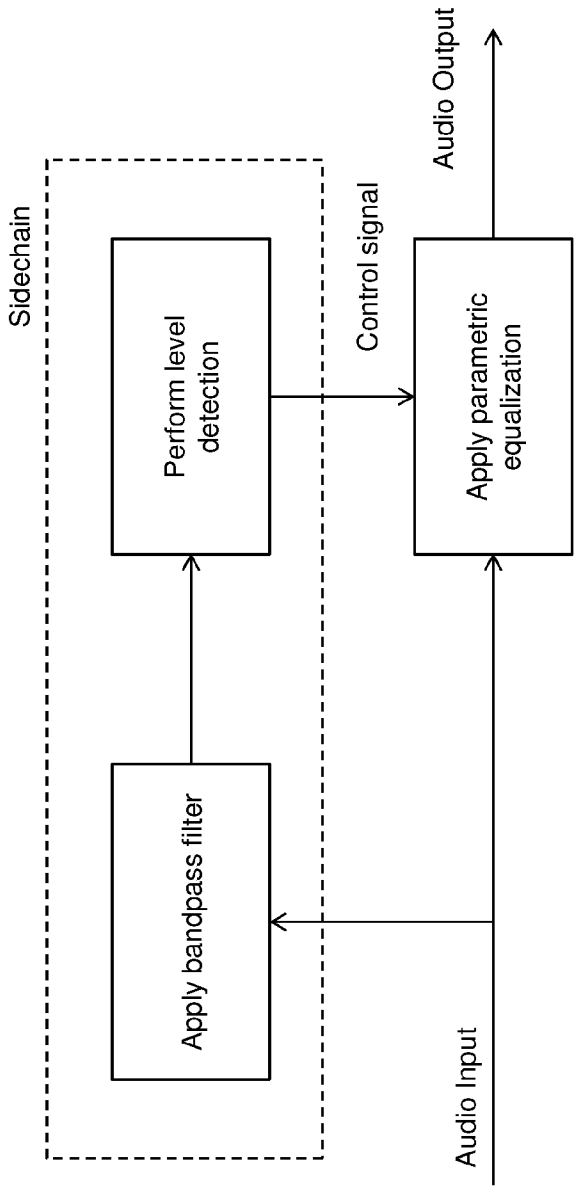

Sidechain
Perform level detection
Apply bandpass filter
Control signal
Apply parametric equalization
Audio Output
Audio Input

FIG. 15B

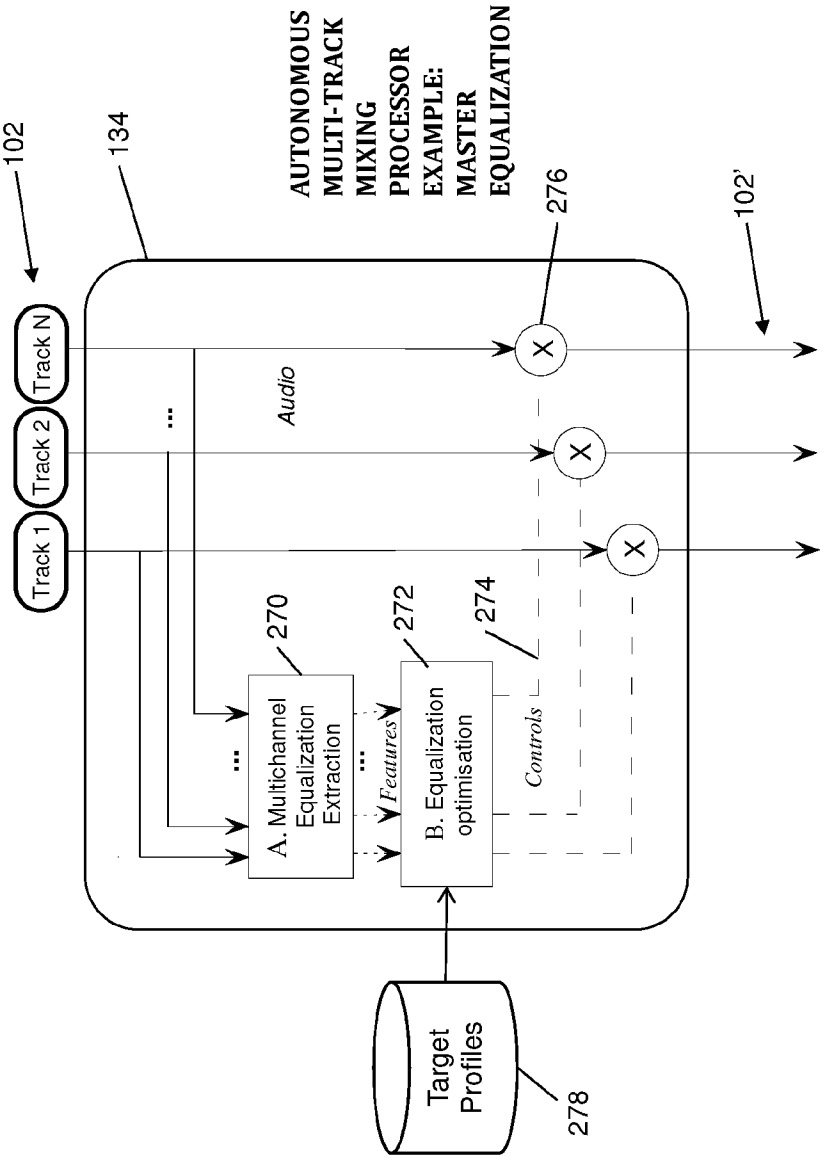

AUTONOMOUS MULTI-TRACK MIXING PROCESSOR EXAMPLE: MASTER EQUALIZATION
102
134
276
102'
Track 1
Track 2
Track N
Audio
A. Multichannel Equalization Extraction
270
Features
B. Equalization optimisation
272
274
Controls
Target Profiles
278

FIG. 16

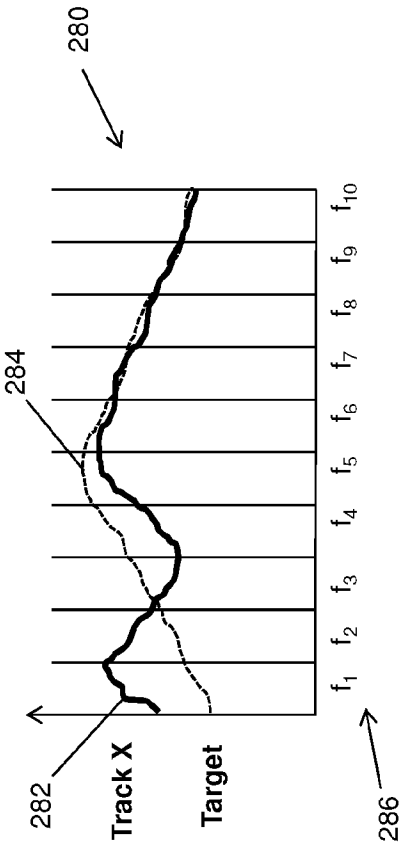


FIG. 17

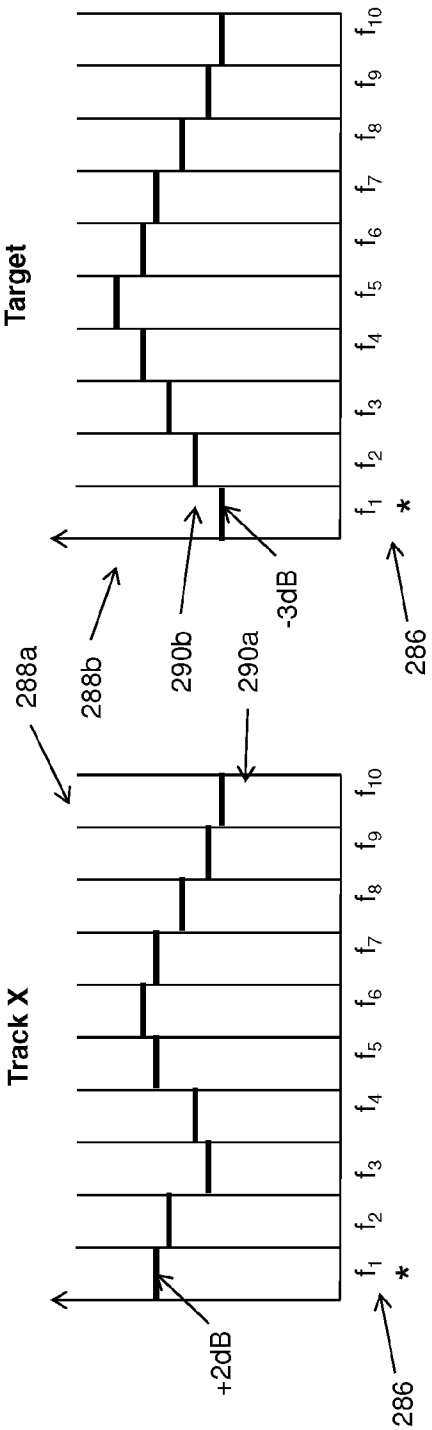


FIG. 18

FIG. 19

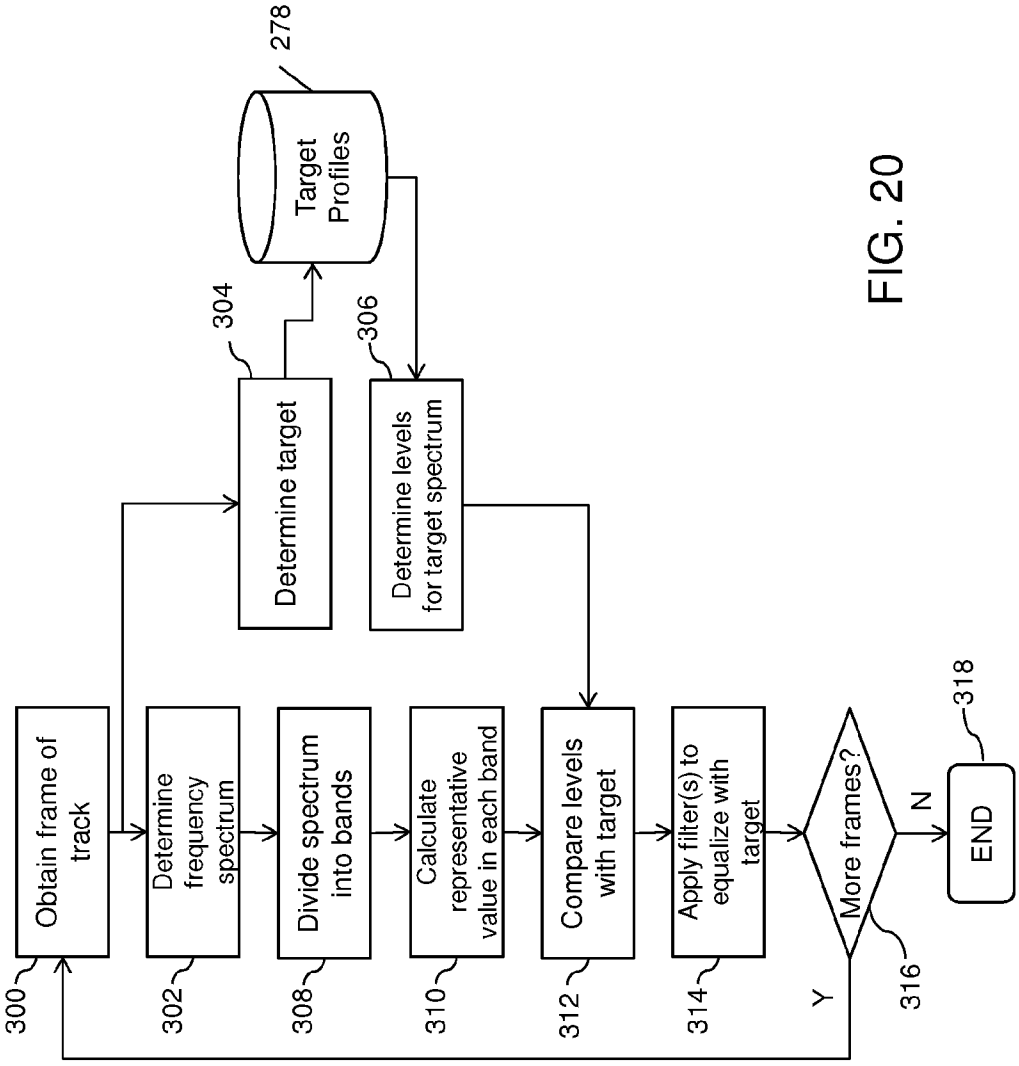

278
Target Profiles
304
306
Determine target
Determine levels for target spectrum
Obtain frame of track
Determine frequency spectrum
Divide spectrum into bands
Calculate representative value in each band
Compare levels with target
Apply filter(s) to equalize with target
More frames?
N
Y
END
318
300
302
308
310
312
314
316

FIG. 20

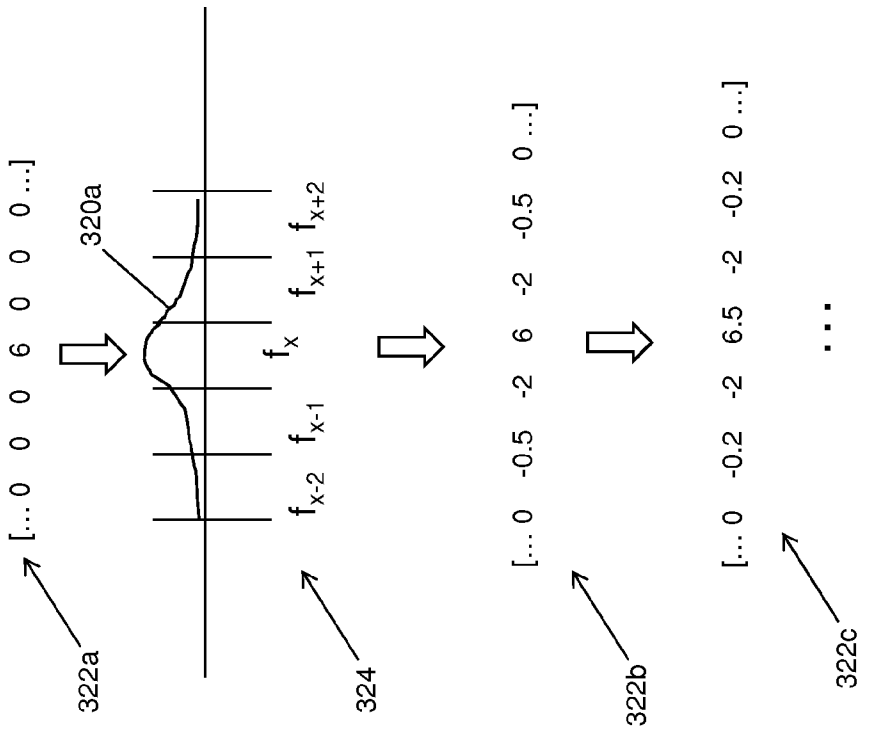

[... 0 0 0 0 6 0 0 0 0 ...]
322a
320a
f_{x-2} f_{x-1} f_x f_{x+1} f_{x+2}
324
[... 0 -0.5 -2 6 -2 -0.5 0 ...]
322b
[... 0 -0.2 -2 6.5 -2 -0.2 0 ...]
322c
...

FIG. 22

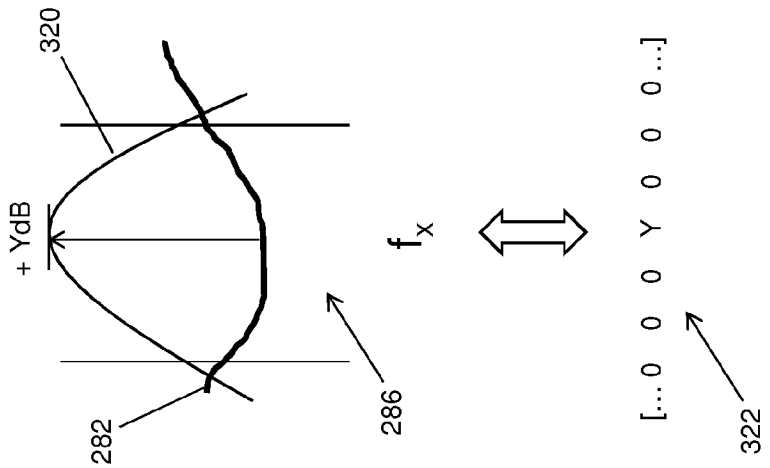

320
+ YdB
282
286
f_x
[... 0 0 0 0 Y 0 0 0 0 ...]
322

FIG. 21

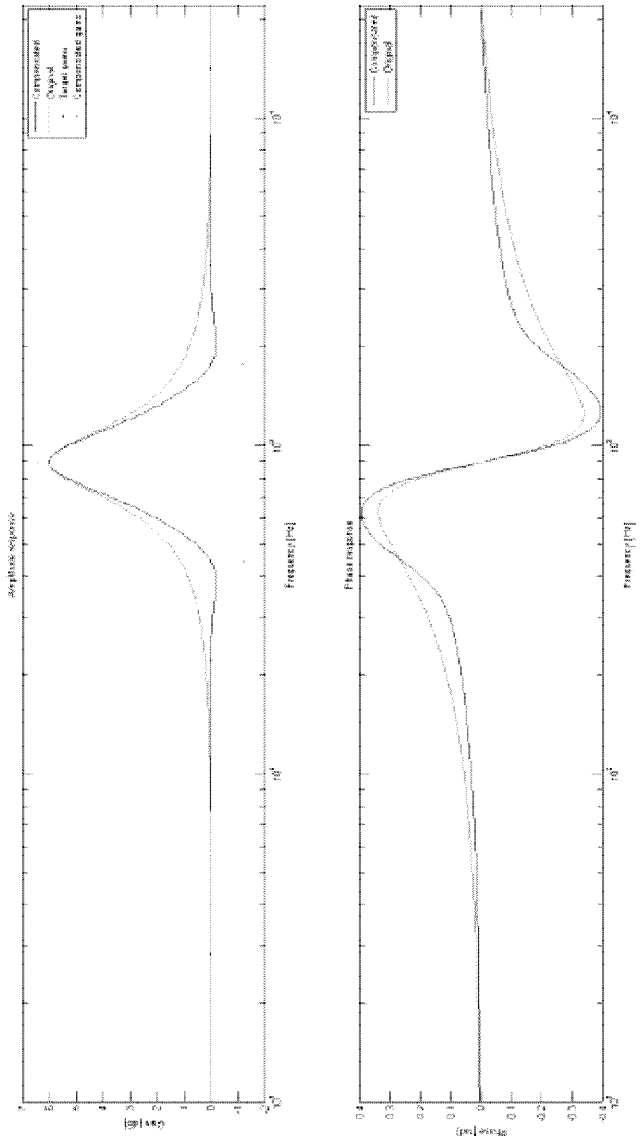


FIG. 23A

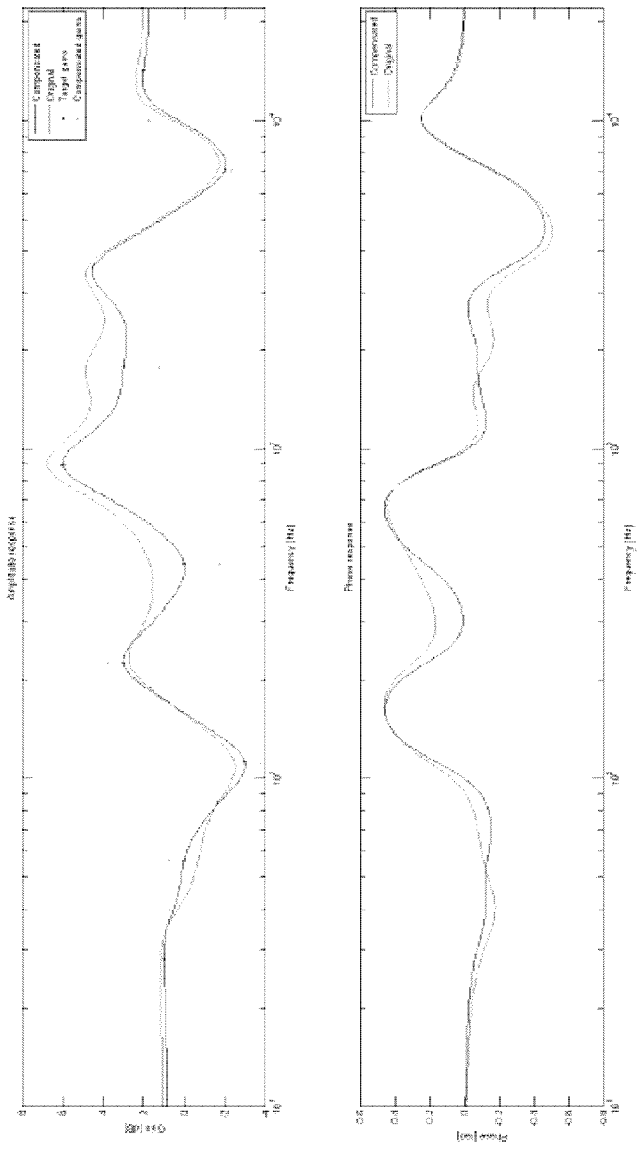
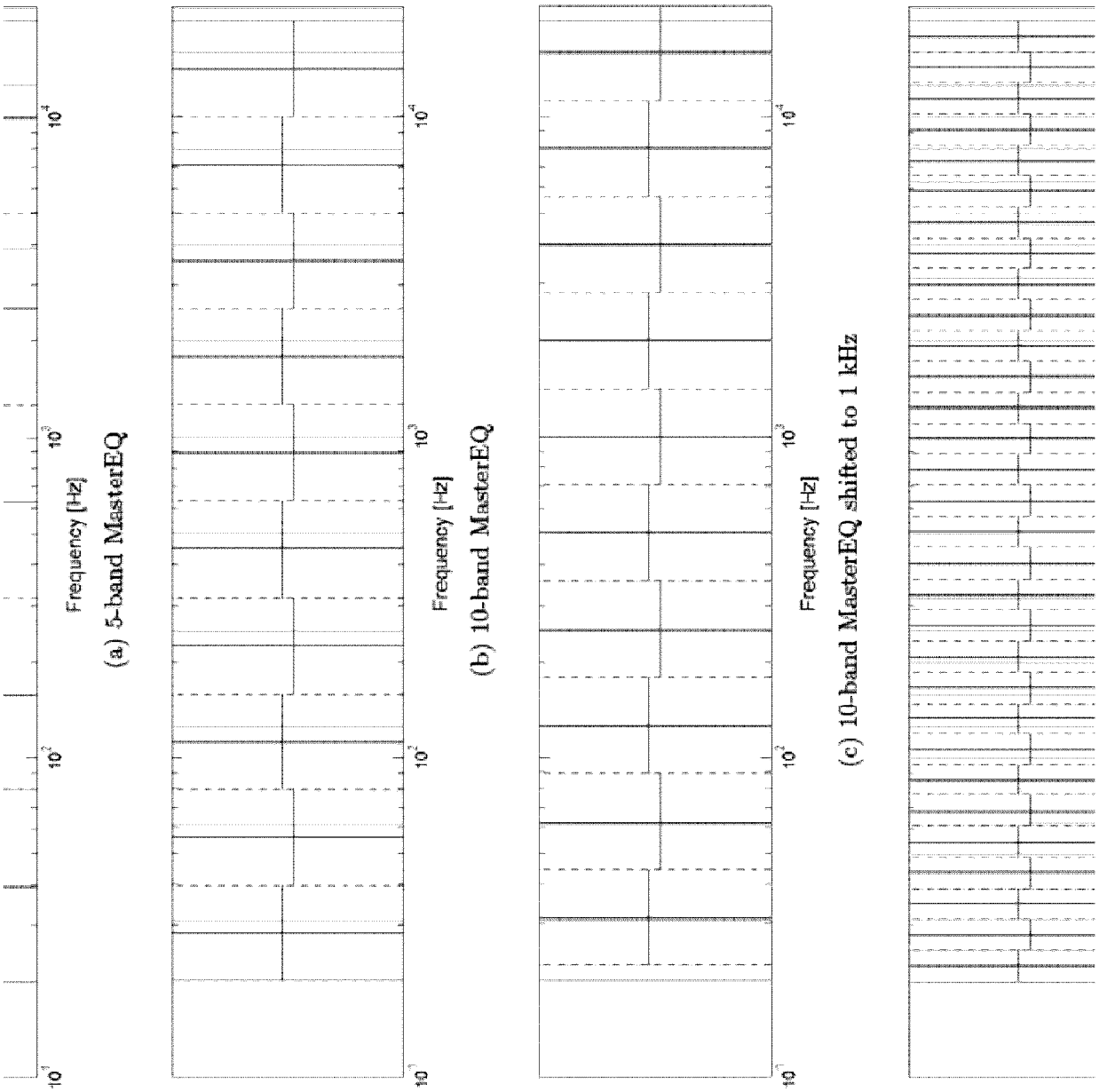


FIG. 23B


Frequency [Hz]
(a) 5-band MasterEQ
Frequency [Hz]
(b) 10-band MasterEQ
Frequency [Hz]
(c) 10-band MasterEQ shifted to 1 kHz

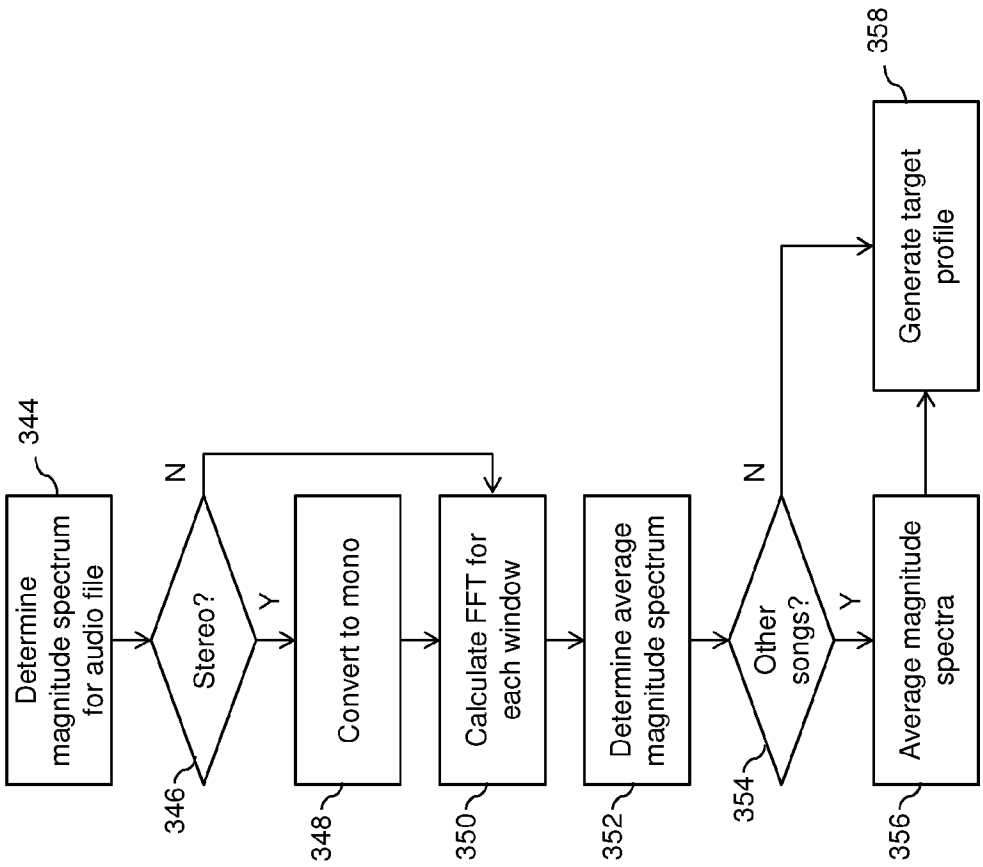

344
Determine magnitude spectrum for audio file
Stereo?
N
Y
346
Convert to mono
348
Calculate FFT for each window
350
Determine average magnitude spectrum
352
Other songs?
N
Y
354
Average magnitude spectra
356
Generate target profile
358

FIG. 26

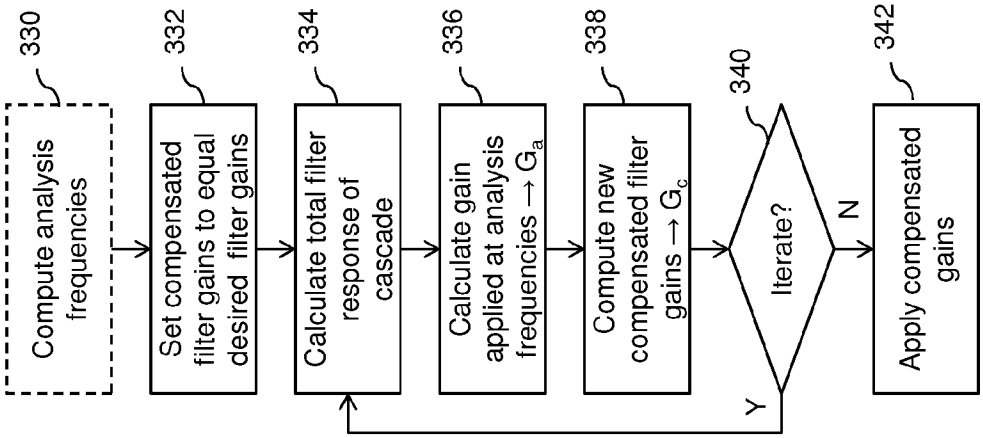

330
Compute analysis frequencies
332
Set compensated filter gains to equal desired filter gains
334
Calculate total filter response of cascade
336
Calculate gain applied at analysis frequencies → G_a
338
Compute new compensated filter gains → G_c
340
Iterate?
Y
N
342
Apply compensated gains

FIG. 25

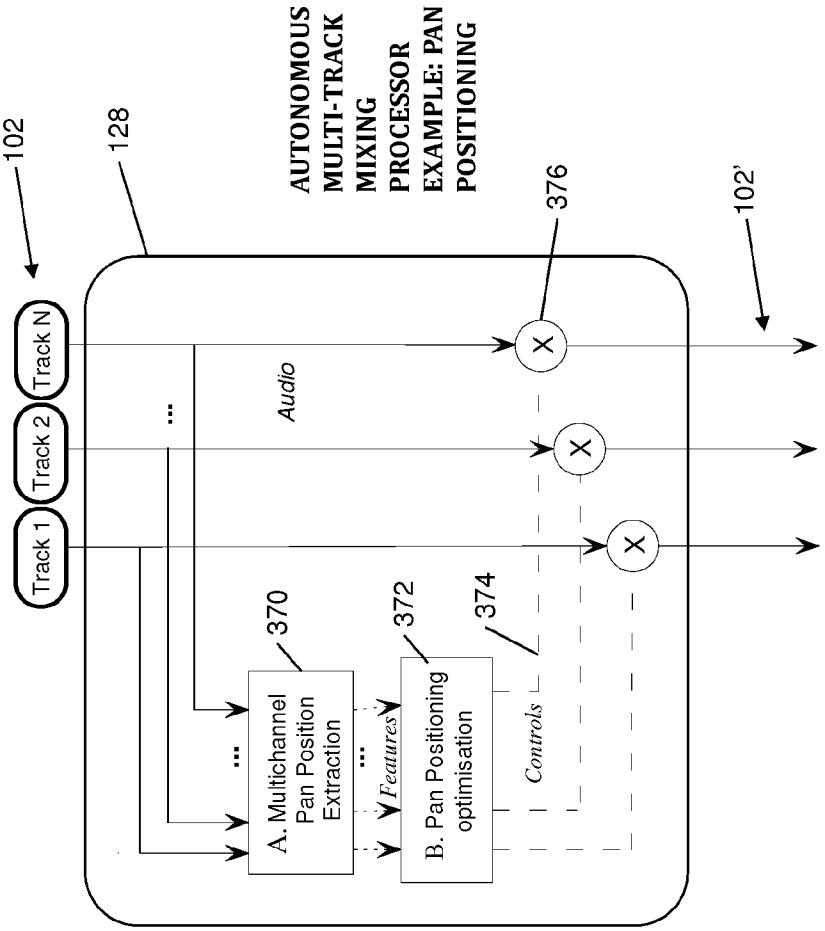

AUTONOMOUS MULTI-TRACK MIXING PROCESSOR EXAMPLE: PAN POSITIONING
102
128
376
102'
Track 1
Track 2
Track N
Audio
A. Multichannel Pan Position Extraction
370
Features
372
B. Pan Positioning optimisation
374
Controls

FIG. 27

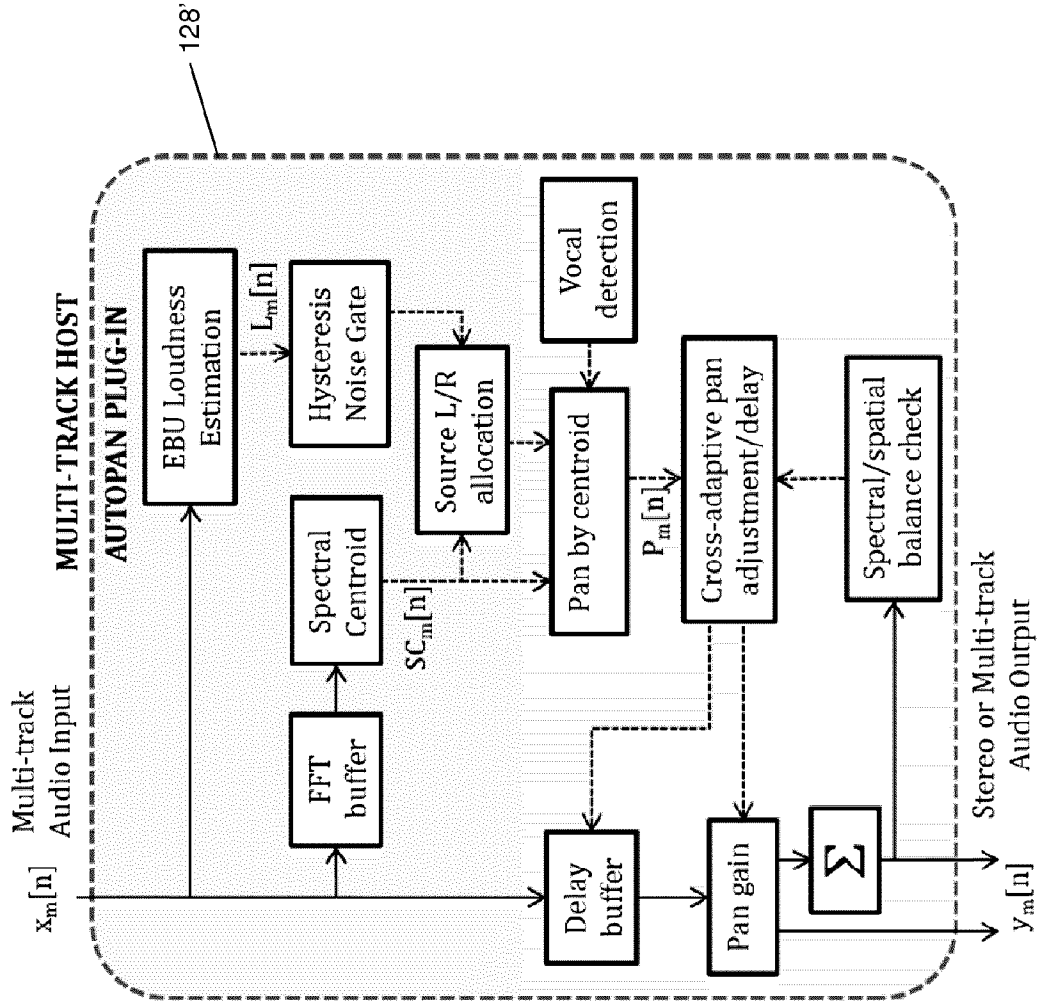

128'
MULTI-TRACK HOST
AUTOPAN PLUG-IN
Multi-track
Audio Input
$x_m[n]$
EBU Loudness Estimation
$L_m[n]$
Hysteresis Noise Gate
Spectral Centroid
$SC_m[n]$
FFT buffer
Source L/R allocation
Vocal detection
Pan by centroid
$P_m[n]$
Cross-adaptive pan adjustment/delay
Spectral/spatial balance check
Delay buffer
Pan gain
Σ
Stereo or Multi-track
Audio Output
$y_m[n]$

FIG. 28

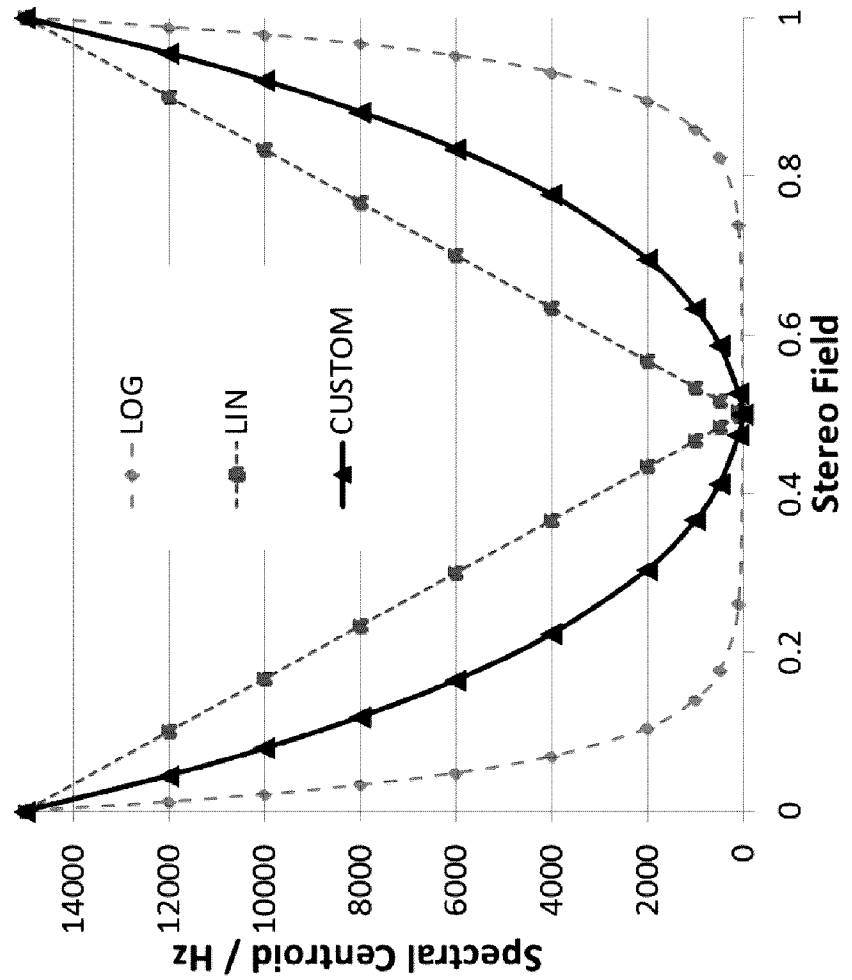

Spectral Centroid / Hz
14000
12000
10000
8000
6000
4000
2000
0
0
0.2
0.4
0.6
0.8
1
Stereo Field
LOG
LIN
CUSTOM

FIG. 29

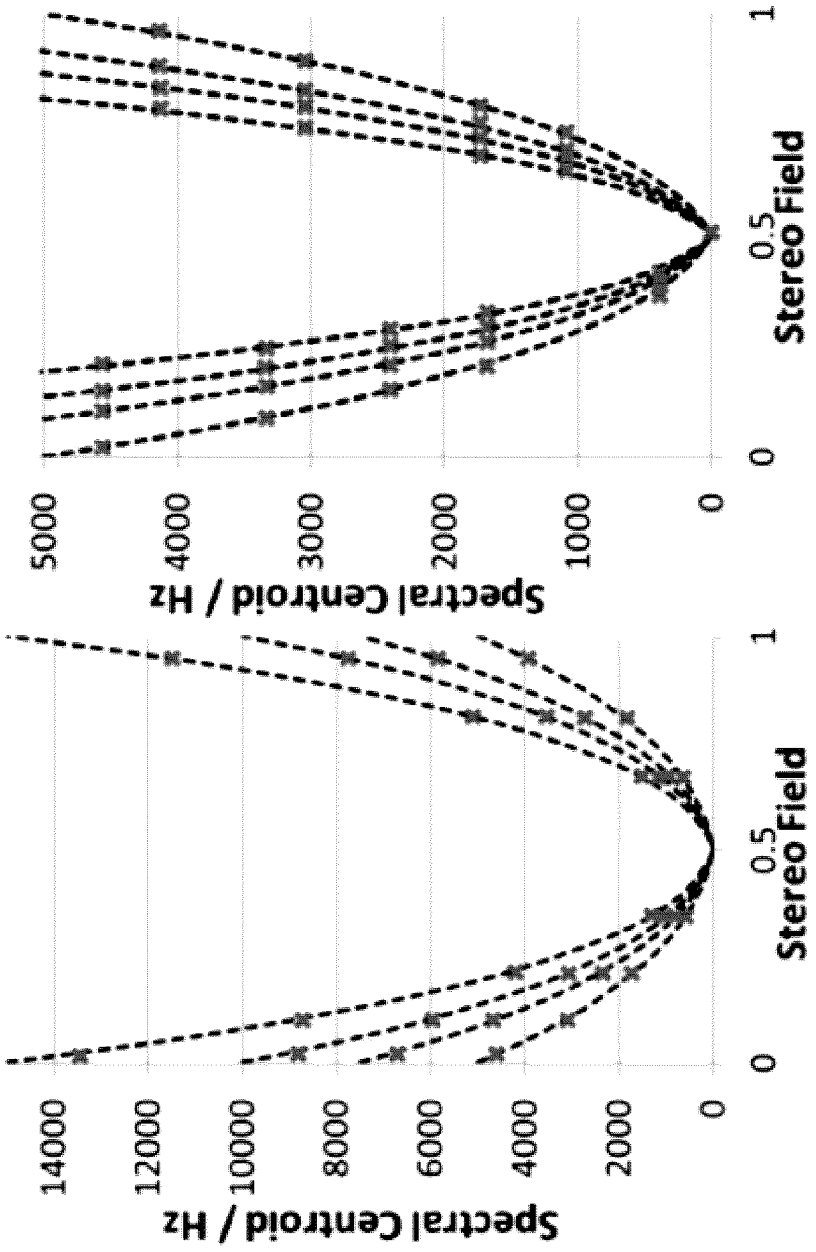

Spectral Centroid / Hz
14000
12000
10000
8000
6000
4000
2000
0
0
0.5
1
Stereo Field
Spectral Centroid / Hz
5000
4000
3000
2000
1000
0
0
0.5
1
Stereo Field

FIG. 30

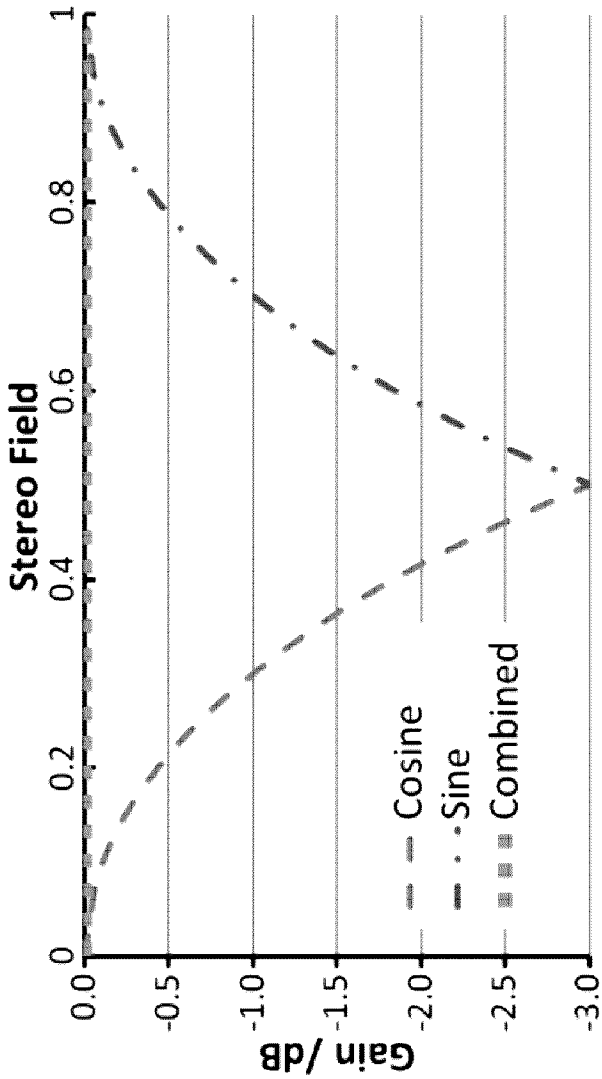

Stereo Field
0
0.2
0.4
0.6
0.8
1
0.0
-0.5
-1.0
-1.5
-2.0
-2.5
-3.0
Gain /dB
Cosine
Sine
Combined

FIG. 31

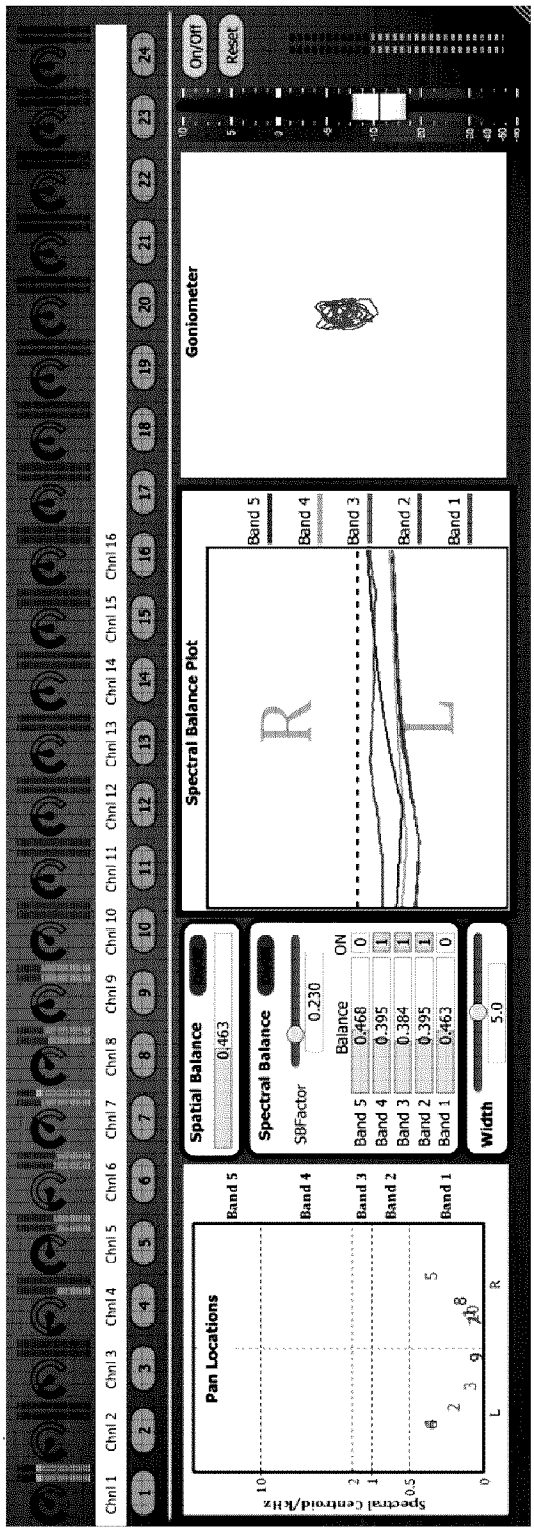
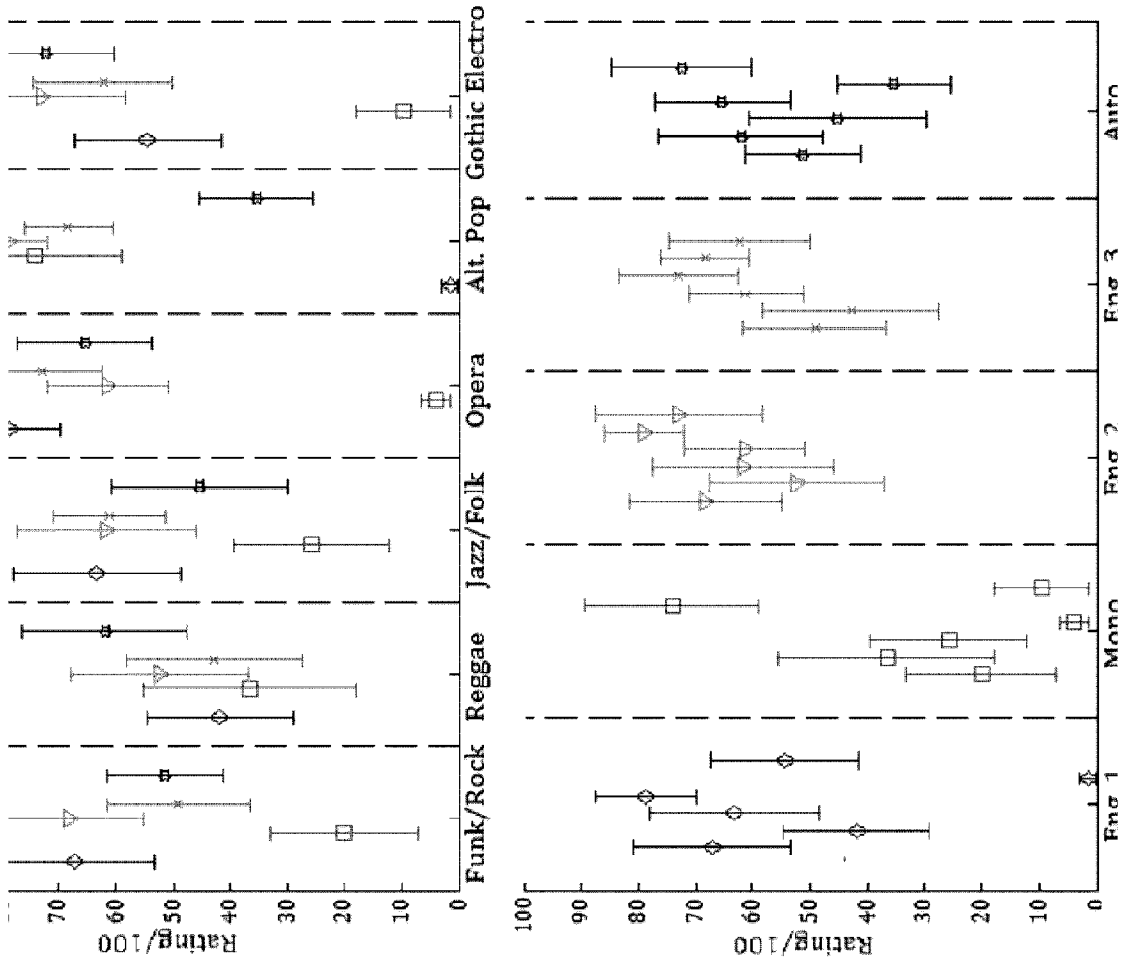
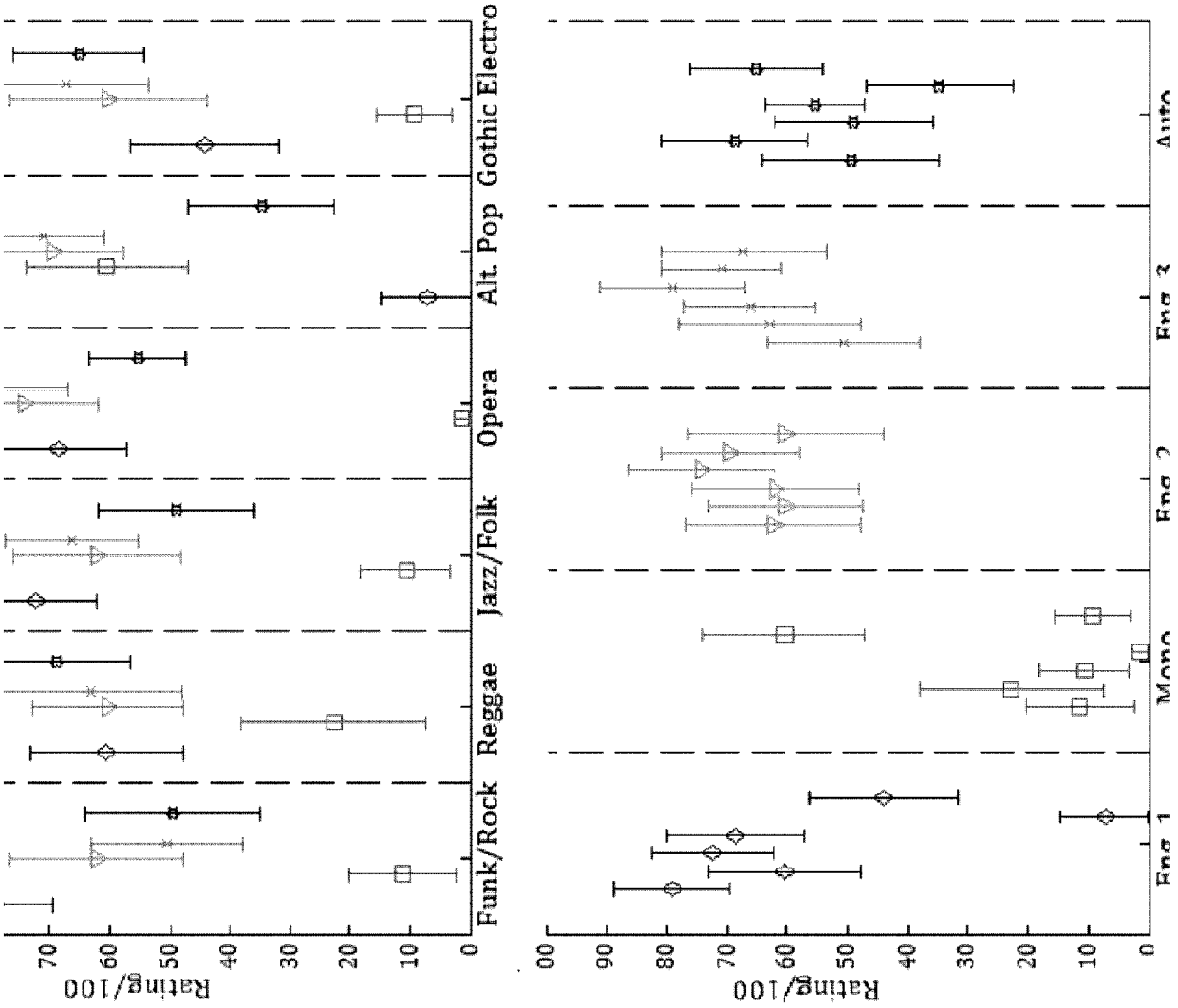
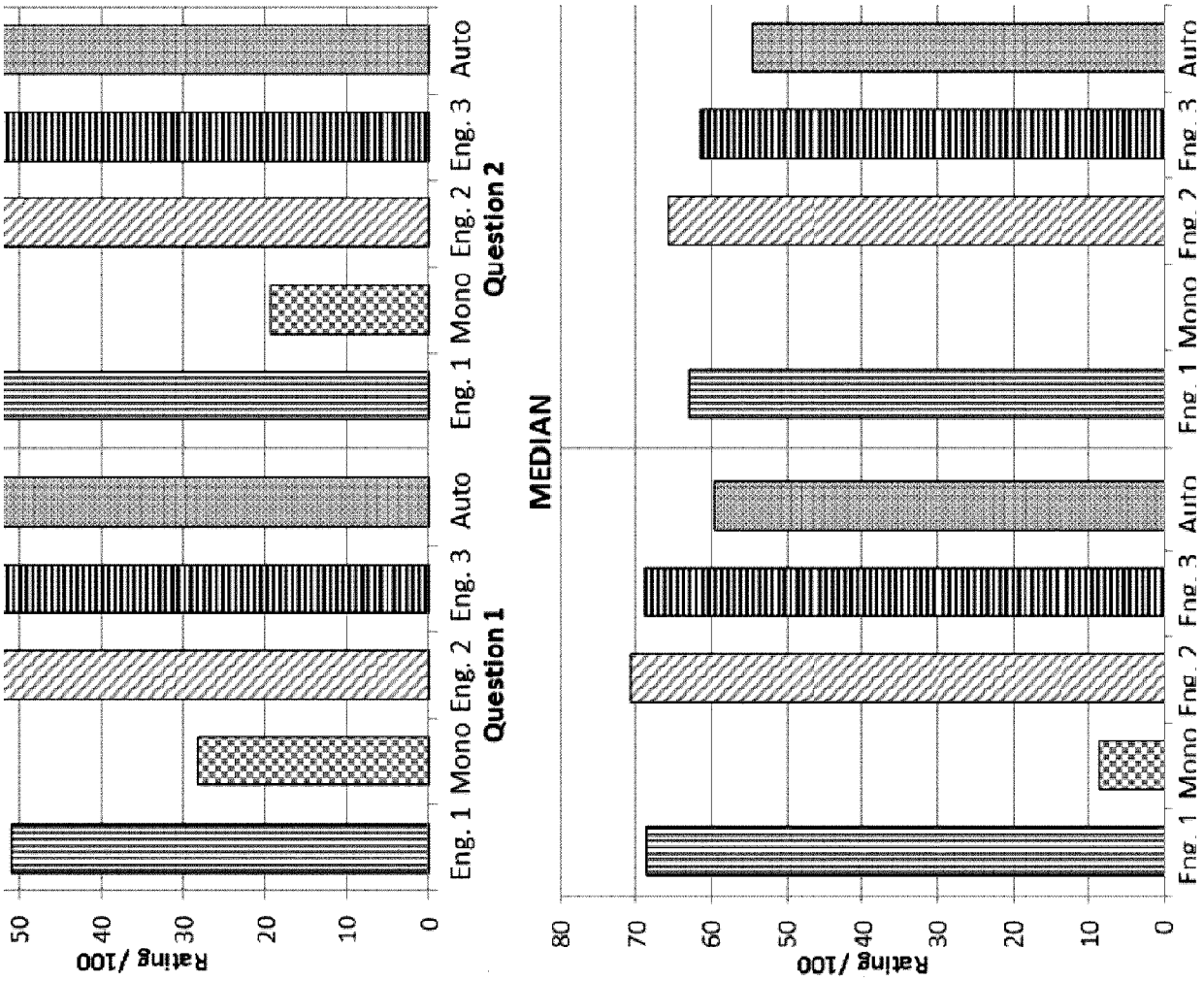

Chnl 1
Chnl 2
Chnl 3
Chnl 4
Chnl 5
Chnl 6
Chnl 7
Chnl 8
Chnl 9
Chnl 10
Chnl 11
Chnl 12
Chnl 13
Chnl 14
Chnl 15
Chnl 16
Pan Locations
Band 5
Band 4
Band 3
Band 2
Band 1
Spectral Centroid/kHz
L
R
Spatial Balance
0.463
Spectral Balance
SBFactor
0.230
Balance
ON
Band 5 0.468 0
Band 4 0.395 1
Band 3 0.384 1
Band 2 0.395 1
Band 1 0.463 0
Width
5.0
Spectral Balance Plot
Goniometer
On/Off
Reset

FIG. 32


Rating/100
0
10
20
30
40
50
60
70
80
90
100
Eng 1
Mono
Eng 2
Eng 3
Auto
Funk/Rock
Reggae
Jazz/Folk
Opera
Alt. Pop
Gothic Electro


Rating/100
0
10
20
30
40
50
60
70
Funk/Rock
Reggae
Jazz/Folk
Opera
Alt. Pop
Gothic Electro
Rating/100
0
10
20
30
40
50
60
70
80
90
00
Eng 1
Eng 2
Eng 3


Rating /100
0
10
20
30
40
50
Eng. 1 Mono
Eng. 2
Eng. 3
Auto
Question 1
Eng. 1 Mono
Eng. 2
Eng. 3
Auto
Question 2
MEDIAN
Rating /100
0
10
20
30
40
50
60
70
80
Eng. 1 Mono
Eng. 2
Eng. 3
Auto
Eng. 1 Mono
Eng. 2
Eng. 3
Auto

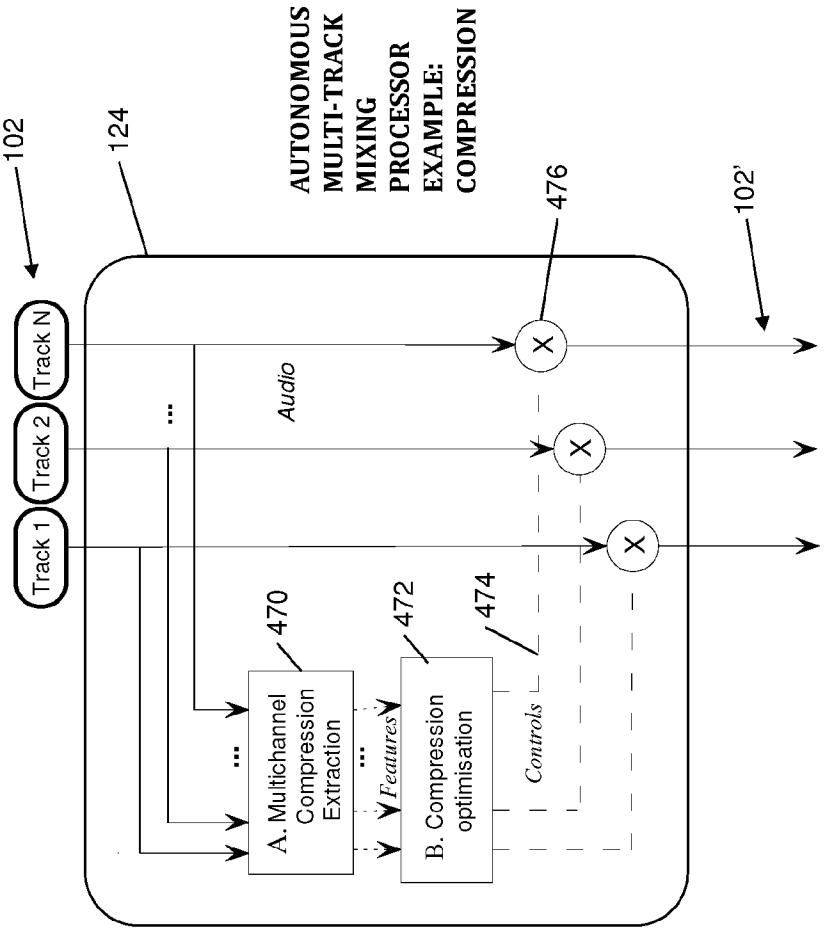

AUTONOMOUS MULTI-TRACK MIXING PROCESSOR EXAMPLE: COMPRESSION
102
124
476
102'
Track 1
Track 2
Track N
Audio
A. Multichannel Compression Extraction
B. Compression optimisation
Features
Controls
470
472
474

FIG. 36

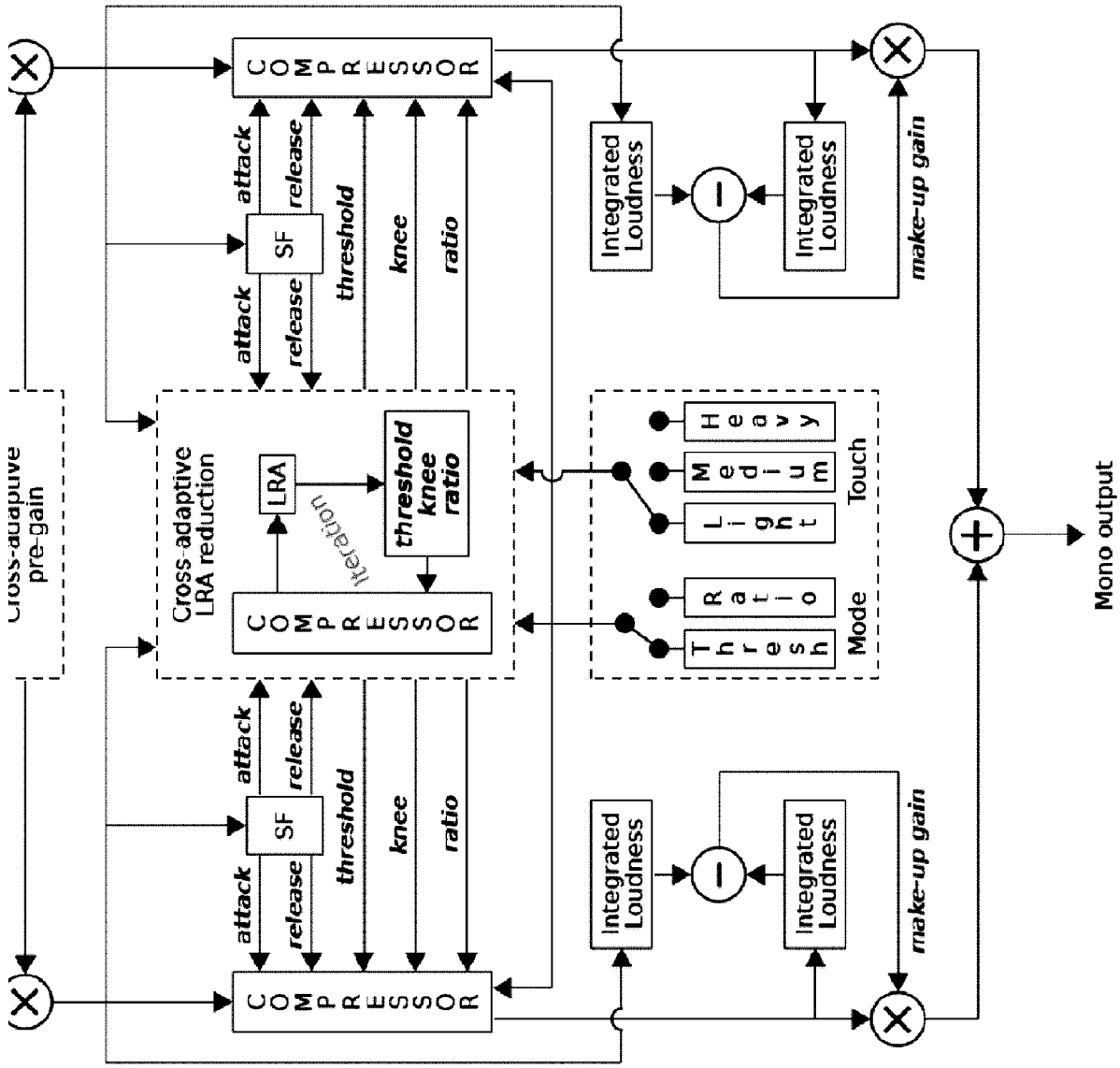

Cross-adaptive pre-gain
Cross-adaptive LRA reduction
COMPRESSOR
SF
attack
release
threshold
knee
ratio
LRA
iteration
threshold knee ratio
Integrated Loudness
make-up gain
Mode
Touch
Threshold
Light
Medium
Heavy
Mono output

FIG. 38A

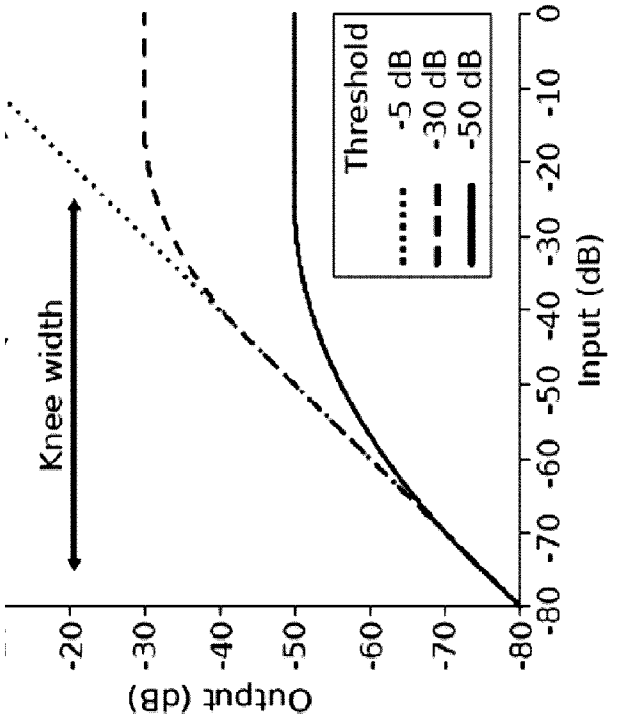


FIG. 38B

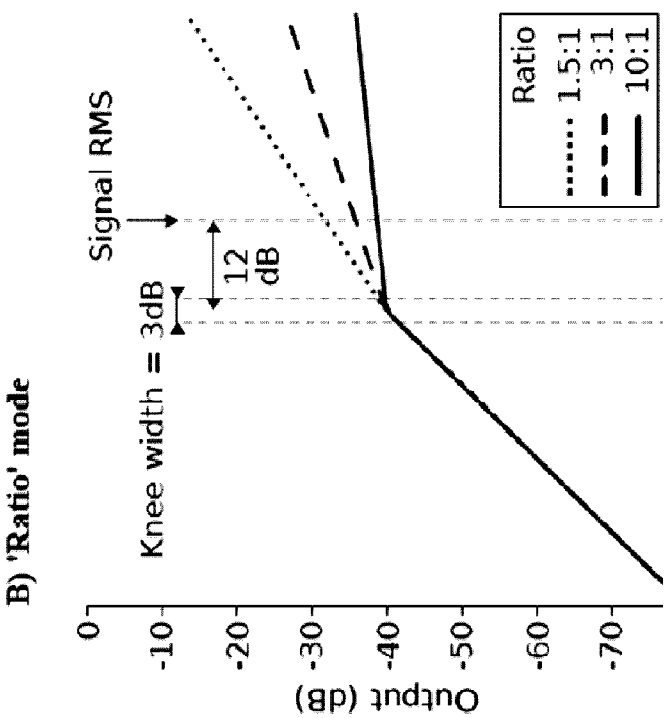


FIG. 39A

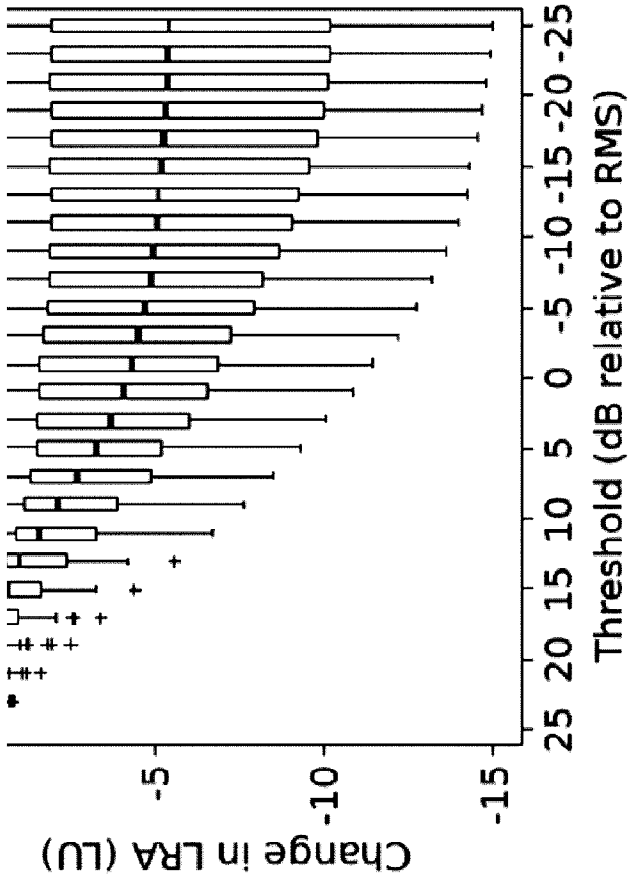
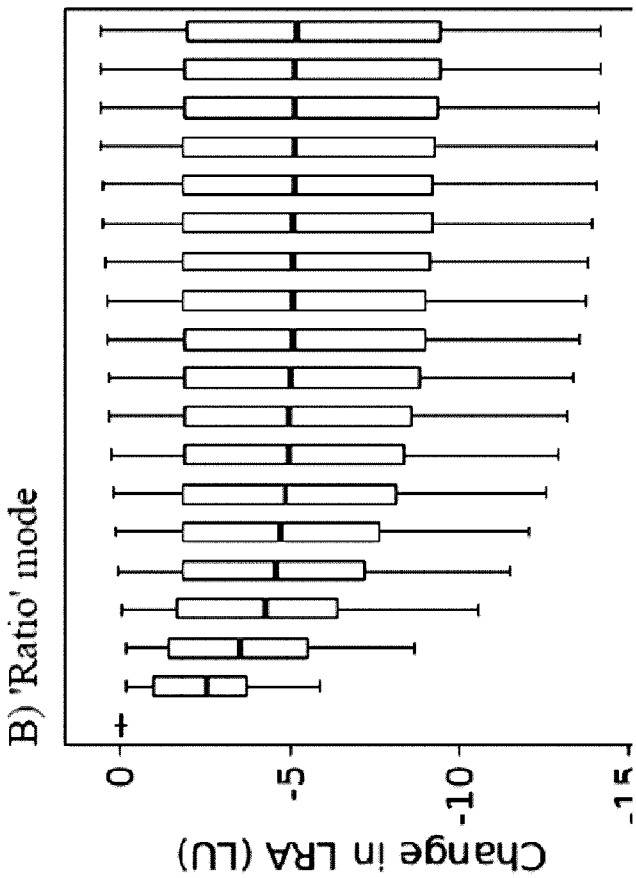


FIG. 39B



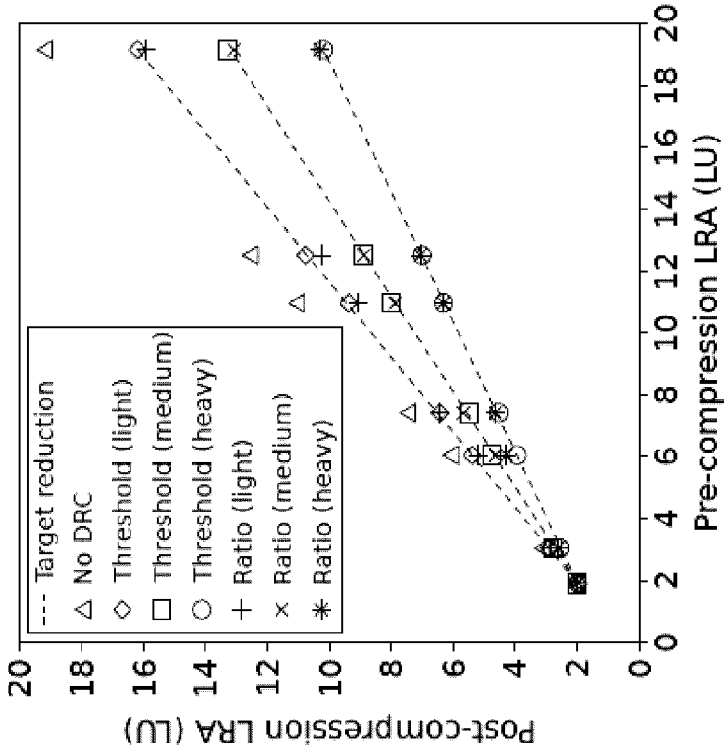

Post-compression LRA (LU)
Pre-compression LRA (LU)
0
2
4
6
8
10
12
14
16
18
20
Target reduction
No DRC
Threshold (light)
Threshold (medium)
Threshold (heavy)
Ratio (light)
Ratio (medium)
Ratio (heavy)

FIG. 40

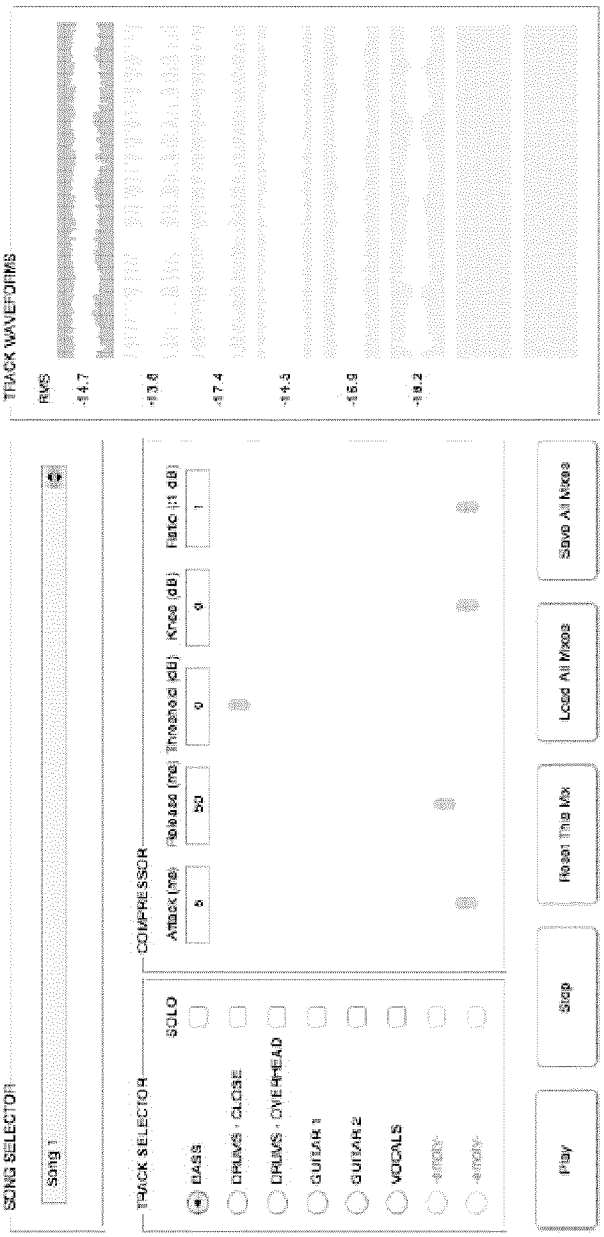

SONG SELECTOR
Song 1
TRACK SELECTOR
SOLO
BASS
DRUMS - CLOSE
DRUMS - OVERHEAD
GUITAR 1
GUITAR 2
VOCALS
empty
empty
COMPRESSOR
Attack (ms)
Release (ms)
Threshold (dB)
Knee (dB)
Ratio (:1 dB)
5
50
0
0
1
TRACK WAVEFORMS
RMS
-14.7
-13.8
-17.4
-14.5
-15.0
-18.2
Play
Stop
Reset This Mix
Load All Mixes
Save All Mixes

FIG. 41

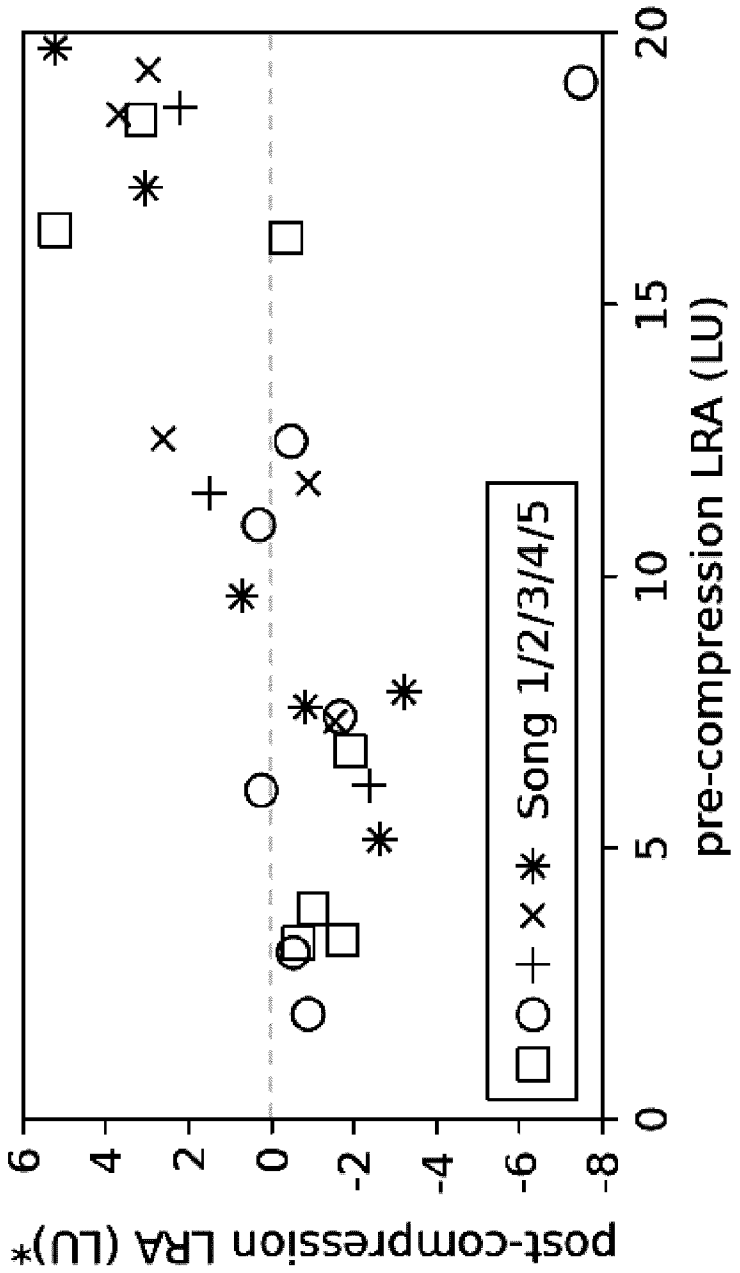

post-compression LRA (LU)*
pre-compression LRA (LU)
Song 1/2/3/4/5
6
4
2
0
-2
-4
-6
-8
0
5
10
15
20

FIG. 42

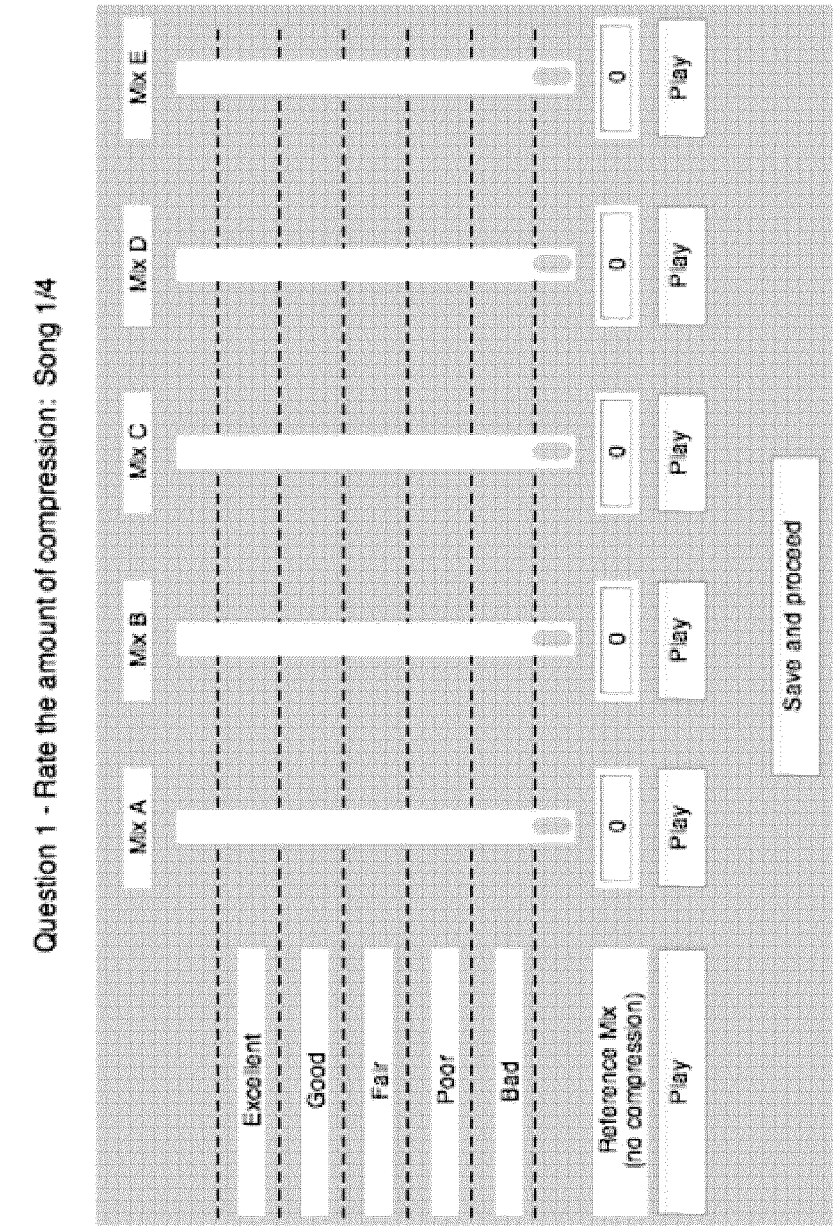

Question 1 - Rate the amount of compression: Song 1/4
Excellent
Good
Fair
Poor
Bad
Reference Mix (no compression)
Play
Mix A
0
Play
Mix B
0
Play
Mix C
0
Play
Mix D
0
Play
Mix E
0
Play
Save and proceed

FIG. 43

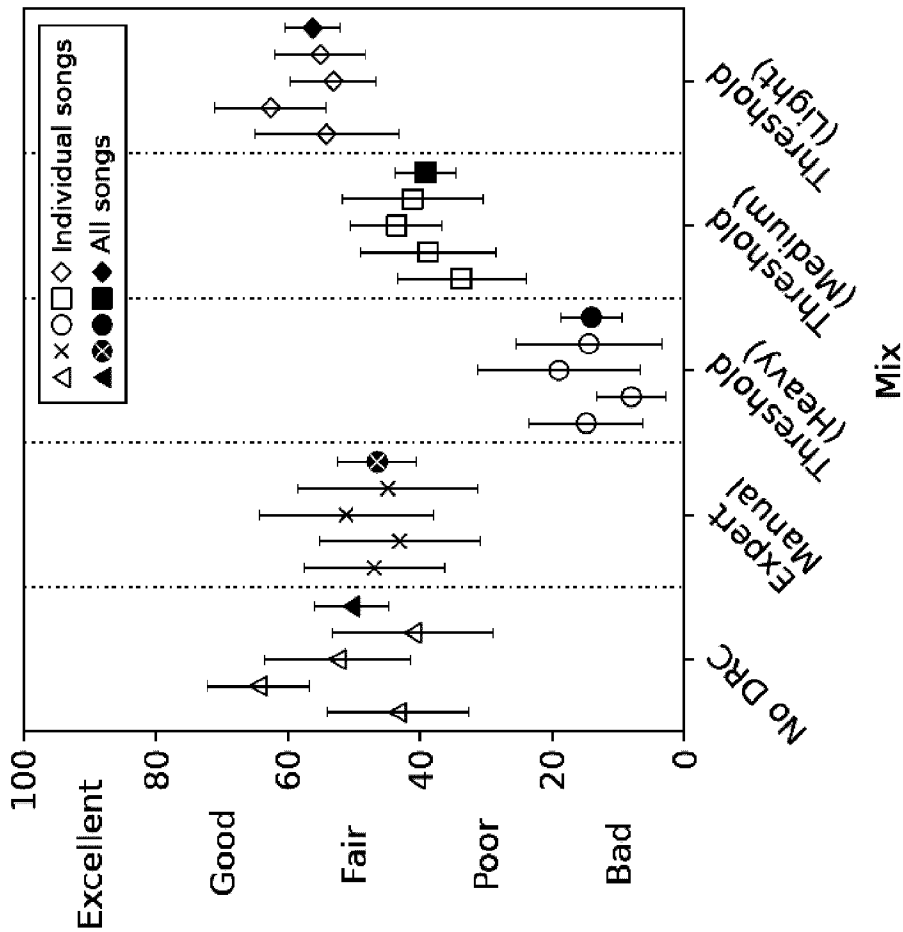

Individual songs
All songs
100
80
60
40
20
0
Excellent
Good
Fair
Poor
Bad
No DRC
Expert Manual
Threshold (Heavy)
Threshold (Medium)
Threshold (Light)
Mix

FIG. 44

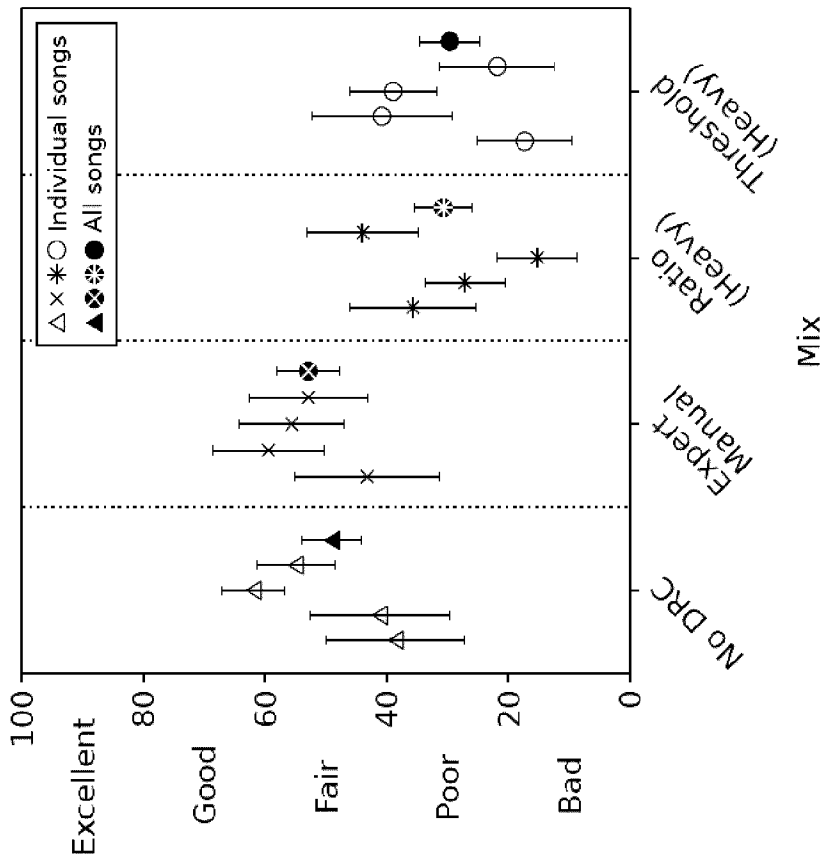

△ × ✳ ○ Individual songs
▲ ⊗ ❂ ● All songs
100
80
60
40
20
0
Excellent
Good
Fair
Poor
Bad
No DRC
Expert Manual
Ratio (Heavy)
Threshold (Heavy)
Mix

FIG. 45

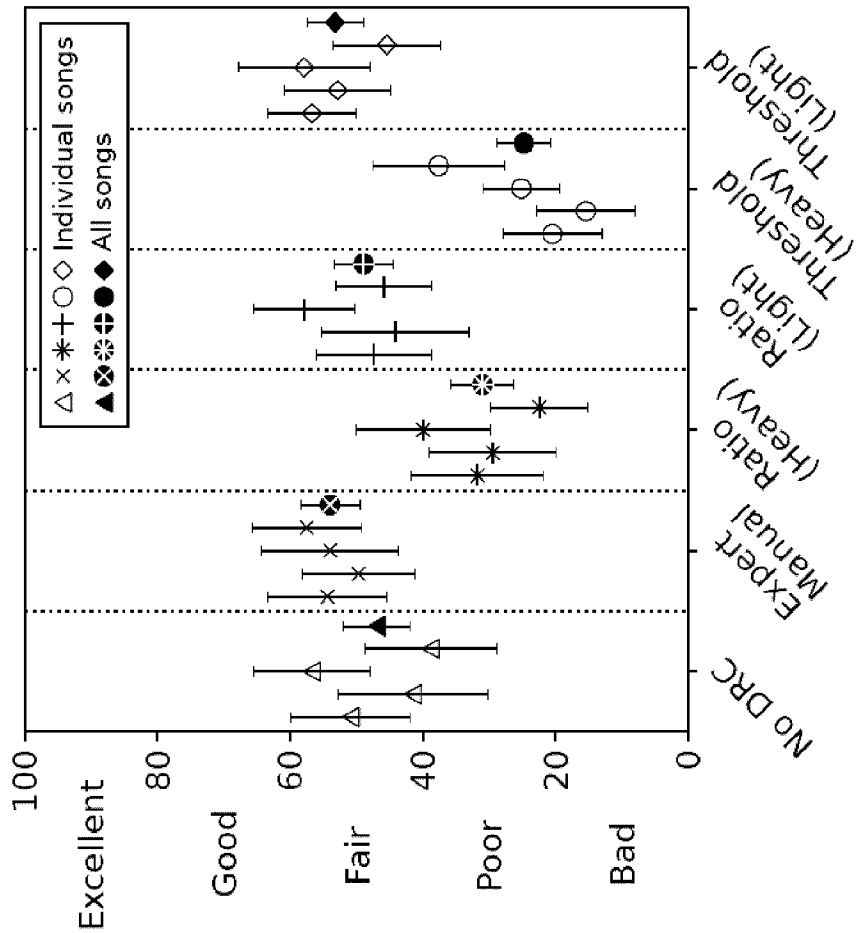

△ × ✳ + ○ ◇ Individual songs
▲ ⊗ ❋ ● ◆ All songs
100
80
60
40
20
0
Excellent
Good
Fair
Poor
Bad
No DRC
Expert Manual
Ratio (Heavy)
Ratio (Light)
Threshold (Heavy)
Threshold (Light)
Mix

FIG. 46

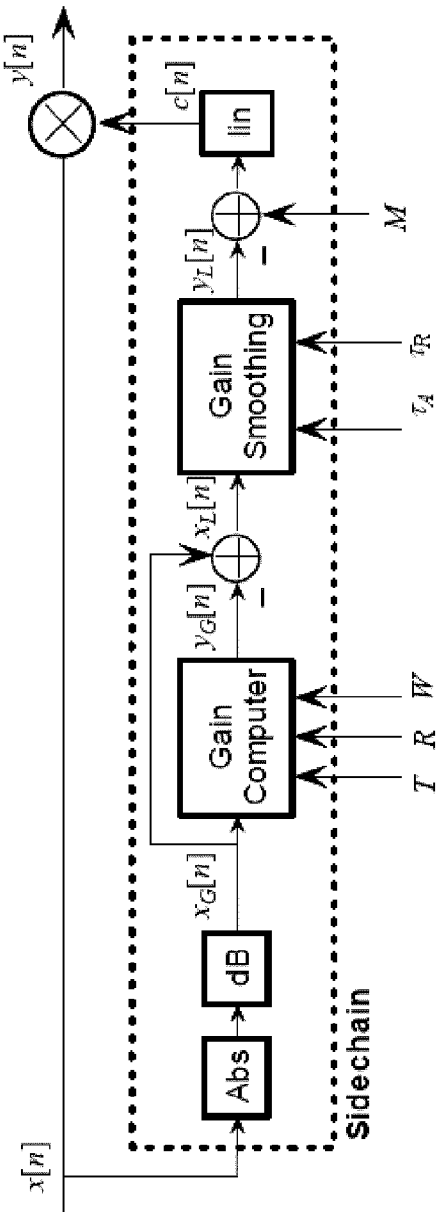

y[n]
c[n]
lin
M
y_L[n]
Gain Smoothing
τ_R
τ_A
x_L[n]
y_G[n]
Gain Computer
W
R
T
x_G[n]
dB
Abs
x[n]
Sidechain

FIG. 47

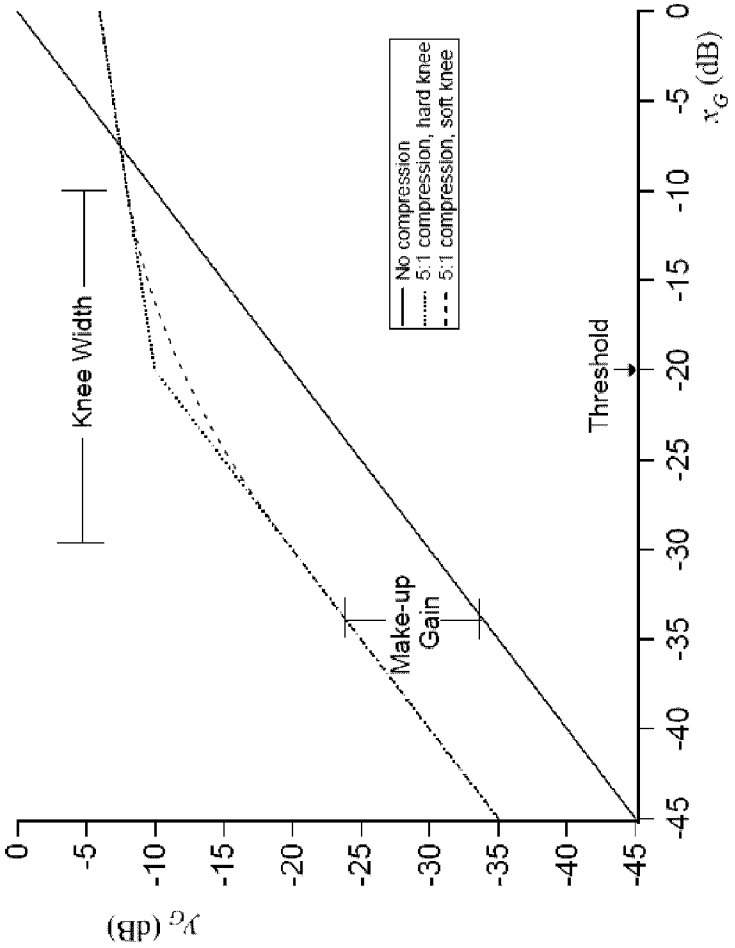
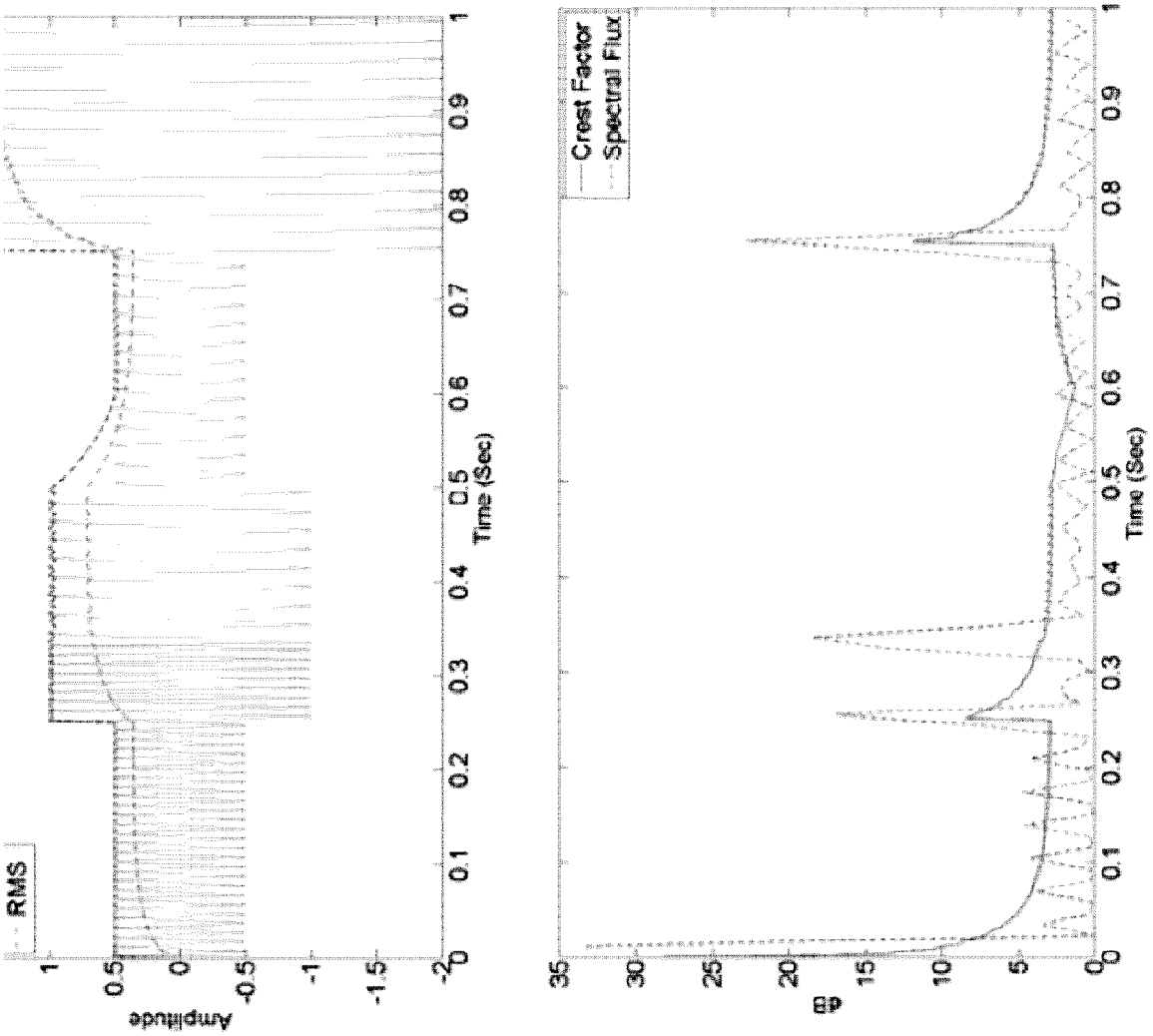

No compression
5:1 compression, hard knee
5:1 compression, soft knee
Knee Width
Make-up Gain
Threshold
y_G (dB)
x_G (dB)
0
-5
-10
-15
-20
-25
-30
-35
-40
-45

FIG. 48


RMS
Amplitude
Time (Sec)
Crest Factor
Spectral Flux
dB
Time (Sec)

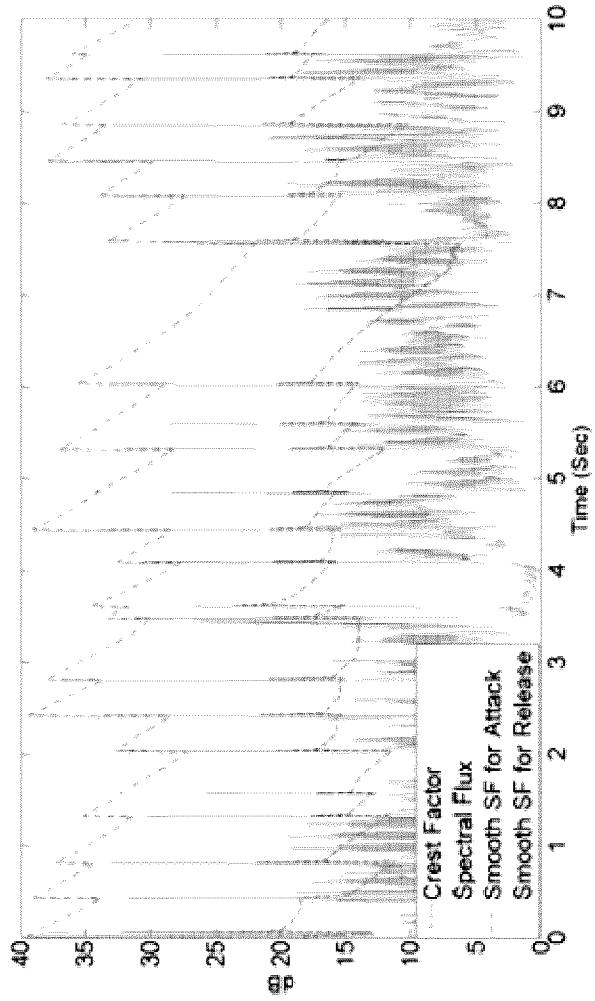

Crest Factor
Spectral Flux
Smooth SF for Attack
Smooth SF for Release
dB
Time (Sec)
0
5
10
15
20
25
30
35
40
0
1
2
3
4
5
6
7
8
9
10

FIG. 50

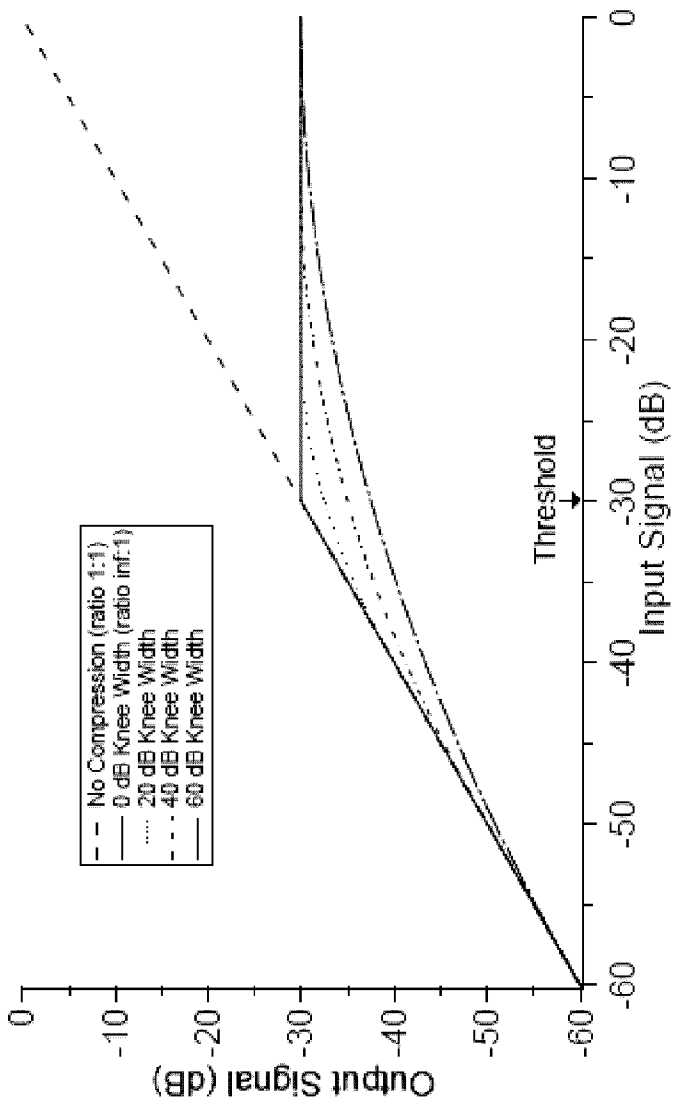
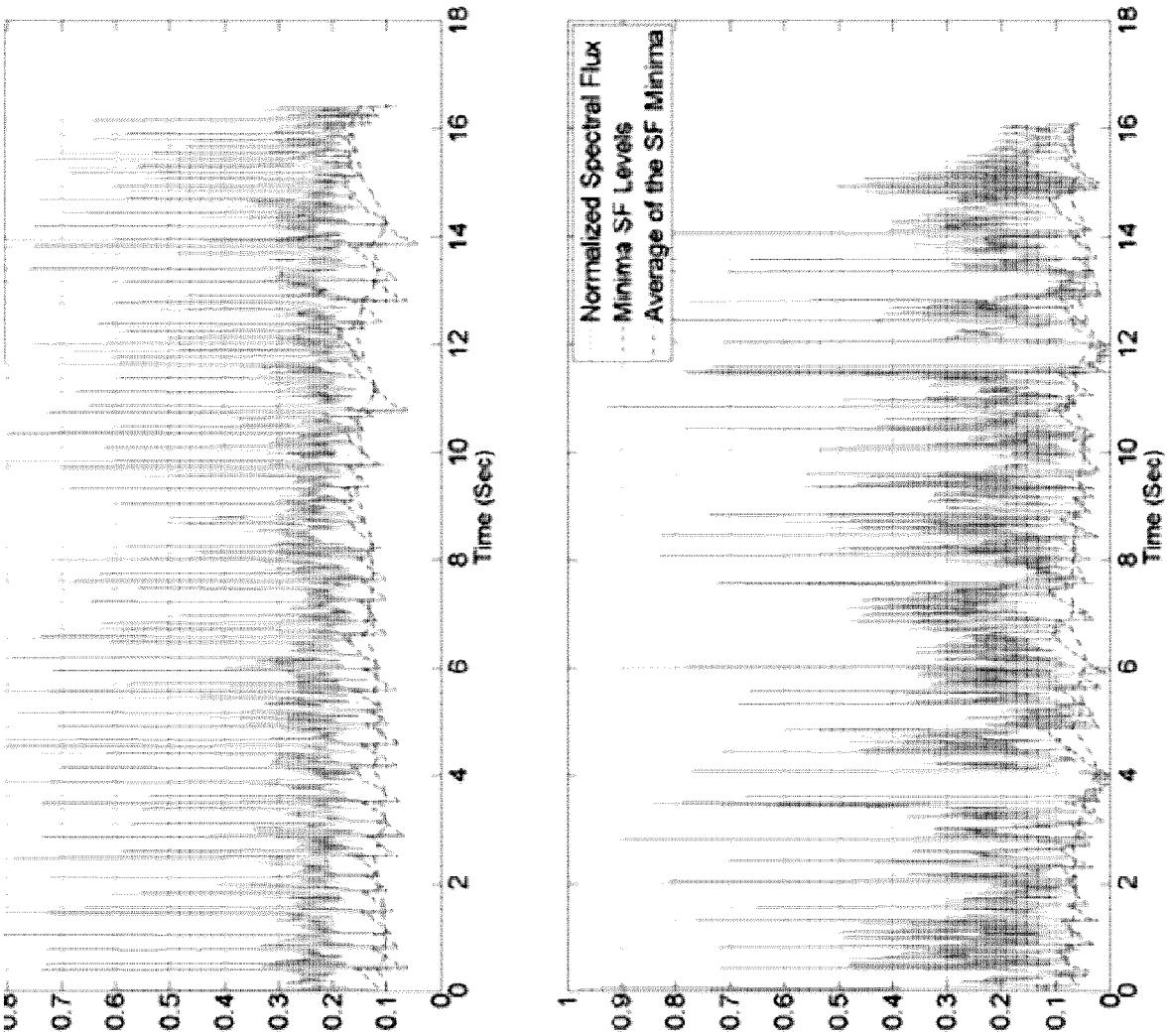

No Compression (ratio 1:1)
0 dB Knee Width (ratio inf:1)
20 dB Knee Width
40 dB Knee Width
60 dB Knee Width
Threshold
Output Signal (dB)
Input Signal (dB)
0
-10
-20
-30
-40
-50
-60

FIG. 51


Normalized Spectral Flux
Minima SF Levels
Average of the SF Minima
Time (Sec)
Time (Sec)

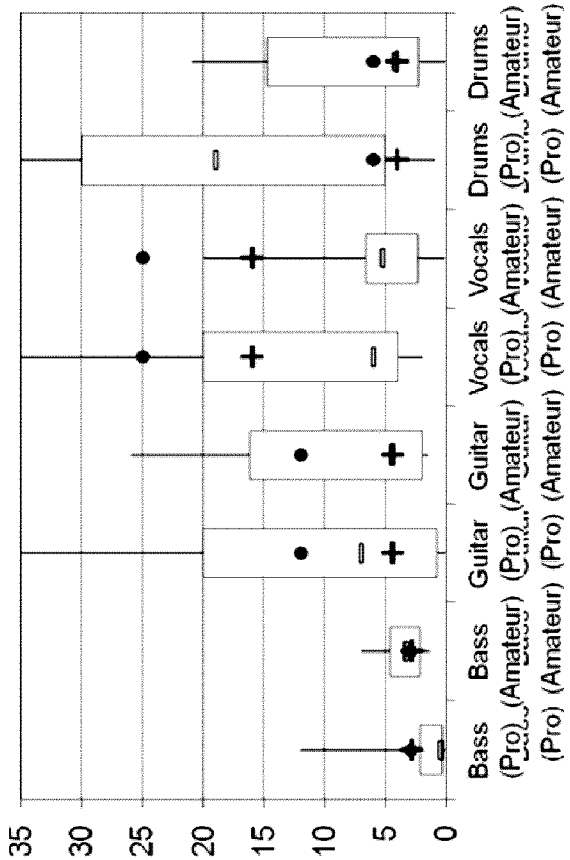

35
30
25
20
15
10
5
0
Bass (Pro)
Bass (Amateur)
Guitar (Pro)
Guitar (Amateur)
Vocals (Pro)
Vocals (Amateur)
Drums (Pro)
Drums (Amateur)

FIG. 53

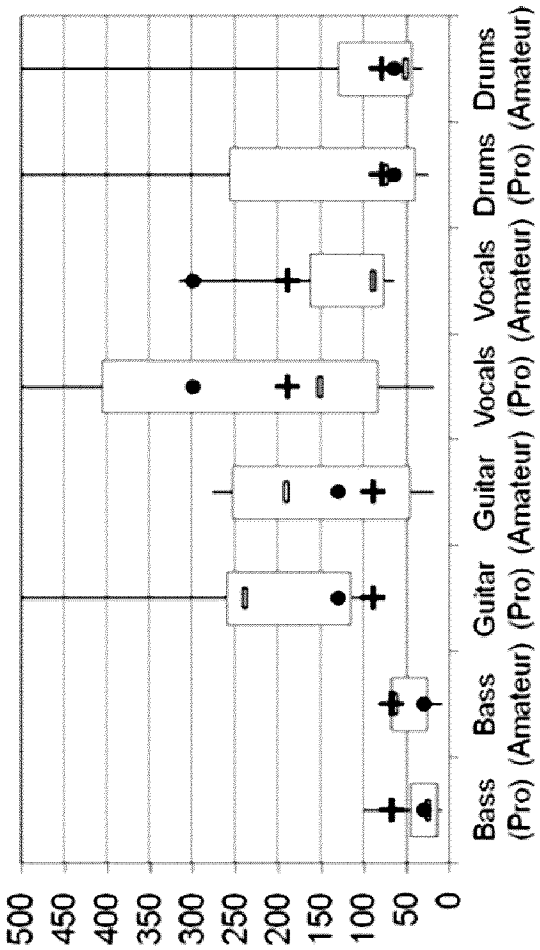

500
450
400
350
300
250
200
150
100
50
0
Bass (Pro)
Bass (Amateur)
Guitar (Pro)
Guitar (Amateur)
Vocals (Pro)
Vocals (Amateur)
Drums (Pro)
Drums (Amateur)

FIG. 54

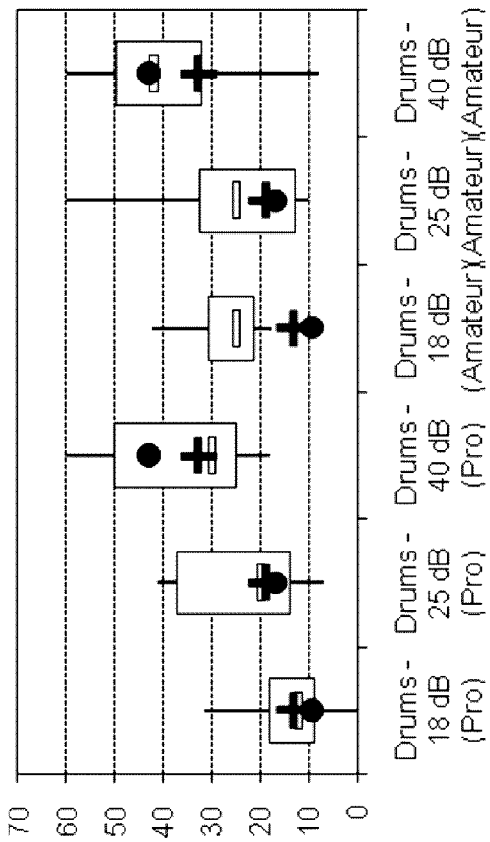

70
60
50
40
30
20
10
0
Drums - 18 dB (Pro)
Drums - 25 dB (Pro)
Drums - 40 dB (Pro)
Drums - 18 dB (Amateur)
Drums - 25 dB (Amateur)
Drums - 40 dB (Amateur)

FIG. 55

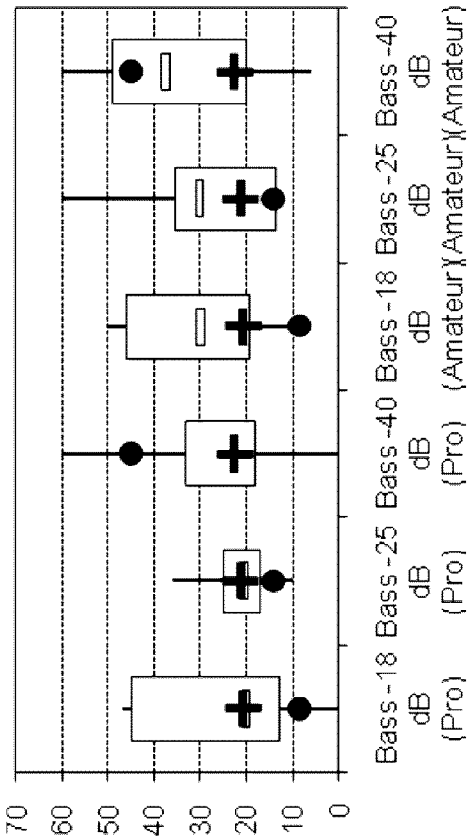

70
60
50
40
30
20
10
0
Bass -18 dB (Pro)
Bass -25 dB (Pro)
Bass -40 dB (Pro)
Bass -18 dB (Amateur)/(Amateur)
Bass -25 dB (Amateur)/(Amateur)
Bass -40 dB (Amateur)/(Amateur)

FIG. 56

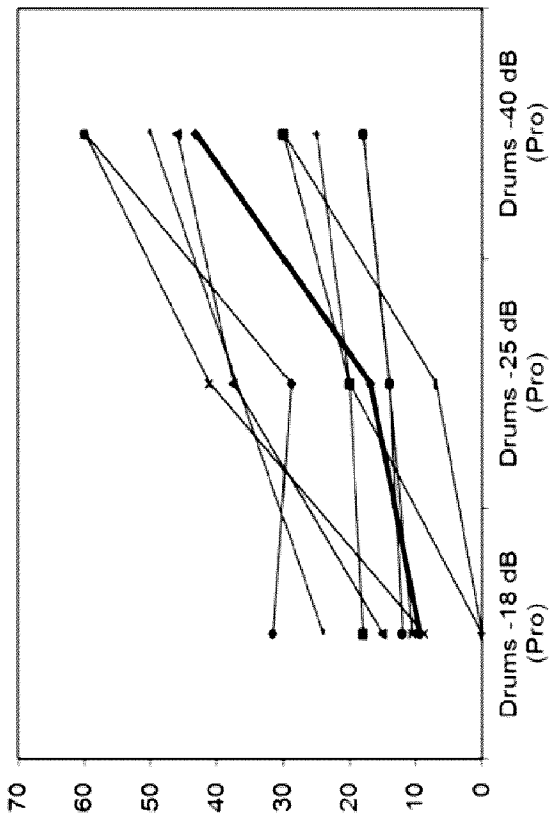

70
60
50
40
30
20
10
0
Drums -18 dB (Pro)
Drums -25 dB (Pro)
Drums -40 dB (Pro)

FIG. 57

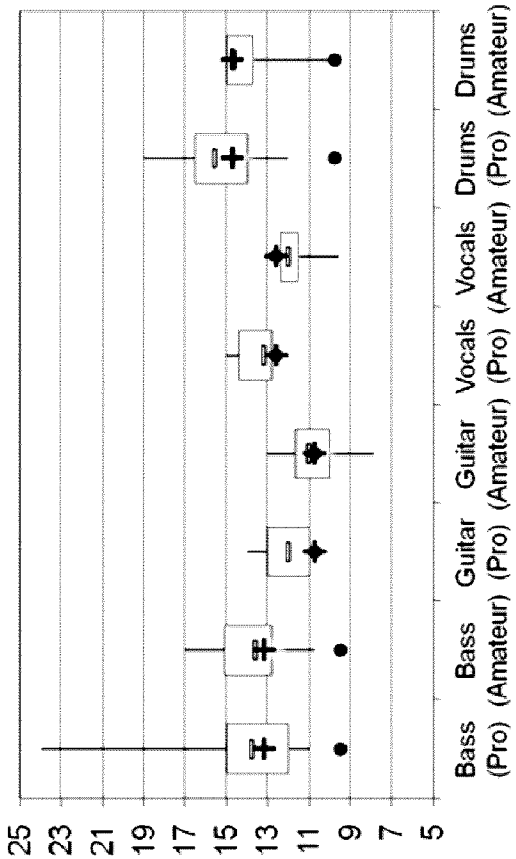

25
23
21
19
17
15
13
11
9
7
5
Bass (Pro)
Bass (Amateur)
Guitar (Pro)
Guitar (Amateur)
Vocals (Pro)
Vocals (Amateur)
Drums (Pro)
Drums (Amateur)

FIG. 58

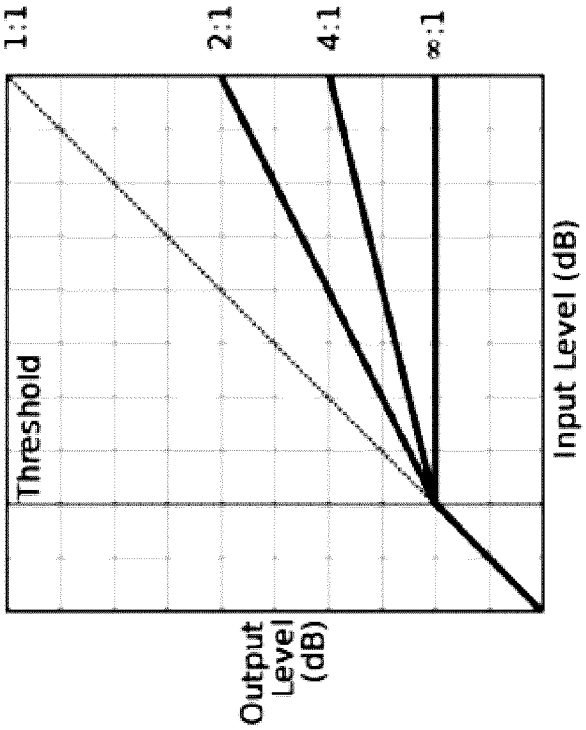


FIG. 59

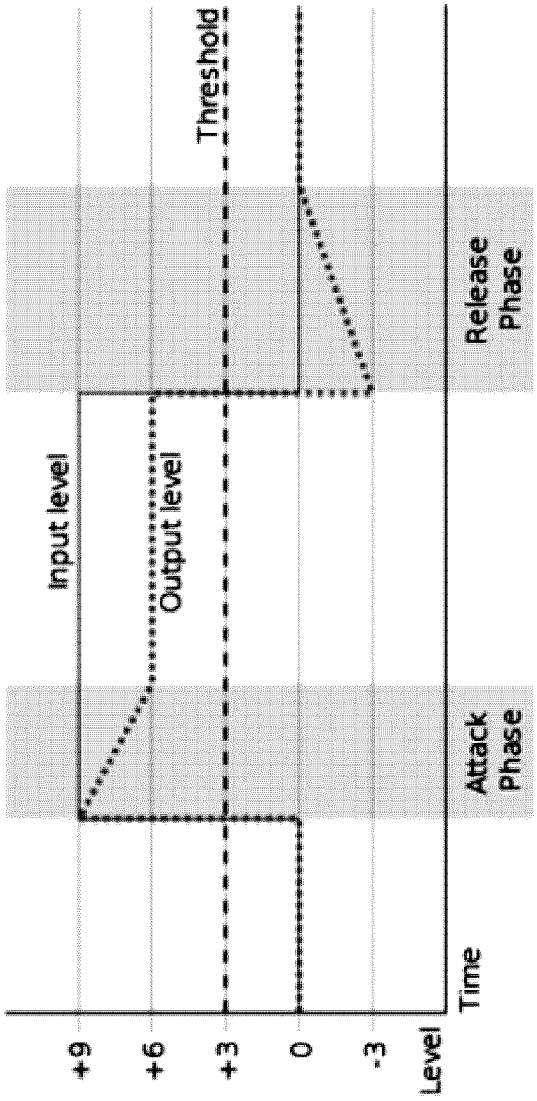

Threshold
Input level
Output level
Release Phase
Attack Phase
Time
Level
+9
+6
+3
0
-3

FIG. 60

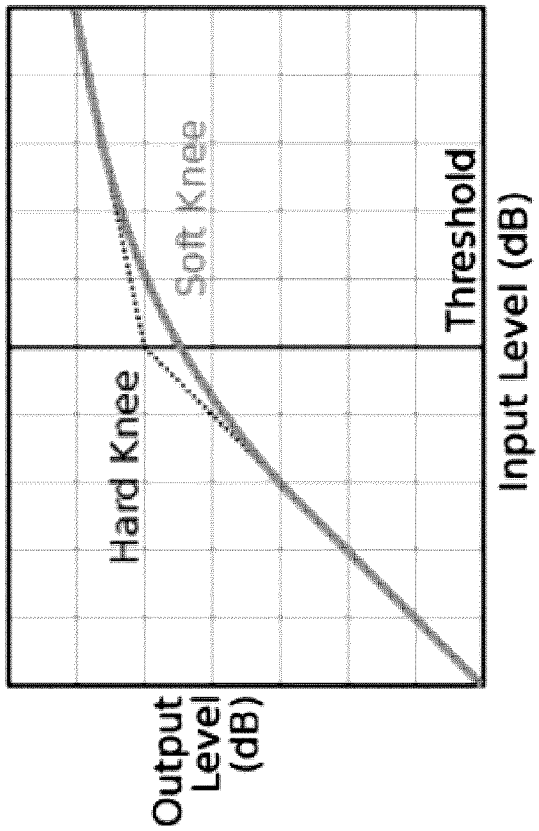

Hard Knee
Soft Knee
Threshold
Input Level (dB)
Output Level (dB)

FIG. 61

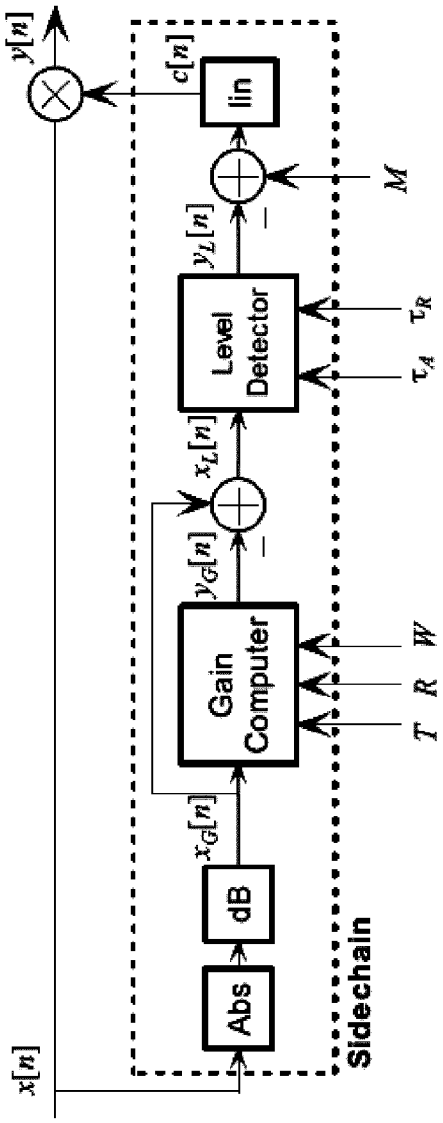
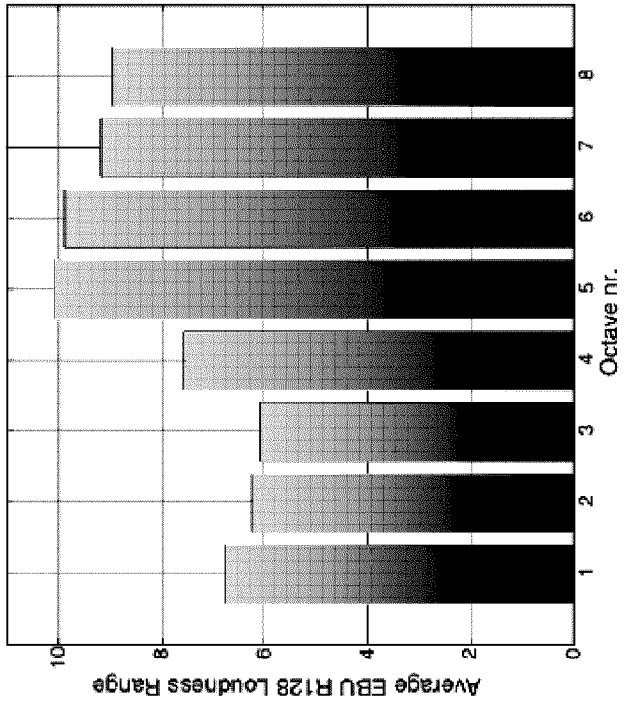

y[n]
c[n]
lin
M
y_L[n]
Level Detector
τ_R
τ_A
x_L[n]
y_G[n]
Gain Computer
W
R
T
x_G[n]
dB
Abs
x[n]
Sidechain

FIG. 62


Average EBU R128 Loudness Range
0
2
4
6
8
10
1
2
3
4
5
6
7
8
Octave nr.

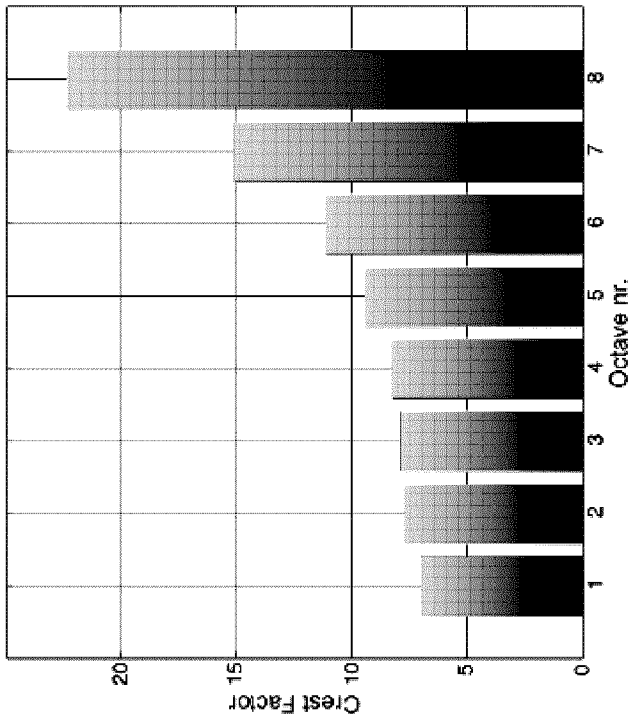

Crest Factor
0
5
10
15
20
1
2
3
4
5
6
7
8
Octave nr.

FIG. 63

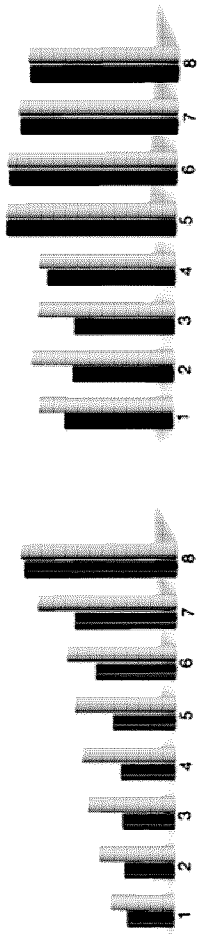


FIG. 64

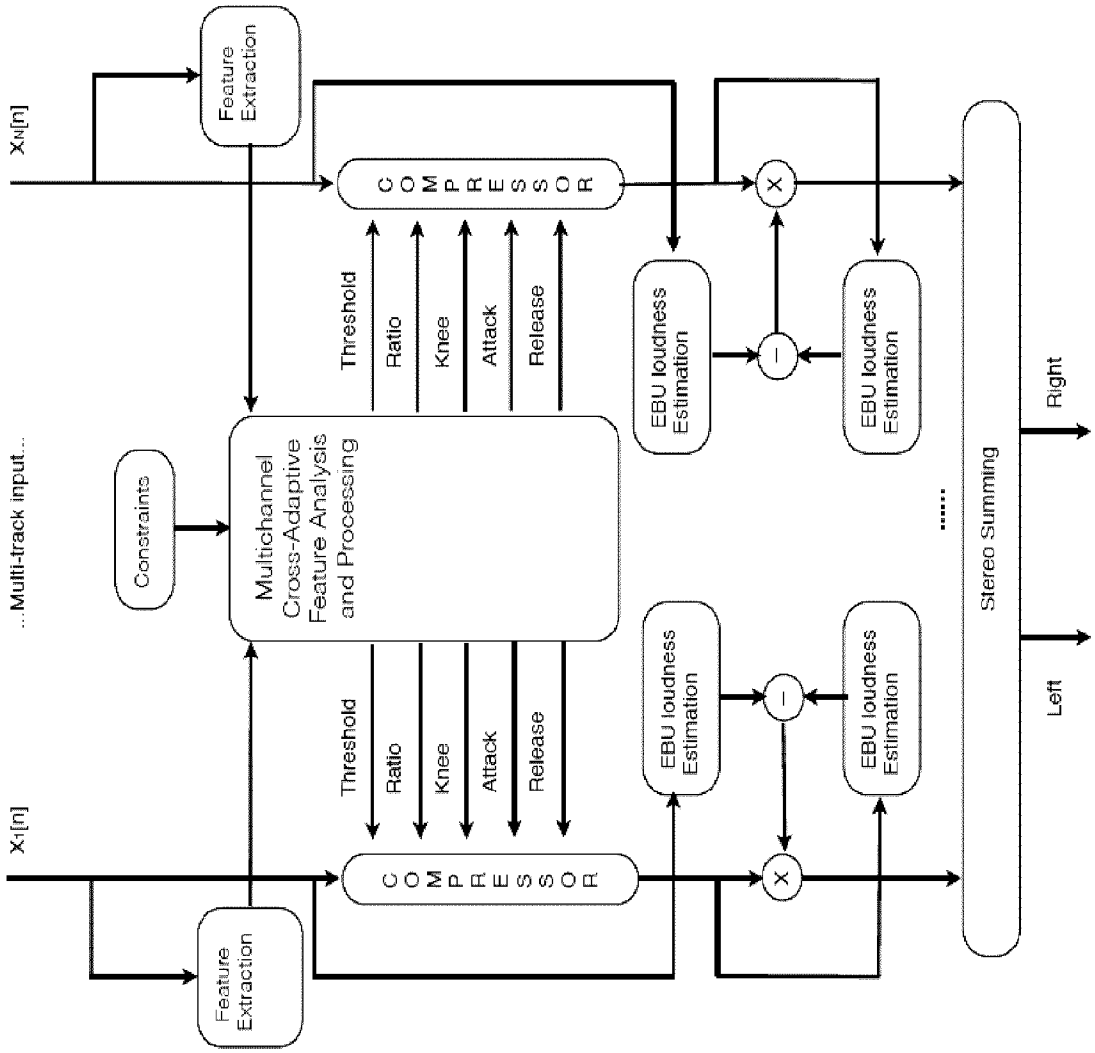

...Multi-track input...
X1[n]
XN[n]
Feature Extraction
Feature Extraction
Constraints
Multichannel Cross-Adaptive Feature Analysis and Processing
Threshold
Ratio
Knee
Attack
Release
COMPRESSOR
COMPRESSOR
EBU loudness Estimation
EBU loudness Estimation
EBU loudness Estimation
EBU loudness Estimation
Stereo Summing
Right
Left

FIG. 65

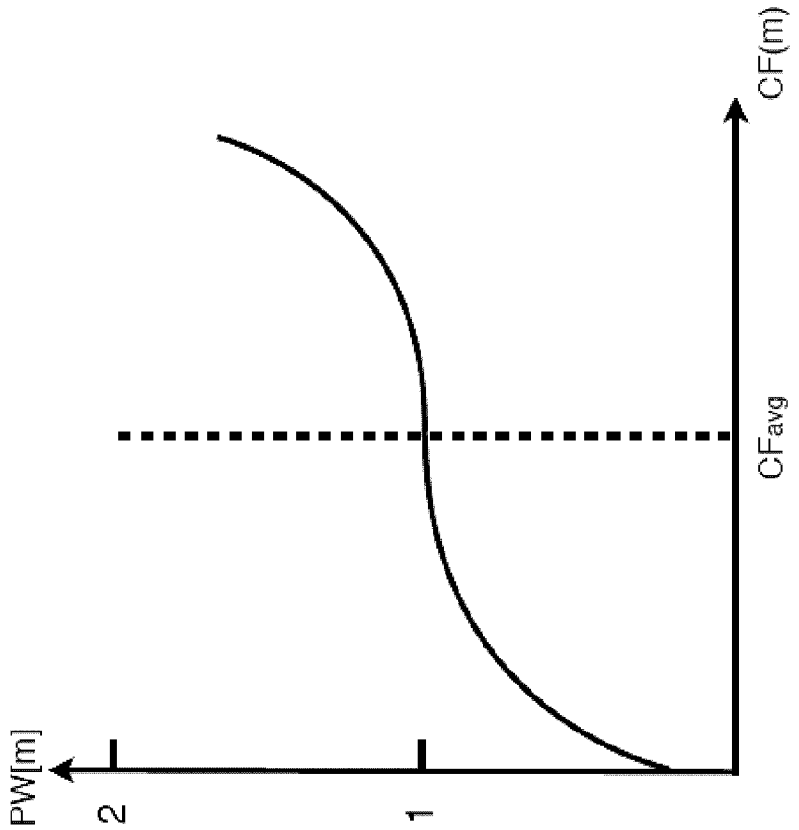


FIG. 66

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2013/050706

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***H04R 3/04*** (2006.01) , ***G10L 19/008*** (2013.01) , ***H04S 3/00*** (2006.01)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: ***H04R 3/04*** (2006.01) , ***G10L 19/008*** (2013.01) , ***H04S 3/00*** (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
TotalPatent, Esp@cenet, Canadian Patent Database, USPTO Database, IEEE Xplore.
Keywords: automatic, equalization, multi, track, audio, mixing, signal, filter, value, profile, band, gain, frequency, spectrum, target, profile, mask, compress.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, E	WO2013/167884 A1 (REISS ET AL.) 14 November 2013 -see whole document.	1-20, 28 and 33
X	US2012/0130516 A1 (REINSCH ET AL.) 24 May 2012 -see abstract; -see paragraphs [0023]-[0043]; -see figure 2B.	1-20, 28 and 33
A	PEREZ-GONZALEZ E ET AL: "Automatic gain and fader control for live mixing", APPLICATIONS OF SIGNAL PROCESSING TO AUDIO AND ACOUSTICS, 2009. WASPAA '09. IEEE WORKSHOP ON, IEEE, PISCATAWAY, NJ, USA, 18 October 2009 (2009-10-18), pages 1-4, XP031575123, ISBN: 978-1-4244-3678-1	1-20, 28 and 33
A	Enrique Perez Gonzalez ET AL: "Automatic mixing: live downmixing stereo panner", Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07), 10 September 2007 (2007-09-10), XP055076731, Bordeaux, France Retrieved from the Internet: URL:http://dafx.labri.fr/main/papers/p063.pdf	1-20, 28 and 33

[ ] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 January 2014 (17-01-2014)	05 March 2014 (05-03-2014)

Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 001-819-953-2476

Authorized officer
Hassan Bayaa (819) 997-7810

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2013/050706

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
WO2013167884A1	14 November 2013 (14-11-2013)	GB201208012D0	20 June 2012 (20-06-2012)
US2012130516A1	24 May 2012 (24-05-2012)	None	

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2013/050706

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of the first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons :

1. [ ] Claim Nos. :
because they relate to subject matter not required to be searched by this Authority, namely :

2. [ ] Claim Nos. :
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically :

3. [ ] Claim Nos. :
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows :

See extra sheet

1. [ ] As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. [ ] As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. [ ] As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claim Nos. :

4. [X] No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim Nos. : 1-20, 28 and 33

Remark on Protest

[ ] The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

[ ] The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

[ ] No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2013/050706

The claims are directed to a plurality of inventive concepts as follows:

Group A - Claims 1-20, 28 and 33 are directed to a method of performing automatic equalization in an automatic multi-track audio mixing system.
Group B - Claims 21-24 are directed to a method of performing gain compensation.
Group C - Claims 25-27 are directed to a method of generating a target spectrum for equalization of audio content.
Group D - Claims 29-31 are directed to a method of equalization of an audio track using a high pass filter.
Group E - Claim 32 is directed to an automatic multi-track audio mixing system.

The claims must be limited to one inventive concept as set out in Rule 13 of the PCT.