

CLOCK DRIFT COMPENSATION METHOD AND APPARATUS

Technical Field of the Invention

[0001] Various aspects of the present invention relate generally to a method and system that compensates for drift between sampling rates applied to a far-end signal and a near-end signal. The inventive techniques may be applied in acoustic echo cancellation in telecommunication systems. Further aspects of the invention relate to a method and system for compensating a drift between a far-end signal and a near-end signal arriving at an acoustic echo canceller.

Background of the Invention

[0002] Speech quality is an important factor for telephony system suppliers. Customer demand makes it vital to strive for continuous improvements. An echo, which is a delayed version of what was originally transmitted, is regarded as a severe distraction to the speaker if the delay is long. For short round trip delays of less than approximately 20 ms, the speaker will not be able to distinguish the echo from the side tone in the handset. However, for long-distance communications, such as satellite communications, a remotely generated echo signal often has a substantial delay. Moreover, the speech and channel coding compulsory in digital radio communications systems and for telephony over the Internet protocol (IP telephony, for short) also result in significant delays which make the echoes generated a relatively short distance away clearly audible to the speaker. Hence, canceling the echo is a significant factor in maintaining speech quality.

[0003] An echo canceller typically includes a linear filtering part which essentially is an adaptive filter that tries to adapt to the echo path. In this way, a replica of the echo can be produced from the far-end signal and subtracted from the near-end signal, thereby canceling the echo.

[0004] The filter generating the echo replica may have a finite or infinite impulse response. Most commonly it is an adaptive, linear finite impulse response (FIR) filter with a number of delay lines and a corresponding number of coefficients, or filter delay taps. The coefficients are values, which when multiplied with delayed versions of the filter

input signal, generate an estimate of the echo. The filter is adapted, i.e. updated, so that the coefficients converge to optimum values. A traditional way to cancel out the echo is to update a FIR filter using the normalized least mean square (NLMS) algorithm.

[0005] While employing an acoustic echo cancellation (AEC) system, the sample rates of the render and capture streams may differ by an unknown amount. This will be the case when using different devices for rendering and capturing such as on a personal computer (e.g. a webcam for capture and built-in output) if the clock frequency of either device varies from its nominal value. The sample rate mismatch causes the far-end and near-end signals arriving to the AEC to “drift” away from each other. The linear filter stage, forced to adapt to this drifting delay, may be unable to converge to a good estimate. Echo cancellation performance can suffer considerably as a result.

[0006] Thus, during echo cancellation, it is desired to compensate for the drift between the far-end signal and the near-end signal arriving at the AEC to address the sample rate mismatch problem. Other systems and methods may also benefit from reducing or eliminating sample rate drift.

Summary of the Invention

[0007] This Summary introduces a selection of concepts in a simplified form in order to provide a basic understanding of some aspects of the present disclosure. This Summary is not an extensive overview of the disclosure, and is not intended to identify key or critical elements of the disclosure or to delineate the scope of the disclosure. This Summary merely presents some of the concepts of the disclosure as a prelude to the Detailed Description provided below.

[0008] According to an aspect of the present invention, a method to compensate for a drift between a first signal and a second signal is disclosed. The method includes receiving, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of the first signal and a second data set containing information related to a sample rate of the second signal. The method also includes producing a one-time estimate of a drift between the first and second signals based on the first data set and the second data set during the predetermined amount of time, and resampling one, or both, of the first and

second signals according to the estimated drift to compensate for the drift between the first and second signals.

[0009] In accordance with a further aspect of the present invention, the method includes performing echo cancellation where the first signal is a far-end signal to be rendered and the second signal is a near-end captured signal.

[0010] According to a further aspect of the present invention, the method includes performing echo suppression with the near-end signal and the far-end signal.

[0011] According to yet another aspect of the present invention, the information related to the sample rate of the first signal includes counts of number of samples rendered per frame and the information related to the sample rate of the second signal includes counts of number of samples captured per frame.

[0012] In accordance with a further aspect of the present invention, the information related to the sample rate of the first signal includes timestamps indicating when frames are rendered and the information related to the sample rate of the second signal includes timestamps indicating when frames are captured.

[0013] According to yet another aspect of the present invention, the method includes resampling the first signal.

[0014] According to a further aspect of the present invention, the method includes applying resampling in a first signal path bound for an echo canceller and not a second signal path bound for rendering.

[0015] According to yet another aspect of the present invention, the resampling step utilizes a linear resampling method.

[0016] According to a further aspect of the present invention, the method includes pruning outliers from the first data set and second data set prior to producing the estimate. According to an aspect of the present invention, pruning is based on rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

[0017] According to another aspect of the present invention, pruning is based on rejecting outliers greater than a first threshold determined from the first data set and the second data set or less than a second threshold determined from the first data set and the second data set.

[0018] According to an aspect of the present invention, the first threshold and the second threshold are proportional to a measure of spread of the first data set and the second data set.

[0019] According to yet another aspect of the present invention, the measure of spread is an absolute deviation.

[0020] According to another aspect of the present invention, the method includes utilizing linear regression to produce the estimate.

[0021] According to a further aspect of the present invention, the method includes disabling the resampling when the estimated drift is less than a first predetermined threshold or greater than a second predetermined threshold.

[0022] In addition, according an aspect of the present invention, the method includes, computing a first coherence measure by comparing correlations between the first signal and the second signal, computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter, and applying the first and second coherence measures to compute echo cancellation information.

[0023] According to yet another aspect of the present invention, the method includes computing a first coherence measure by comparing correlations between the first signal and the second signal, computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter, and applying the first and second coherence measures to compute suppression factors.

[0024] According to a further aspect of the present invention, the suppression factors are applied to the error signal to substantially reduce the echo in the error signal.

[0025] According to another aspect of the present invention, a system to compensate for a drift between a first signal and a second signal is disclosed. The system includes a drift estimator and a resampler. The drift estimator receives, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of the first signal and a second data set containing information related to a sample rate of the second signal, and produces a one-time estimate of a drift between the first and second signals based on the first data set and the second data set during the predetermined amount of time. The resampler

resamples one, or both, of the first and second signals according to the estimated drift to compensate for the drift between the first and second signals.

[0026] According to an aspect of the present invention, the system also includes an echo canceller operatively connected to the resampler, the echo canceller performing echo cancellation where the first signal is a far-end signal to be rendered and the second signal is a near-end captured signal.

[0027] In accordance with another aspect of the present invention, the echo canceller performs echo suppression with the near-end signal and the far-end signal.

[0028] According to an aspect of the present invention, the resampler is configured to resample the first signal.

[0029] According to another aspect of the present invention, the resampler is configured to apply resampling in a first signal path bound for an echo canceller and not a second signal path bound for rendering.

[0030] According to a further aspect of the present invention, the resampler is configured to utilize a linear resampling method.

[0031] According to a further aspect of the present invention, the drift estimator is configured to prune outliers from the first data set and second data set prior to producing the estimate.

[0032] According to a further aspect of the present invention, the drift estimator is configured to prune by rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

[0033] According to another aspect of the present invention, the drift estimator is configured to prune by rejecting outliers greater than a first threshold determined from the first data set and the second data set or less than a second threshold determined from the first data set and the second data set.

[0034] According to an aspect of the present invention, the drift estimator is also configured to apply linear regression to produce the estimate.

[0035] In accordance with an aspect of the present invention, the resampler is configured to disable the resampling when the estimated drift is less than a first predetermined threshold

or greater than a second predetermined threshold.

[0036] According to a further aspect of the present invention, the system includes a non-linear processor that is configured to compute a first coherence measure by comparing correlations between the first signal and the second signal, compute a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter, and apply the first and second coherence measures to compute echo cancellation information.

[0037] In addition, according to an aspect of the present invention, the non-linear processor is configured to compute a first coherence measure by comparing correlations between the first signal and the second signal, compute a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and apply the first and second coherence measures to compute suppression factors.

[0038] According to a further aspect of the present invention, the non-linear processor is configured to apply the suppression factors to the error signal to substantially reduce the echo in the error signal.

[0039] According to a further aspect of the present invention, a computer-readable storage medium having stored thereon computer executable program to compensate for a drift between a first signal and a second signal is disclosed. The computer program when executed causes a processor to execute the steps of: receiving, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of the first signal and a second data set containing information related to a sample rate of the second signal, producing a one-time estimate of a drift between the first and second signals based on the first data set and the second data set during the predetermined amount of time, and resampling one, or both, of the first and second signals according to the estimated drift to compensate for the drift between the first and second signals.

[0040] According to an aspect of the present invention, the computer program when executed causes the processor to further execute the step of performing echo cancellation where the first signal is a far-end signal to be rendered and the second signal is a near-end captured signal.

[0041] According to another aspect of the present invention, the computer program when executed causes the processor to further execute the step of performing echo suppression with the near-end signal and the far-end signal.

[0042] According to an aspect of the present invention, the computer program when executed causes the processor to further execute the step of resampling the first signal.

[0043] According to a further aspect of the present invention, the computer program when executed causes the processor to further execute the step of applying resampling in a first signal path bound for an echo canceller and not a second signal path bound for rendering.

[0044] According to another aspect of the present invention, the computer program when executed causes the processor to further execute the step of performing resampling by applying a linear resampling method.

[0045] According to yet another aspect of the present invention, the computer program when executed causes the processor to further execute the step of pruning outliers from the first data set and second data set prior to producing the estimate.

[0046] According to an aspect of the present invention, the computer program when executed causes the processor to further execute the step of pruning based on rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

[0047] According to a further aspect of the present invention, the computer program when executed causes the processor to further execute the step of pruning based on rejecting outliers greater than a first threshold determined from the first data set and the second data set or less than a second threshold determined from the first data set and the second data set.

[0048] According to a further aspect of the present invention, the computer program when executed causes the processor to further execute the step of utilizing linear regression to produce the estimate.

[0049] In accordance with an aspect of the present invention, the computer program when executed causes the processor to further execute the step of disabling the resampling when the estimated drift is less than a first predetermined threshold or greater than a second predetermined threshold.

[0050] According to another aspect of the present invention, the computer program when executed causes the processor to further execute the steps of computing a first coherence measure by comparing correlations between the first signal and the second signal, computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter, and applying the first and second coherence measures to compute echo cancellation information.

[0051] According to a further aspect of the present invention, the computer program when executed causes the processor to further execute the steps of computing a first coherence measure by comparing correlations between the first signal and the second signal, computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter, and applying the first and second coherence measures to compute suppression factors.

[0052] According to yet another aspect of the present invention, the computer program when executed causes the processor to further execute the step of applying suppression factors to the error signal to substantially reduce the echo in the error signal.

Brief Description of the Drawings

[0053] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and together with the description, serve to explain the principles of the invention.

[0054] Fig. 1 is a block diagram of an acoustic echo canceller in accordance with an embodiment of the present invention.

[0055] Fig. 2 illustrates a more detailed block diagram describing the functions that may be performed in the adaptive filter of Fig. 1 in accordance with an embodiment of the present invention.

[0056] Fig. 3 illustrates computational stages of the adaptive filter of Fig. 2 in accordance with an embodiment of the present invention.

[0057] Fig. 4 illustrates a more detailed block diagram describing how the block G_m in Fig. 3 may operate in accordance with an embodiment of the present invention.

[0058] Fig. 5 illustrates a flow diagram describing computational processes of the non-linear processor of Fig. 1 in accordance with an embodiment of the present invention.

[0059] Fig. 6 is a block diagram of an acoustic echo canceller and a drift estimator in accordance with an embodiment of the present invention.

[0060] Fig. 7 is a flow diagram illustrating operations performed by the acoustic echo canceller and the drift estimator according to an embodiment of the present invention illustrated in Fig. 6.

[0061] Fig. 8 is a block diagram illustrating an exemplary computing device that is arranged for acoustic echo cancellation in accordance with an embodiment of the present invention.

Detailed Description

[0062] The following detailed description of the embodiments of the invention refers to the accompanying drawings. The following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims and equivalents thereof.

[0063] Fig. 1 illustrates an acoustic echo canceller (AEC) 100 in accordance with an exemplary embodiment of the present invention.

[0064] The AEC 100 is designed as a high quality echo canceller for voice and audio communication over packet switched networks. More specifically, the AEC 100 is designed to cancel acoustic echo 130 that emerges due to the reflection of sound waves of a render device 10 from boundary surfaces and other objects back to a near-end capture device 20. The echo 130 may also exist due to the direct path from render device 10 to the capture device 20.

[0065] Render device 10 may be any of a variety of audio output devices, including a loudspeaker or group of loudspeakers configured to output sound from one or more channels. Capture device 20 may be any of a variety of audio input devices, such as one or more microphones configured to capture sound and generate input signals. For example, render device 10 and capture device 20 may be hardware devices internal to a computer system, or external peripheral devices connected to a computer system via wired and/or wireless

connections. In some arrangements, render device 10 and capture device 20 may be components of a single device, such as a microphone, telephone handset, etc. Additionally, one or both of render device 10 and capture device 20 may include analog-to-digital and/or digital-to-analog transformation functionalities.

[0066] With reference to Fig. 1, the echo canceller 100 includes a linear filter 102, a nonlinear processor (NLP) 104, a far-end buffer 106, and a blocking buffer 108. A far-end signal 110 generated at the far-end and transmitted to the near-end is input to the filter 102 via the far-end buffer (FEBuf) 106 and the blocking buffer 108. The far-end signal 110 is also input to a play-out buffer 112 located near the render device 10. The output signal 116 of the far-end buffer 106 is input to the blocking buffer 108 and the output signal 118 of the blocking buffer is input to the linear filter 102.

[0067] The far-end buffer 106 is configured to compensate for and synchronize to buffering at sound devices (not shown). The blocking buffer 108 is configured to block the signal samples for a frequency-domain transformation to be performed by the linear filter 102 and the NLP 104.

[0068] The linear filter 102 is an adaptive filter. Linear filter 102 operates in the frequency domain through, e.g., the Discrete Fourier Transform (DFT). The DFT may be implemented as a Fast Fourier Transform (FFT).

[0069] The other input to the filter 102 is the near-end signal (Sin) 122 from the capture device 20 via a recording buffer 114. The near-end signal 122 includes near-end speech 120 and the echo 130. The NLP 104 receives three signals as input. It receives (1) the far-end signal via the far-end buffer 106 and blocking buffer 108, (2) the near-end signal via the recording buffer 114, and (3) the output signal 124 of the filter 102. The output signal 124 is also referred to as an error signal. In a case when the NLP 104 attenuates the output signal 124, a comfort noise signal is generated which will be explained later.

[0070] According to an exemplary embodiment, each frame is divided into 64 sample blocks. Since this choice of block size does not produce an integer number of blocks per frame the signal needs to be buffered before the processing. This buffering is handled by the blocking buffer 108 as discussed above. Both the filter 102 and the NLP 104 operate in the frequency domain and utilize DFTs of 128 samples.

[0071] The performance of the AEC 100 is influenced by the operation of the play-out buffer 112 and the recording buffer 114 at the sound device. The AEC 100 may not start unless the combined size of the play-out buffer 112 and the recording buffer 114 is reasonably stable within a predetermined limit. For example, if the combined size is stable within ± 8 ms of the first started size, for four consecutive frames, the AEC 100 is started by filling up the internal far-end buffer 106.

[0072] Fig. 2 illustrates a more detailed block diagram describing the functions performed in the filter 102 of Fig. 1. Fig. 3 illustrates computational stages of the filter 102 in accordance with an embodiment of the present invention.

[0073] With reference to Fig. 2, the adaptive filter 102 includes a first transform section 200, an inverse transform section 202, a second transform section 204, and an impulse response section (H) 206. The far-end signal $x(n)$ 210 to be rendered at the render device 10 is input to the first transform section 200. The output signal $X(n, k)$ of the first transform section 200 is input to the impulse response section 206. The output signal $Y(n, k)$ is input to the second transform section 202 which outputs the signal $y(n)$. This signal $y(n)$ is then subtracted from the near-end signal $d(n)$ 220 captured by the capture device 20 to output an error signal $e(n)$ 230 as the output of the linear stage of the filter 102. The error signal 230 is also input to the second transform section 204 the output signal of which, $E(n, k)$, is also input to the impulse response section 206.

[0074] The above-mentioned adaptive filtering approach relates to an implementation of a standard blocked time-domain Least Mean Square (LMS) algorithm. According to an embodiment of the invention, the complexity reduction is due to the filtering and the correlations being performed in the frequency domain, where time-domain convolution is replaced by multiplication. The error is formed in the time domain and is transformed to the frequency domain for updating the filter 102 as illustrated in Fig. 2.

[0075] There is a signal delay in the system due to the transform blocking. To reduce delay the filter 102 is partitioned in smaller segments and by overlap-save processing the overall delay is kept to the segment length. This method is referred to as partitioned block frequency domain method or multi-delay partitioned block frequency adaptive filter. For simplicity it is referred to as FLMS.

[0076] The operation of the FLMS method is illustrated in Fig. 3. Fig. 4 illustrates a

more detailed block diagram describing block G_m in the FLMS method of Fig. 3 in accordance with an embodiment of the present invention.

[0077] With a total filter length $L = M \cdot N$ partitioned in blocks of N samples and with $F = 2N \times 2N$ Discrete Fourier Transform (DFT) matrix, the time domain impulse response of the filter 102, $w(n)$, $n = 0, 1, \dots, L - 1$, can be expressed in the frequency domain as a collection of partitioned filters

$$\mathbf{W}_m(k) = F \begin{bmatrix} I_N \\ 0_N \end{bmatrix} \mathbf{w}_m(k), \quad (1)$$

where $\mathbf{w}_m(k) = [w_{mN} \dots w_{(m+1)N-1}]^T$,

I_N is a $N \times N$ -sized identity matrix, and 0_N is a $N \times N$ -sized zero matrix. This means that the time domain vector is appended with N zeros before the Fourier transform.

[0078] The time domain filter coefficients, $w(n)$ are not utilized in the algorithm and equation (1) is presented to establish the relation between the time- and frequency-domain coefficients.

[0079] As illustrated in Fig. 3, the far-end samples, $x(n)$ 310, are blocked into vectors of $2N$ samples, i.e. two blocks, at step S312,

$$\mathbf{x}(k-m) = [x((k-m-2)N) \dots x((k-m)N-1)]^T$$

and transformed into a sequence of DFT vectors at step S314,

$$X(k-m) = \text{diag}(F\mathbf{x}(k-m)).$$

[0080] This is implemented as a table of delayed DFT vectors, since the diagonal matrix also can be expressed as $X(k-m) = D^m X(k)$, where D is a delay operator. For each delayed block altering is performed as the multiplication of the diagonal matrix $X(k-m)$

with a filter partition

$$\mathbf{Y}_m(k) = \mathbf{X}(k - m) \mathbf{W}_m(k) \quad m = 0, 1, \dots, M-1$$

[0081] The estimated echo signal is then obtained as the N last coefficients of the inverse transformed sum of the filter products performed at step S320 from which first block is discarded at step S322. The estimated echo signal is represented as

$$\mathbf{y}(k) = ([y((k-1)N) \dots y(kN-1)])^T = [0_N \ I_N] \mathbf{F}^{-1} \sum_{m=0}^{M-1} \mathbf{Y}_m(k)$$

[0082] The error is then formed in the time domain as

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{y}(k)$$

and this is also the output of the filter 102 of the AEC 100 as shown in Fig. 1. To adjust the filter coefficients, N zeros are inserted at step S316 to the error vector, and the augmented vector is transformed at step S318 as

$$\mathbf{E}(k) = \mathbf{F} \begin{bmatrix} \mathbf{I}_N \\ 0_N \end{bmatrix} \mathbf{e}(k).$$

[0083] Fig. 4 illustrates a more detailed block diagram describing block G_m in Fig. 3 in accordance with an embodiment of the present invention where the filter coefficient update can be expressed as

$$\mathbf{W}_m(k+1) = \mathbf{W}_m(k) + \mathbf{F} \begin{bmatrix} \mathbf{I}_N & 0_N \\ 0_N & 0_N \end{bmatrix} \mathbf{F}^{-1} \mu_0 \mathbf{X}^*(k-m) \mathbf{B}(k),$$

with a stepsize $\mu_0 = 0.5$ and where $\mathbf{B}(k)$, as shown in Fig. 4, is a modified error vector. The

modification includes a power normalization followed by a magnitude limiter 410. The normalized error vector, as also shown in Fig. 4, is

$$\mathbf{A}(k) = \Omega(k)\mathbf{E}(k),$$

where

$$\Omega(k) = \text{diag} ([1/p_0 \ 1/p_1 \ \dots \ 1/p_{2N-1}])$$

is a diagonal step size matrix controlling the adjustment of each frequency component using power estimates

$$p_j(k) = \lambda_p p_j(k-1) + (1 - \lambda_p) M |X_{j,j}|^2, \quad j = 0, 1, \dots, 2N-1,$$

recursively calculated with a forgetting factor $\lambda_p = 0.9$ and individual DFT coefficients $X_{j,j} = \{X(k)\}_{j,j}$ is input to the magnitude limiter 410. The component magnitudes are then limited to a constant maximum magnitude, $A_0 = 1.5 \times 10^{-6}$, into the vector $\mathbf{B}(k)$ with components

$$B_j(k) = \begin{cases} \frac{A_j(k)}{|A_j(k)|} A_0 & : |A_j(k)| > A_0 \\ A_j(k) & : |A_j(k)| \leq A_0 \end{cases}$$

[0084] As illustrated in Fig. 4, the diagonal matrix $\mathbf{X}(k-m)$ is conjugated by the conjugate unit 420 which is then multiplied with vector $\mathbf{B}(k)$ prior to performing an inverse DFT transform by the Inverse Discrete Fourier Transform (IDFT) unit 430. Then the discard last block unit 440 discards the last block. After discarding the last block, a zero block is appended by the append zero block unit 450 prior to performing a DFT by the DFT unit 460. Then, a block delay is introduced by the delay unit 480 which outputs $\mathbf{W}_m(k)$.

[0085] Fig. 5 illustrates a flow diagram describing computational processes of the NLP 104 of Fig. 1 in accordance with an embodiment of the present invention.

[0086] The NLP 104 of the AEC 100 accepts three signals as input: i) the far-end signal

x(n) 110 to be rendered by the render device 10, ii) the near-end signal d(n) 122 captured by the capture device 20, and iii) the output error signal e(n) 124 of the linear stage performed at the filter 102. The error signal e(n) 124 typically contains residual echo that should be removed for good performance. The objective of the NLP 104 is to remove this residual echo.

[0087] The first step is to transform all three input signals to the frequency domain. At step S501, the far-end signal 110 is transformed to the frequency domain. At step S501', the near-end signal 122 is transformed to the frequency domain and at step S501'', the error signal 124 is transformed to the frequency domain. The NLP 104 is block-based and shares the block length N of the linear stage, but uses an overlap-add method rather than overlap-save: consecutive blocks are concatenated, windowed and transformed. By defining o as the element-wise product operator, the kth transformed block is expressed as

$$\mathbf{X}_k = \mathbf{F} \left(\mathbf{w}_{2N} \circ \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{bmatrix} \right), \quad k \geq 0, \quad \mathbf{x}_{-1} = \mathbf{0}_N$$

where F is the 2N DFT matrix as before, $\mathbf{x}_k$ is a length N time-domain sample column vector and $\mathbf{w}_{2N}$ is a length 2N square-root Hanning window column vector with entries

$$w(n) = \sqrt{\frac{1}{2} \left[1 - \cos \left(\frac{2\pi n}{2N} \right) \right]}, \quad n = 0, 1, \dots, 2N - 1$$

[0088] The window is chosen such that the overlapping segments satisfy

$$w^2(n) + w^2(n - N) = 1, \quad n = N, N + 1, \dots, 2N$$

to provide perfect reconstruction. According to an embodiment of the invention, the length 2N DFT vectors are retained. Preferably, however, the redundant N - 1 complex coefficients are discarded.

[0089] $\mathbf{X}_k$, $\mathbf{D}_k$ and $\mathbf{E}_k$ refer to the frequency-domain representations of the kth far-end, near-end and error blocks, respectively.

[0090] According to a further embodiment of the invention, echo suppression is achieved by

multiplying each frequency band of the error signal $e(n)$ 124 with a suppression factor between 0 and 1. According to a preferred embodiment, each band corresponds to an individual DFT coefficient. In general, however, each band may correspond to an arbitrary range of frequencies. Comfort noise is added and after undergoing an inverse FFT, the suppressed signal is windowed, and overlapped and added with the previous block to obtain the output.

[0091] For analysis, the power spectral density (PSD) of each signal is obtained. At step S503, the PSD of the far-end signal $x(n)$ 110 is computed. At step S503', the PSD of the near-end signal $d(n)$ 122 is computed and at step S503'', the PSD of the error signal $e(n)$ 124 is computed. The PSDs of the far-end signal 110, near-end signal 122, and the error signal 124 are represented by S_x , S_d , and S_e , respectively.

[0092] In addition, the complex-valued cross-PSDs between i) the far-end signal $x(n)$ 110 and near-end signal $d(n)$ 122, and ii) the near-end signal $d(n)$ 122 and error signal $e(n)$ 124 are also obtained. At step S504, the complex-valued cross-PSD between the far-end signal 110 and the near-end signal 122 is computed and at step S504', the complex-valued cross-PSD between the near-end signal 122 and the error signal 124 is computed. The complex-valued cross-PSD of the far-end signal 110 and near-end signal 122 is represented as S_{xd} . The complex-valued cross-PSD of the near-end signal 122 and error signal 124 is represented as S_{de} . The PSDs are exponentially smoothed to avoid sudden erroneous shifts in echo suppression. The PSDs are given by

$$S_{X_k Y_k} = \lambda_S S_{X_{k-1} Y_{k-1}} + (1 - \lambda_S) X_k \circ Y_k^*, \quad k > 0, \quad S_{X_0 Y_0} = 0_N$$

where the "*" here represents the complex conjugate, and where the exponential smoothing factor is given by

$$\lambda_S = \begin{cases} 0.9 & \text{if } f_s = 8000 \\ 0.93 & \text{otherwise} \end{cases}$$

[0093] Note that $X_k = Y_k$ for the "auto" PSDs, which are therefore real-valued while the cross-PSDs are complex valued.

[0094] Rather than using the current input far-end block, an old block is selected to

best synchronize it with the corresponding echo in the near-end at step S505. The index of the partition, m , with maximum energy in the linear filter is chosen as follows:

$$\bar{d} = \arg \max_m (||\mathbf{W}_m||^2)$$

[0095] This estimated delay index is used to select the best block at step S507 for use in the far-end PSDs. Additionally, the far-end auto-PSD is thresholded at step S509 in order to avoid numerical instability as follows:

$$\mathbf{S}'_{X_k X_k} = \max(\mathbf{S}_{X_k X_k}, S_0), \quad S_0 = 15$$

[0096] It is sometimes the case that the linear filter 102 diverges from a good echo path estimate. This tends to result in a highly distorted error signal, which although still useful for analysis, should not be used for output. According to an embodiment of the invention, divergence may be detected fairly easily, as it usually adds rather than removes energy from the near-end signal $d(n)$ 122. The divergence state determined at step S511 is utilized to either select (S512) $\mathbf{E}_k$ or $\mathbf{D}_k$ as follows: If

$$||\mathbf{S}_{E_k E_k}||_1 > ||\mathbf{S}_{D_k D_k}||_1$$

then the "diverge" state is entered, in which the effect of the linear stage is reversed by setting $\mathbf{E}_k = \mathbf{D}_k$. The diverge state is left if

$$\sigma_0 ||\mathbf{S}_{E_k E_k}||_1 < ||\mathbf{S}_{D_k D_k}||_1, \quad \sigma_0 = 1.05$$

Furthermore, if divergence is very high, such as

$$||\mathbf{S}_{E_k E_k}||_1 > \sigma_1 ||\mathbf{S}_{D_k D_k}||_1, \quad \sigma_1 = 19.95$$

the linear filter 102 resets to its initial state

$$\mathbf{W}_m(k) = \mathbf{0}_N, \quad m = 0, 1, \dots, M-1$$

The PSDs are used to compute the coherence measures for each frequency band between i) the far-end signal 110 and near-end signal 122 at step S513 as follows:

$$\mathbf{c}_{xd} = \frac{\mathbf{S}_{X_{k-\bar{d}}D_k} \circ \mathbf{S}_{X_{k-\bar{d}}D_k}^*}{\mathbf{S}_{X_{k-\bar{d}}X_{k-\bar{d}}} \circ \mathbf{S}_{D_kD_k}}$$

and ii) the near-end signal 122 and error signal 124 at step S515 as follows:

$$\mathbf{c}_{de} = \frac{\mathbf{S}_{D_kE_k} \circ \mathbf{S}_{D_kE_k}^*}{\mathbf{S}_{D_kD_k} \circ \mathbf{S}_{E_kE_k}}$$

where the “*” here again represents the complex conjugate.

[0097] Denote a $\mathbf{c}$ vector entry in position n as $c(n)$. Coherence is a frequency- domain analog to time-domain correlation. It is a measure of similarity with $0 \leq c(n) \leq 1$; where a higher coherence corresponds to more similarity.

[0098] The primary effect of the NLP 104 is achieved through directly suppressing the error signal 124 with the coherence measures. Generally speaking, the output is given by

$$\tilde{\mathbf{Y}}_k = \mathbf{E}_k \circ \mathbf{c}_{de}.$$

Under the assumption that the linear stage is working properly, $c(n)_{de} \approx 1$ when no echo has been removed, allowing the error to pass through unchanged. In the opposite case of the linear stage having removed echo, $1 \gg c(n)_{de} \geq 0$, resulting in a suppression of the error, ideally removing any residual echo remaining after the linear filtering by the filter 102 at the linear stage.

[0099] According to an embodiment of the invention, $\mathbf{c}_{xd}$ is considered to increase robustness, as described below, though $\mathbf{c}_{de}$ tends to be more useful in practice. Contrary to $\mathbf{c}_{de}$, $\mathbf{c}_{xd}$ is relatively high when there is echo 130, and low otherwise. To have the two measures in the same "domain" a modified coherence is defined as follows: $\mathbf{c}'_{xd} = 1 - \mathbf{c}_{xd}$.

[00100] It is preferred that to achieve high AEC performance, the echo 130 is suppressed while allowing simultaneous near-end speech 120 to pass through. The NLP 104 is configured to achieve this because the coherence is calculated independently for each frequency band. Thus, bands containing echo are fully or partially suppressed, while bands free of echo are not affected.

[00101] According to an embodiment of the invention, several data analysis method are used to tweak the coherence before it is applied as a suppression factor, $\mathbf{s}$. First, the average

coherence across a set of preferred bands is computed at step S517 for $\mathbf{c}_{de}$, and at step S517' for $\mathbf{c}'_{xd}$ as

$$\bar{c} = \frac{1}{n_1 - n_0 + 1} \sum_{n=n_0}^{n_1} c(n), \quad \{n_0, n_1\} = \left\lfloor \{500, 3500\} \frac{N}{f_s} \right\rfloor$$

where f_s is the sampling frequency. The preferred bands were chosen from frequency regions most likely to be accurate across a range of scenarios.

[00102] At step S518, the system either selects $\bar{c}_{de}$ or $\bar{c}_{xd}$. According to an exemplary embodiment, $\bar{c}_{xd}$ is tracked over time to determine the broad state of the system at step S521. The purpose of this is to avoid suppression when the echo path is close to zero (e.g. during a call with a headset). First, a thresholded minimum of $\bar{c}_{xd}$ is computed at step S519 as follows:

$$\hat{c}_{xd_k} = \begin{cases} \bar{c}_{xd_k} & \text{if } \bar{c}_{xd_k} < \hat{c}_{xd_{k-1}} < 0.75 \\ \min(\bar{c}_{xd_{k-1}} + \mu_c, 1) & \text{otherwise} \end{cases}, \quad k > 0, \quad \hat{c}_{xd_0} = 1$$

with a step-size $\mu_c = 0.0006m_{fs}$ and factor m_{fs} given by

$$m_{fs} = \begin{cases} 1 & \text{if } f_s = 8000 \\ 0.5 & \text{otherwise} \end{cases}$$

[00103] This is used to construct two decision variables

$$u_{c_k} = \begin{cases} 0 & \text{if } \bar{c}_{de_k} < 0.95 \text{ and } \bar{c}_{xd_k} < 0.8 \\ 1 & \text{if } \bar{c}_{de_k} > 0.98 \text{ and } \bar{c}_{xd_k} > 0.9 \\ u_{c_{k-1}} & \text{otherwise} \end{cases}, \quad k > 0, \quad \hat{u}_{c_0} = 0$$

, and

$$u_{e_k} = \begin{cases} 0 & \text{if } \hat{c}_{xd_k} = 1 \text{ or } u_k = 1 \\ 1 & \text{otherwise} \end{cases}, \quad k > 0, \quad \hat{u}_{e_0} = 0$$

[00104] The system is considered in the “coherent state” when $u_c = 1$ and in the “echo state” when $u_e = 1$. In the echo state, the system may contain echo and otherwise does not contain echo. The echo state is provided through an interface for potential use by other audio processing components.

[00105] While in the echo state, the suppression factor s is computed at step S520 by

selecting the minimum of $\mathbf{c}_{de}$, $\mathbf{c}'_{xd}$ in each band as

$$s = \min(\mathbf{c}_{de}, \mathbf{c}'_{xd})$$

[00106] Two overall suppression factors are computed at step S533 and S527 from order statistics across the preferred bands:

$$\{s_h, s_l\} = \{s_{(n_h)}, s_{(n_l)}\}, \quad \{n_h, n_l\} = \lfloor n_0 + \{0.5, 0.75\}(n_1 - n_0 + 1) \rfloor$$

[00107] This approach of selecting suppression factors is more robust to outliers than the average, and allows tuning through the exact selection of the order statistic position.

[00108] While in the “no echo state” (i.e. $u_e = 0$), suppression is limited by selecting suppression factors as follows at step S520, S524 and S518:

$$\{s, s_h, s_l\} = \begin{cases} \{\mathbf{c}_{de}, \bar{\mathbf{c}}_{de}, \bar{\mathbf{c}}_{de}\} & \text{if } \hat{\mathbf{c}}_{xd_k} = 1 \text{ and } u_c = 1 \\ \{\mathbf{c}'_{xd}, \bar{\mathbf{c}}_{xd}, \bar{\mathbf{c}}_{xd}\} & \text{otherwise} \end{cases}$$

[00109] Across most scenarios, there is a typical suppression level required to reasonably remove all residual echo. This is considered to be the target suppression, s_t . A scalar “overdrive” is applied to s to weight the bands towards s_t . This improves performance in more difficult cases where the coherence measures are not accurate enough by themselves. The minimum s_l level is computed at step S527 and tracked at step S529 over time

$$\{\hat{s}'_{l_k}, \hat{s}_{l_k}\} = \begin{cases} \{s_t, s_l\} & \text{if } s_l < \hat{s}_{l_{k-1}} < 0.6 \\ \{\hat{s}'_{l_{k-1}}, \min(\hat{s}_{l_{k-1}} + \mu_s, 1)\} & \text{otherwise} \end{cases}, \quad k > 0, \hat{s}_{l_0} = 1$$

with a step-size $\mu_s = 0.0008 m_{fs}$.

[00110] When the minimum $\hat{s}'_{lk}$ is unchanged for two consecutive blocks, the overdrive γ is set at step S531 such that applying it to the minimum will result in the target suppression level:

$$\gamma = \frac{s_t}{\log(\hat{s}_l)}$$

γ is smoothed and threshold as

$$\hat{\gamma}_k = \lambda_\gamma \hat{\gamma}_{k-1} + (1 - \lambda_\gamma) \max(\gamma, \gamma_0), \quad \lambda_\gamma = \begin{cases} 0.99 & \text{if } \gamma < \hat{\gamma}_k \\ 0.9 & \text{otherwise} \end{cases}$$

such that it will tend to move faster upwards than downwards. s_t and γ_0 are configurable to control the suppression aggressiveness; by default they are set to -11.5 and 2, respectively. Additionally, when

$$\hat{c}_{xd_k} = 1$$

the smoothed overdrive is reset to the minimum,

$$\hat{\gamma}_k = \gamma_0.$$

[00111] The s_h level is computed at step S533. Next, the final suppression factors s_γ are produced according to the following algorithm. At step S525 s is first weighted towards s_h according to a weighting vector $\mathbf{v}_{sN}$ with components $0 \leq v_{sN}(n) \leq 1$:

$$s_v(n) = \begin{cases} v_{sN}(n)s_h + [1 - v_{sN}(n)]s(n) & \text{if } s(n) > s_h \\ s(n) & \text{otherwise} \end{cases}$$

[00112] The weighting is selected to influence typically less accurate bands more heavily. Applying the overdriving at step S535, the following is derived:

$$s_\gamma(n) = s_v(n)^{\hat{\gamma}_k v_{\gamma N}(n)}$$

where $\mathbf{v}_{\gamma N}$ is another weighting vector fulfilling a similar purpose as $\mathbf{v}_{sN}$. Overdriving through raising to a power serves to accentuate valleys in s_v . Finally, at step S536 the frequency-domain output block is given by

$$\tilde{\mathbf{Y}}_k = \mathbf{s}_\gamma \circ \mathbf{E}_k + \mathbf{N}'_k$$

where N_{sk} is artificial noise and at step S537, an inverse transform is performed to obtain the output signal $y(n)$. The suppression removes near-end noise as well as echo, resulting in an audible change in the noise level. This issue is mitigated by adding generated "comfort noise" to replace the lost noise. The generation of N'_k will be discussed in a later section below.

[00113] The overlap-add transformation is inverted to arrive at the length N time-domain output signal $\tilde{y}_k$ as

$$\begin{bmatrix} \tilde{y}_k \\ y'_{k+1} \end{bmatrix} = (F^{-1} \tilde{Y}_k) \circ w_{2N} + \begin{bmatrix} y'_k \\ 0_N \end{bmatrix}, \quad k \geq 0, \quad y'_0 = 0_N$$

[00114] To generate comfort noise, a reliable estimate of the true near-end background noise is required. According to an embodiment of the invention, a minimum statistics method is utilized to generate the comfort noise. More specifically, at every block a modified minimum of the near-end PSD is computed for each band:

$$\tilde{N}_k(n) = \begin{cases} \lambda_N (S_{Dk}(n) + \mu(\tilde{N}_{k-1}(n) - S_{Dk}(n))) & \text{if } S_{Dk}(n) < \tilde{N}_{Dk}(n) \\ \lambda_N \tilde{N}_{k-1}(n) & \text{otherwise} \end{cases}, \quad k > 0, \quad \tilde{N}_0(n) = 10^6$$

with a step-size $\mu = 0.1$ and ramp $\lambda_N = 1.0002$. $\tilde{N}_0(n)$ is set such that it will be greater than a reasonable noise power. S_{Dk} is very similar to that discussed above, but is instead computed from the un-windowed DFT coefficients of the linear filter 102 computed at the linear stage.

[00115] White noise may be produced by generating a random complex vector, u_{2N} , on the unit circle. This is shaped to match $\tilde{N}_{Dk}$ and weighted by the suppression levels to give the following comfort noise:

$$N'_k = \tilde{N}_k \circ u_{2N} \circ \sqrt{1 - s_\gamma \circ s_\gamma}$$

[00116] Fig. 6 is a block diagram of AEC 100 and a drift estimator 600 in accordance with an embodiment of the present invention.

[00117] According to an exemplary embodiment of the present invention, the drift estimator 600 is provided between the render device 10 and the capture device 20 so that any drift between the far-end signal 110 and the near-end signal 120 can be compensated

prior to arriving at the AEC 100.

[00118] According to a further embodiment of the present invention, the AEC 100 includes a resampler 602, a far-end buffer 604, a linear adaptive filter 606, and a non-linear post-processor (NLP) 608. Note that, according to an exemplary embodiment, the linear adaptive filter 606 performs the same functionalities as the filter 102 described above with reference to Figs. 1-4 in addition to the functionalities described herein with reference to Fig. 6 and the far-end buffer 604 performs the same functionalities as the far-end buffer 106 and the blocking buffer 108 described above with reference to Fig 1-4 in addition to the functionalities described herein with reference to Fig. 6. Similarly, the NLP 608 performs the same functionalities as the NLP 104 described above with reference to Figs. 1 and 5 in addition to the functionalities described herein with reference to Fig. 6.

[00119] The drift estimator 600 receives, as input, information about the sample rates of both the far-end signal 110 and the near-end signal 122 as shown by the dotted lines in Fig. 6. The output from the drift estimator 600 is input to the resampler 602. The resampler 602 also receives the far-end signal 110 as input. The output from the resampler 602 is input to the far-end buffer 604. The linear adaptive filter 606 receives, as input, the far-end 122 and the output signal of the resampler 602 via the far-end buffer 604 and outputs the error signal 124.

[00120] The NLP 608 receives three signals as input. It receives the far-end signal 110 via the far-end buffer 604, the near-end signal 122 directly output from the capture device 20, and the error signal 124 output from the linear adaptive filter 606. The NLP 608 outputs an outgoing signal directed to a far-end device (not shown).

[00121] As mentioned earlier, especially with USB devices, the clock frequency of the render and capture devices differs. This causes a mismatch between the rate of samples rendered and captured, and consequently results in a drift between the far-end signal 110 and the near-end signal 120 as they arrive at the AEC 100.

[00122] According to an embodiment of the present invention, the linear adaptive filter 606 is configured to adapt to the echo path on a sample level. A bulk delay may be tolerable, but this delay should not change over time. If the far-end signal 110 and the near-end signal 120 are drifting, it indicates that the delay is changing over time.

[00123] To compensate for the above-noted drifting problem, the AEC 100, in accordance with an embodiment of the invention, receives information from the sound devices about the number of samples rendered and captured with every frame. For instance, if the capture device's clock frequency is faster than the render device's, there will be, on average, more recorded samples than played-out samples.

[00124] The drift estimator 600 takes as input the number of samples rendered and captured during a frame period. This information is obtained from an audio hardware abstraction layer (HAL) of the operating system. The cumulative difference between these values is stored as a time-series. Using this information, the drift estimator 600 estimates the drift using linear regression.

[00125] According to an embodiment of the present invention, sample counts have been utilized herein by the drift estimator 600 because they are readily available through the Windows Wave API. Alternatively, however, timestamps of rendered and captured frames or other forms of sample rate information could conceivably be used by the drift estimator 600. Further, different methods for estimating a slope of the drift time-series could also be used, such as a repeated median method rather than linear regression.

[00126] According to an embodiment of the present invention the resampler 602 may be placed in the near-end path or earlier in the far-end path such that it affects the rendered signal of the render device 10 (render device). According to an exemplary embodiment, the resampler 602 may apply resampling in the far-end signal path bound for the AEC 100 and not the near-end signal path bound for rendering. In addition, the resampler 602 may disable the resampling when the estimated drift is less than a first predetermined threshold or greater than a second predetermined threshold. The first predetermined threshold could be, for example, equal to 0.001. The second predetermined threshold could be, for example, equal to -0.001. These exemplary threshold values are not intended to limit the scope of the disclosure in any way. Instead, numerous other threshold values may be used in addition to or instead of those used in the various examples described herein.

[00127] With a reliable drift estimate, the far-end signal 110 is resampled by the resampler 602 in compensation, such that the far-end signal 110 and the near-end signal 122 arriving at the AEC 100 no longer drift away from one another. According to an embodiment of the present invention, a linear resampling technique is used. Other methods, such as a sinc

resampling technique could also be used without departing from the scope of the present invention.

[00128] The difference between subsequent points in the time-series is a noisy estimate of the clock drift. A reliable estimate can be obtained by the drift estimator 600 by estimating the slope of the time-series trend (the trend is linear because in practice the drift rate does not change). As mentioned earlier, according to an embodiment of the present invention linear regression is used.

[00129] Large "glitches", or outliers, may sometimes appear in the time-series. This is addressed by analyzing the time-series to produce a rejection threshold. Data points lying outside this rejection threshold are pruned from the data set before drift estimation occurs as described in Fig. 7 below.

[00130] According to an exemplary embodiment of the present invention, the drift estimator 600 may prune the data set by rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

[00131] According to another exemplary embodiment of the present invention, the drift estimator 600 may prune by rejecting outliers greater than a first threshold determined from the first data set and the second data set or less than a second threshold determined from the first data set and the second data set.

[00132] The first predetermined threshold could be, for example, equal to 4% of the nominal sampling rate of the audio devices. The second predetermined threshold could be, for example, equal to negative 4% of the nominal sampling rate of the audio devices. These exemplary threshold values are not intended to limit the scope of the disclosure in any way. Instead, numerous other threshold values may be used in addition to or instead of those used in the various examples described herein.

[00133] According to a further exemplary embodiment of the present invention, the first threshold and the second threshold are proportional to a measure of spread of the first data set and the second data set. A measure of spread may describe the variability or dispersion present in a data set. In other words, the measure of spread may give a notion of how "spread out" the data is. According to an exemplary embodiment, the measure of spread may be an absolute deviation.

[00134] As mentioned earlier, the drift estimate output by drift estimator 600 is input to the resampler 602. The far-end signal 110 is resampled to precisely counteract the drift between the far-end signal 110 and the near-end signal 120. The linear filter stage, as mentioned earlier with respect to Fig. 5 no longer must adapt to the changing delay and can converge as intended. Preferably, this operates only on the signal passed to the AEC 100, where the resampling distortion will not impact the rendered signal. This allows the resampler 602 to perform a crude but very computationally efficient linear resampling. Although the signal may become highly distorted, this negligibly impacts AEC performance relative to performing a high quality transparent resampling.

[00135] Fig. 7 is a flow diagram illustrating operations performed by the AEC 100 and the drift estimator 600 according to an embodiment of the present invention illustrated in Fig. 6. At step S701, the drift estimator 600 receives, as input, for a predetermined amount of time, a first data set corresponding to the rate at which samples are rendered at the far-end and a second data set corresponding to the rate at which samples are captured at the near-end. At step S703, a one-time estimate of a difference between the first data set and second data set is produced during the predetermined amount of time to estimate the drift between the far-end signal 110 and the near-end signal 120. Optionally, at step S702, samples from both first and second data sets may be pruned and linear regression may be applied prior to producing the one-time estimate. At step S705, the far-end signal 110 is resampled according to the estimated drift to compensate for the drift between the far-end signal 110 and near-end signal 120.

[00136] Fig. 8 is a block diagram illustrating an example computing device 800 that may be utilized to implement the AEC 100 including, but not limited to, the NLP 104, 608, the filter 102, 606, the far-end buffer 106 and the blocking buffer 108, 604 as well as the processes illustrated in Figs. 3, 5, and 7 in accordance with the present disclosure. In a very basic configuration 801, computing device 800 typically includes one or more processors 810 and system memory 820. A memory bus 830 can be used for communicating between the processor 810 and the system memory 820.

[00137] Depending on the desired configuration, processor 810 can be of any type including but not limited to a microprocessor (μ P), a microcontroller (μ C), a digital signal processor (DSP), or any combination thereof. Processor 810 can include one more levels of

caching, such as a level one cache 811 and a level two cache 812, a processor core 813, and registers 814. The processor core 813 can include an arithmetic logic unit (ALU), a floating point unit (FPU), a digital signal processing core (DSP Core), or any combination thereof. A memory controller 815 can also be used with the processor 810, or in some implementations the memory controller 815 can be an internal part of the processor 810.

[00138] Depending on the desired configuration, the system memory 820 can be of any type including but not limited to volatile memory (such as RAM), non-volatile memory (such as ROM, flash memory, etc.) or any combination thereof. System memory 820 typically includes an operating system 821, one or more applications 822, and program data 824. Application 822 includes a clock drift compensation algorithm 823 that is arranged to compensate for a drift between a far-end signal and a near-end signal. Program Data 824 includes clock drift compensation routing data 825 that is useful for compensating a drift between a far-end signal and a near-end signal, as will be further described below. In some embodiments, application 822 can be arranged to operate with program data 824 on an operating system 821 such that a drift between a far-end signal and a near-end signal. This described basic configuration is illustrated in FIG. 8 by those components within dashed line 801.

[00139] Computing device 800 can have additional features or functionality, and additional interfaces to facilitate communications between the basic configuration 801 and any required devices and interfaces. For example, a bus/interface controller 840 can be used to facilitate communications between the basic configuration 801 and one or more data storage devices 850 via a storage interface bus 841. The data storage devices 850 can be removable storage devices 851, non-removable storage devices 852, or a combination thereof. Examples of removable storage and non-removable storage devices include magnetic disk devices such as flexible disk drives and hard-disk drives (HDD), optical disk drives such as compact disk (CD) drives or digital versatile disk (DVD) drives, solid state drives (SSD), and tape drives to name a few. Example computer storage media can include volatile and nonvolatile, 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.

[00140] System memory 820, removable storage 851 and non-removable storage 852

are all examples of computer storage media. Computer storage media includes, but is not limited to, 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 computing device 800. Any such computer storage media can be part of device 800.

[00141] Computing device 800 can also include an interface bus 842 for facilitating communication from various interface devices (e.g., output interfaces, peripheral interfaces, and communication interfaces) to the basic configuration 801 via the bus/interface controller 840. Example output devices 860 include a graphics processing unit 861 and an audio processing unit 862, which can be configured to communicate to various external devices such as a display or speakers via one or more A/V ports 863. Example peripheral interfaces 870 include a serial interface controller 871 or a parallel interface controller 872, which can be configured to communicate with external devices such as input devices (e.g., keyboard, mouse, pen, voice input device, touch input device, etc.) or other peripheral devices (e.g., printer, scanner, etc.) via one or more I/O ports 873. An example communication device 880 includes a network controller 881, which can be arranged to facilitate communications with one or more other computing devices 890 over a network communication via one or more communication ports 882. The communication connection is one example of a communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. A “modulated data signal” can be a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media can include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), infrared (IR) and other wireless media. The term computer readable media as used herein can include both storage media and communication media.

[00142] Computing device 800 can be implemented as a portion of a small-form factor portable (or mobile) electronic device such as a cell phone, a personal data assistant (PDA), a personal media player device, a wireless web-watch device, a personal headset device, an application specific device, or a hybrid device that include any of the above functions.

Computing device 800 can also be implemented as a personal computer including both laptop computer and non-laptop computer configurations.

[00143] There is little distinction left between hardware and software implementations of aspects of systems; the use of hardware or software is generally (but not always, in that in certain contexts the choice between hardware and software can become significant) a design choice representing cost vs. efficiency tradeoffs. There are various vehicles by which processes and/or systems and/or other technologies described herein can be effected (e.g., hardware, software, and/or firmware), and that the preferred vehicle will vary with the context in which the processes and/or systems and/or other technologies are deployed. For example, if an implementer determines that speed and accuracy are paramount, the implementer may opt for a mainly hardware and/or firmware vehicle; if flexibility is paramount, the implementer may opt for a mainly software implementation; or, yet again alternatively, the implementer may opt for some combination of hardware, software, and/or firmware.

[00144] The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof.

[00145] In one embodiment, several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, can be equivalently implemented in integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of

skill in the art in light of this disclosure.

[00146] In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies regardless of the particular type of signal bearing medium used to actually carry out the distribution. Examples of a signal bearing medium include, but are not limited to, the following: a recordable type medium such as a floppy disk, a hard disk drive, a Compact Disc (CD), a Digital Video Disk (DVD), a digital tape, a computer memory, etc.; and a transmission type medium such as a digital and/or an analog communication medium (e.g., a fiber optic cable, a waveguide, a wired communications link, a wireless communication link, etc.).

[00147] Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth herein, and thereafter use engineering practices to integrate such described devices and/or processes into data processing systems. That is, at least a portion of the devices and/or processes described herein can be integrated into a data processing system via a reasonable amount of experimentation.

[00148] Those having skill in the art will recognize that a typical data processing system generally includes one or more of a system unit housing, a video display device, a memory such as volatile and non-volatile memory, processors such as microprocessors and digital signal processors, computational entities such as operating systems, drivers, graphical user interfaces, and applications programs, one or more interaction devices, such as a touch pad or screen, and/or control systems including feedback loops and control motors (e.g., feedback for sensing position and/or velocity; control motors for moving and/or adjusting components and/or quantities). A typical data processing system may be implemented utilizing any suitable commercially available components, such as those typically found in data computing/communication and/or network computing/communication systems.

[00149] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

[00150] While various aspects and embodiments have been disclosed herein, other aspects

and embodiments will be apparent to those skilled in the art. The various aspects and embodiments disclosed herein are for purposes of illustration and are not intended to be limiting, with the true scope and spirit being indicated by the following claims.

What is claimed is:

1. A method to compensate for a drift between a first signal and a second signal, comprising:
 - receiving, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of said first signal and a second data set containing information related to a sample rate of said second signal;
 - producing a one-time estimate of a drift between said first and second signals based on said first data set and said second data set during said predetermined amount of time; and
 - resampling one, or both, of said first and second signals according to said estimated drift to compensate for the drift between said first and second signals.
2. The method according to claim 1, further comprising performing echo cancellation where said first signal is a far-end signal to be rendered and said second signal is a near-end captured signal.
3. The method according to claim 2, further comprising performing echo suppression with said near-end signal and said far-end signal.
4. The method according to claim 1, wherein said information related to the sample rate of said first signal includes counts of number of samples rendered per frame and said information related to the sample rate of said second signal includes counts of number of samples captured per frame.
5. The method according to claim 1, wherein said information related to the sample rate of said first signal includes timestamps indicating when frames are rendered and said information related to the sample rate of said second signal includes timestamps indicating when frames are captured.
6. The method according to any of claims 1-5, further comprising resampling said first signal.
7. The method according to any of claims 1-6, wherein said resampling is applied in a

first signal path bound for an echo canceller and not a second signal path bound for rendering.

8. The method according to any of claims 1-7, wherein said resampling step utilizes a linear resampling method.

9. The method according to any of claims 1-8, further comprising pruning outliers from said first data set and second data set prior to producing said estimate.

10. The method according to any of claims 1-9, further comprising pruning based on rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

11. The method according to any of claims 1-9, further comprising pruning based on rejecting outliers greater than a first threshold determined from said first data set and said second data set or less than a second threshold determined from said first data set and said second data set.

12. The method according to any of claims 10-11, wherein said first threshold and said second threshold are proportional to a measure of spread of said first data set and said second data set.

13. The method according to claim 12, wherein said measure of spread is an absolute deviation.

14. The method according to claim 1, further comprising utilizing linear regression to produce said estimate.

15. The method according to claim 1, further comprising disabling said resampling when said estimated drift is less than a first predetermined threshold or greater than a second predetermined threshold.

16. The method according to any of claims 1-15, further comprising:
 - computing a first coherence measure by comparing correlations between the first signal and the second signal;
 - computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and
 - applying the first and second coherence measures to compute echo cancellation information.
17. The method according to any of claims 1-15, further comprising:
 - computing a first coherence measure by comparing correlations between the first signal and the second signal;
 - computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and
 - applying the first and second coherence measures to compute suppression factors.
18. The method according to claim 17 wherein said suppression factors are applied to said error signal to substantially reduce the echo in said error signal.
19. A system to compensate for a drift between a first signal and a second signal, comprising:
 - a drift estimator that receives, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of said first signal and a second data set containing information related to a sample rate of said second signal, and produces a one-time estimate of a drift between said first and second signals based on said first data set and said second data set during said predetermined amount of time; and
 - a resampler operatively connected to said drift estimator, said resampler resampling one, or both, of said first and second signals according to said estimated drift to compensate for the drift between said first and second signals.
20. The system according to claim 19, further comprising an echo canceller operatively

connected to said resampler, said echo canceller performing echo cancellation where said first signal is a far-end signal to be rendered and said second signal is a near-end captured signal.

21. The system according to claim 20, wherein said echo canceller performing echo suppression with said near-end signal and said far-end signal.

22. The system according to claim 19, wherein said information related to the sample rate of said first signal includes counts of number of samples rendered per frame and said information related to the sample rate of said second signal includes counts of number of samples captured per frame.

23. The system according to claim 19, wherein said information related to the sample rate of said first signal includes timestamps indicating when frames are rendered and said information related to the sample rate of said second signal includes timestamps indicating when frames are captured.

24. The system according to any of claims 19-23, wherein said resampler resamples said first signal.

25. The system according to any of claims 19-24, wherein said resampler applying resampling in a first signal path bound for an echo canceller and not a second signal path bound for rendering.

26. The system according to any of claims 19-25, wherein said resampler utilizes a linear resampling method.

27. The system according to any of claims 19-26, wherein said drift estimator is configured to prune outliers from said first data set and second data set prior to producing said estimate.

28. The system according to any of claims 19-27, wherein said drift estimator is

configured to prune by rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

29. The system according to any of claims 19-27, wherein said drift estimator is configured to prune by rejecting outliers greater than a first threshold determined from said first data set and said second data set or less than a second threshold determined from said first data set and said second data set.

30. The system according to any of claims 28-29, wherein said first threshold and said second threshold are proportional to a measure of spread of said first data set and said second data set.

31. The system according to claim 30, wherein said measure of spread is an absolute deviation.

32. The system according to claim 19, wherein said drift estimator is configured to apply linear regression to produce said estimate.

33. The system according to claim 19, wherein said resampler is configured to disable said resampling when said estimated drift is less than a first predetermined threshold or greater than a second predetermined threshold.

34. The system according to any of claims 19-33, further comprising a non-linear processor operatively connected to said resampler, wherein said non-linear processor is configured to:

- compute a first coherence measure by comparing correlations between the first signal and the second signal;

- compute a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

- apply the first and second coherence measures to compute echo cancellation information.

35. The system according to any of claims 19-33, further comprising a non-linear processor operatively connected to said resampler, wherein said non-linear processor is configured to:

compute a first coherence measure by comparing correlations between the first signal and the second signal;

compute a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

apply the first and second coherence measures to compute suppression factors.

36. The system according to claim 35, wherein said non-linear processor is configured to apply said suppression factors to said error signal to substantially reduce the echo in said error signal.

37. A computer-readable storage medium having stored thereon computer executable program to compensate for a drift between a first signal and a second signal, the computer program when executed causes a processor to execute the steps of:

receiving, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of said first signal and a second data set containing information related to a sample rate of said second signal;

producing a one-time estimate of a drift between said first and second signals based on said first data set and said second data set during said predetermined amount of time; and

resampling one, or both, of said first and second signals according to said estimated drift to compensate for the drift between said first and second signals.

38. The computer-readable storage medium of claim 37, wherein the computer program when executed causes the processor to further execute the step of performing echo cancellation where said first signal is a far-end signal to be rendered and said second signal is a near-end captured signal.

39. The computer-readable storage medium of claim 38, wherein the computer program when executed causes the processor to further execute the step of performing echo suppression with said near-end signal and said far-end signal.

40. The computer-readable storage medium of claim 37, wherein said information related to the sample rate of said first signal includes counts of number of samples rendered per frame and said information related to the sample rate of said second signal includes counts of number of samples captured per frame.

41. The computer-readable storage medium of claim 37, wherein said information related to the sample rate of said first signal includes timestamps indicating when frames are rendered and said information related to the sample rate of said second signal includes timestamps indicating when frames are captured.

42. The computer-readable storage medium of any of claims 37-41, wherein the computer program when executed causes the processor to further execute the step of resampling said first signal.

43. The computer-readable storage medium of any of claims 37-42, wherein the computer program when executed causes the processor to further execute the step of applying resampling in a first signal path bound for an echo canceller and not a second signal path bound for rendering.

44. The computer-readable storage medium of any of claims 37-43, wherein the computer program when executed causes the processor to further execute the step of performing resampling by applying a linear resampling method.

45. The computer-readable storage medium of any of claims 37-44, wherein the computer program when executed causes the processor to further execute the step of pruning outliers from said first data set and second data set prior to producing said estimate.

46. The computer-readable storage medium of any of claims 37-45, wherein the computer program when executed causes the processor to further execute the step of pruning based on rejecting outliers greater than a first predetermined threshold or less than a second predetermined threshold.

47. The computer-readable storage medium of any of claims 37-45, wherein the computer program when executed causes the processor to further execute the step of pruning based on rejecting outliers greater than a first threshold determined from said first data set and said second data set or less than a second threshold determined from said first data set and said second data set.

48. The computer-readable storage medium of any of claims 46-47, wherein said first threshold and said second threshold are proportional to a measure of spread of said first data set and said second data set.

49. The computer-readable storage medium of claim 48, wherein said measure of spread is an absolute deviation.

50. The computer-readable storage medium of claim 37, wherein the computer program when executed causes the processor to further execute the step of utilizing linear regression to produce said estimate.

51. The computer-readable storage medium of claim 37, wherein the computer program when executed causes the processor to further execute the step of disabling said resampling when said estimated drift is less than a first predetermined threshold or greater than a second predetermined threshold.

52. The computer-readable storage medium of any of claims 37-51, wherein the computer program when executed causes the processor to further execute the steps of:

 computing a first coherence measure by comparing correlations between the first signal and the second signal;

 computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

 applying the first and second coherence measures to compute echo cancellation information.

53. The computer-readable storage medium of any of claims 37-51, wherein the computer program when executed causes the processor to further execute the steps of:

computing a first coherence measure by comparing correlations between the first signal and the second signal;

computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

applying the first and second coherence measures to compute suppression factors.

54. The computer-readable storage medium of claim 53, wherein the computer program when executed causes the processor to further execute the step of applying said suppression factors to said error signal to substantially reduce the echo in said error signal.

AMENDED CLAIMS

received by the International Bureau on 21 August 2012 (21.08.2012)

1. A method to compensate for a drift between a first signal and a second signal, comprising:

receiving, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of said first signal and a second data set containing information related to a sample rate of said second signal;

estimating a drift between said first and second signals based on said first data set and said second data set during said predetermined amount of time; and

responsive to determining that the estimated drift is greater than a predetermined threshold value, resampling one, or both, of said first and second signals according to said estimated drift to compensate for the drift between said first and second signals, wherein the resampling is applied in a signal path bound for an echo canceller.

2. The method according to claim 1, further comprising performing echo cancellation where said first signal is a far-end signal to be rendered and said second signal is a captured near-end signal.

3. The method according to claim 2, further comprising performing echo suppression with said near-end signal and said far-end signal.

4. The method according to claim 1, wherein said information related to the sample rate of said first signal includes number of samples rendered per frame and said information related to the sample rate of said second signal includes number of samples captured per frame.

5. The method according to claim 1, wherein said information related to the sample rate of said first signal includes timestamps for rendered frames and said information related to the sample rate of said second signal includes timestamps for captured frames.

6. The method according to any of claims 1-5, wherein the drift between said first and second signals is estimated using a repeated median method.

7. The method according to any of claims 1-6, wherein said resampling step utilizes a sinc resampling technique.
8. The method according to any of claims 1-6, wherein said resampling step utilizes a linear resampling method.
9. The method according to any of claims 1-8, further comprising pruning outliers from said first data set and said second data set prior to estimating the drift.
10. The method according to claim 9, wherein said pruning step includes rejecting outliers greater than a rejection threshold determined from said first data set and said second data set.
11. The method according to claim 9, wherein said pruning step includes rejecting outliers less than a rejection threshold determined from said first data set and said second data set.
12. The method according to claim 10 or 11, wherein said rejection threshold is proportional to a measure of spread of said first data set and said second data set.
13. The method according to claim 12, wherein said measure of spread is an absolute deviation.
14. The method according to claim 1, further comprising utilizing linear regression to produce said estimate.
15. The method according to claim 1, further comprising disabling said resampling when said estimated drift is less than the predetermined threshold value.
16. The method according to any of claims 1-15, further comprising:
computing a first coherence measure by comparing correlations between the first signal and the second signal;

computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

applying the first and second coherence measures to compute echo cancellation information.

17. The method according to any of claims 1-15, further comprising:

computing a first coherence measure by comparing correlations between the first signal and the second signal;

computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

applying the first and second coherence measures to compute suppression factors.

18. The method according to claim 17 wherein said suppression factors are applied to said error signal to reduce the echo in said error signal.

19. A system to compensate for a drift between a first signal and a second signal, said system comprising:

a drift estimator that receives, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of said first signal and a second data set containing information related to a sample rate of said second signal, and estimates a drift between said first and second signals based on said first data set and said second data set during said predetermined amount of time; and

a resampler operatively connected to said drift estimator, said resampler configured to, in response to determining that the estimated drift is greater than a predetermined threshold value, resample one, or both, of said first and second signals according to said estimated drift to compensate for the drift between said first and second signals, wherein the resampling is applied in a signal path bound for echo cancellation processing.

20. The system according to claim 19, further comprising an echo canceller operatively connected to said resampler, said echo canceller configured to perform echo cancellation

where said first signal is a far-end signal to be rendered and said second signal is a captured near-end signal.

21. The system according to claim 20, wherein said echo canceller performing echo suppression with said near-end signal and said far-end signal.

22. The system according to claim 19, wherein said information related to the sample rate of said first signal includes number of samples rendered per frame and said information related to the sample rate of said second signal includes number of samples captured per frame.

23. The system according to claim 19, wherein said information related to the sample rate of said first signal includes timestamps for rendered frames, and said information related to the sample rate of said second signal includes timestamps for captured frames.

24. The system according to any of claims 19-23, wherein said drift estimator estimates the drift between said first and second signals using a repeated median method.

25. The system according to any of claims 19-24, wherein said resampler resamples one or both of said first and second signals using a sinc resampling technique.

26. The system according to any of claims 19-24, wherein said resampler resamples one or both of said first and second signals using a linear resampling method.

27. The system according to any of claims 19-26, wherein said drift estimator is configured to prune outliers from said first data set and said second data set prior to estimating the drift.

28. The system according to claim 27, wherein said drift estimator is configured to prune by rejecting outliers greater than a rejection threshold determined from said first and second data sets.

29. The system according to claim 27, wherein said drift estimator is configured to prune by rejecting outliers less than a rejection threshold determined from said first data set and said second data set.
30. The system according to claim 28 or 29, wherein said rejection threshold is proportional to a measure of spread of said first data set and said second data set.
31. The system according to claim 30, wherein said measure of spread is an absolute deviation.
32. The system according to claim 19, wherein said drift estimator is configured to apply linear regression to produce said estimate.
33. The system according to claim 19, wherein said resampler is configured to disable said resampling when said estimated drift is less than the predetermined threshold value.
34. The system according to any of claims 19-33, further comprising a non-linear processor operatively connected to said resampler, wherein said non-linear processor is configured to:
- compute a first coherence measure by comparing correlations between the first signal and the second signal;
 - compute a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and
 - apply the first and second coherence measures to compute echo cancellation information.
35. The system according to any of claims 19-33, further comprising a non-linear processor operatively connected to said resampler, wherein said non-linear processor is configured to:
- compute a first coherence measure by comparing correlations between the first signal and the second signal;
 - compute a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

apply the first and second coherence measures to compute suppression factors.

36. The system according to claim 35, wherein said non-linear processor is configured to apply said suppression factors to said error signal to reduce the echo in said error signal.

37. A computer-readable storage medium having stored thereon computer executable program to compensate for a drift between a first signal and a second signal, the computer program when executed causes a processor to execute the steps of:

receiving, as input, for a predetermined amount of time, a first data set containing information related to a sample rate of said first signal and a second data set containing information related to a sample rate of said second signal;

estimating a drift between said first and second signals based on said first data set and said second data set during said predetermined amount of time; and

responsive to determining that the estimated drift is greater than a predetermined threshold value, resampling one, or both, of said first and second signals according to said estimated drift to compensate for the drift between said first and second signals, wherein the resampling is applied in a signal path bound for an echo canceller.

38. The computer-readable storage medium of claim 37, wherein the computer program when executed causes the processor to further execute the step of performing echo cancellation where said first signal is a far-end signal to be rendered and said second signal is a captured near-end signal.

39. The computer-readable storage medium of claim 38, wherein the computer program when executed causes the processor to further execute the step of performing echo suppression with said near-end signal and said far-end signal.

40. The computer-readable storage medium of claim 37, wherein said information related to the sample rate of said first signal includes number of samples rendered per frame and said information related to the sample rate of said second signal includes number of samples captured per frame.

41. The computer-readable storage medium of claim 37, wherein said information related to the sample rate of said first signal includes timestamps for rendered frames and said information related to the sample rate of said second signal includes timestamps for captured frames.
42. The computer-readable storage medium of any of claims 37-41, wherein the computer program when executed causes the processor to further execute the step of estimating the drift between said first and second signals using a repeated median method.
43. The computer-readable storage medium of any of claims 37-42, wherein the computer program when executed causes the processor to further execute the step of resampling one or both of said first and second signals by applying a sinc resampling technique.
44. The computer-readable storage medium of any of claims 37-42, wherein the computer program when executed causes the processor to further execute the step of resampling one or both of said first and second signals by applying a linear resampling method.
45. The computer-readable storage medium of any of claims 37-44, wherein the computer program when executed causes the processor to further execute the step of pruning outliers from said first data set and said second data set prior to estimating the drift.
46. The computer-readable storage medium of claim 45, wherein the computer program when executed causes the processor to further execute the step of pruning by rejecting outliers greater than a rejection threshold determined from said first data set and said second data set.
47. The computer-readable storage medium of claim 45, wherein the computer program when executed causes the processor to further execute the step of pruning by rejecting outliers less than a rejection threshold determined from said first data set and said second data set.
48. The computer-readable storage medium of claim 46 or 47, wherein said rejection threshold is proportional to a measure of spread of said first data set and said second data set.

49. The computer-readable storage medium of claim 48, wherein said measure of spread is an absolute deviation.

50. The computer-readable storage medium of claim 37, wherein the computer program when executed causes the processor to further execute the step of utilizing linear regression to produce said estimate.

51. The computer-readable storage medium of claim 37, wherein the computer program when executed causes the processor to further execute the step of disabling said resampling when said estimated drift is less than the predetermined threshold value.

52. The computer-readable storage medium of any of claims 37-51, wherein the computer program when executed causes the processor to further execute the steps of:

- computing a first coherence measure by comparing correlations between the first signal and the second signal;

- computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

- applying the first and second coherence measures to compute echo cancellation information.

53. The computer-readable storage medium of any of claims 37-51, wherein the computer program when executed causes the processor to further execute the steps of:

- computing a first coherence measure by comparing correlations between the first signal and the second signal;

- computing a second coherence measure by comparing correlations between the second signal and an error signal containing a residual echo output from a linear adaptive filter; and

- applying the first and second coherence measures to compute suppression factors.

54. The computer-readable storage medium of claim 53, wherein the computer program when executed causes the processor to further execute the step of applying said suppression factors to said error signal to reduce the echo in said error signal.

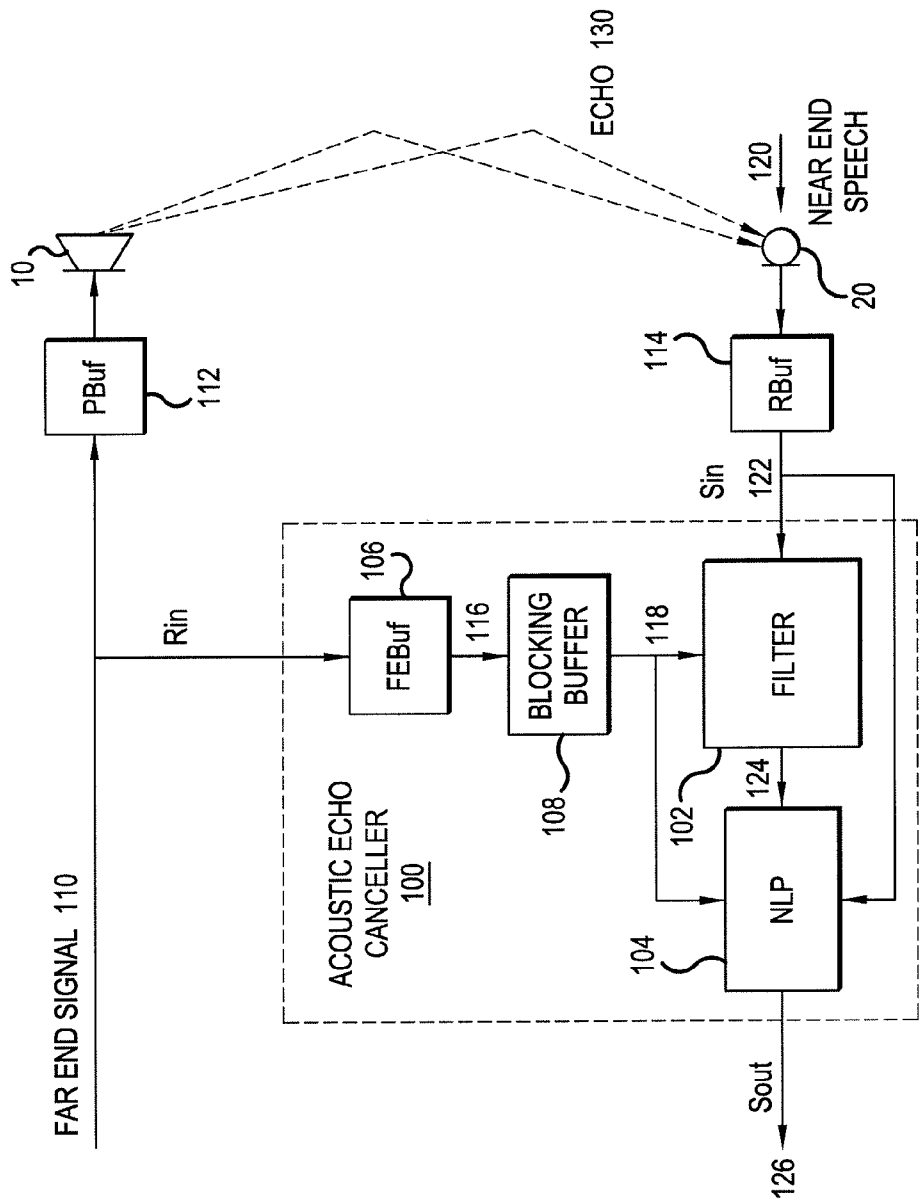


FIGURE 1

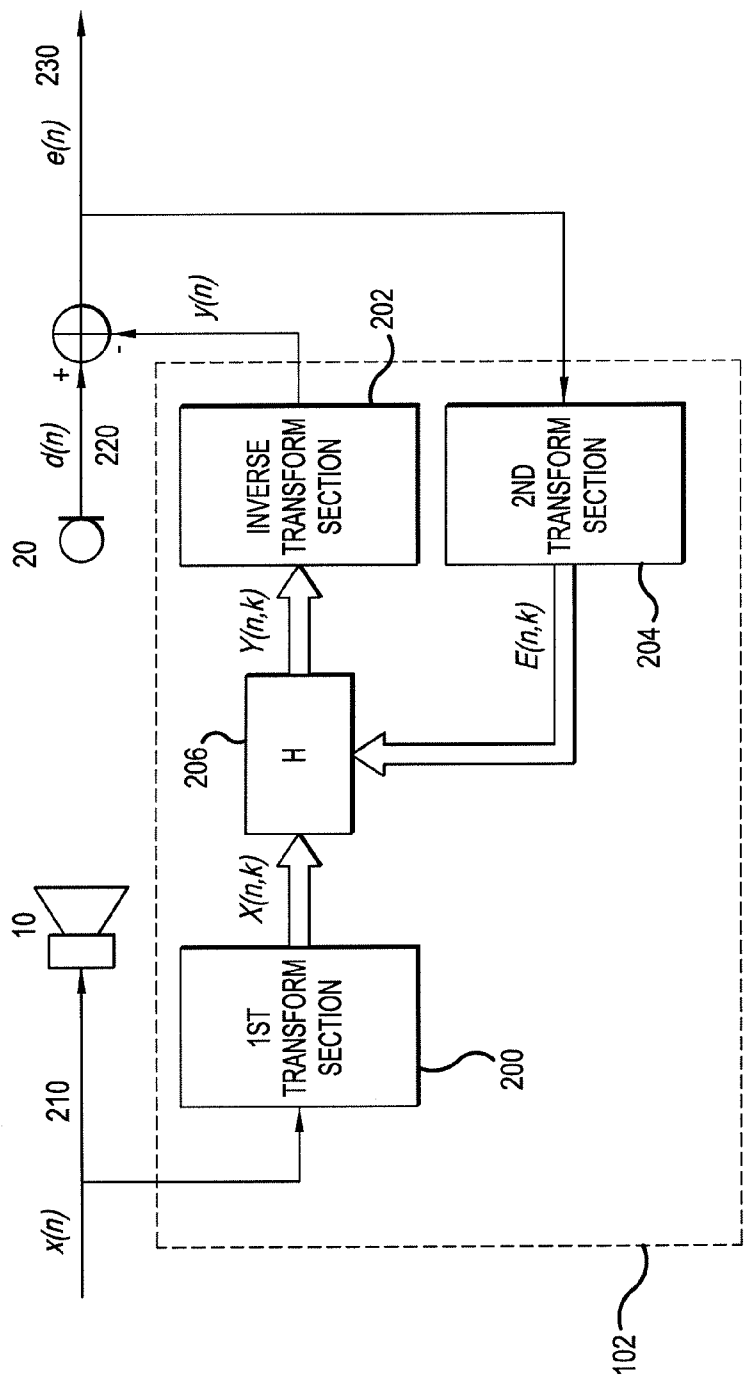


FIGURE 2

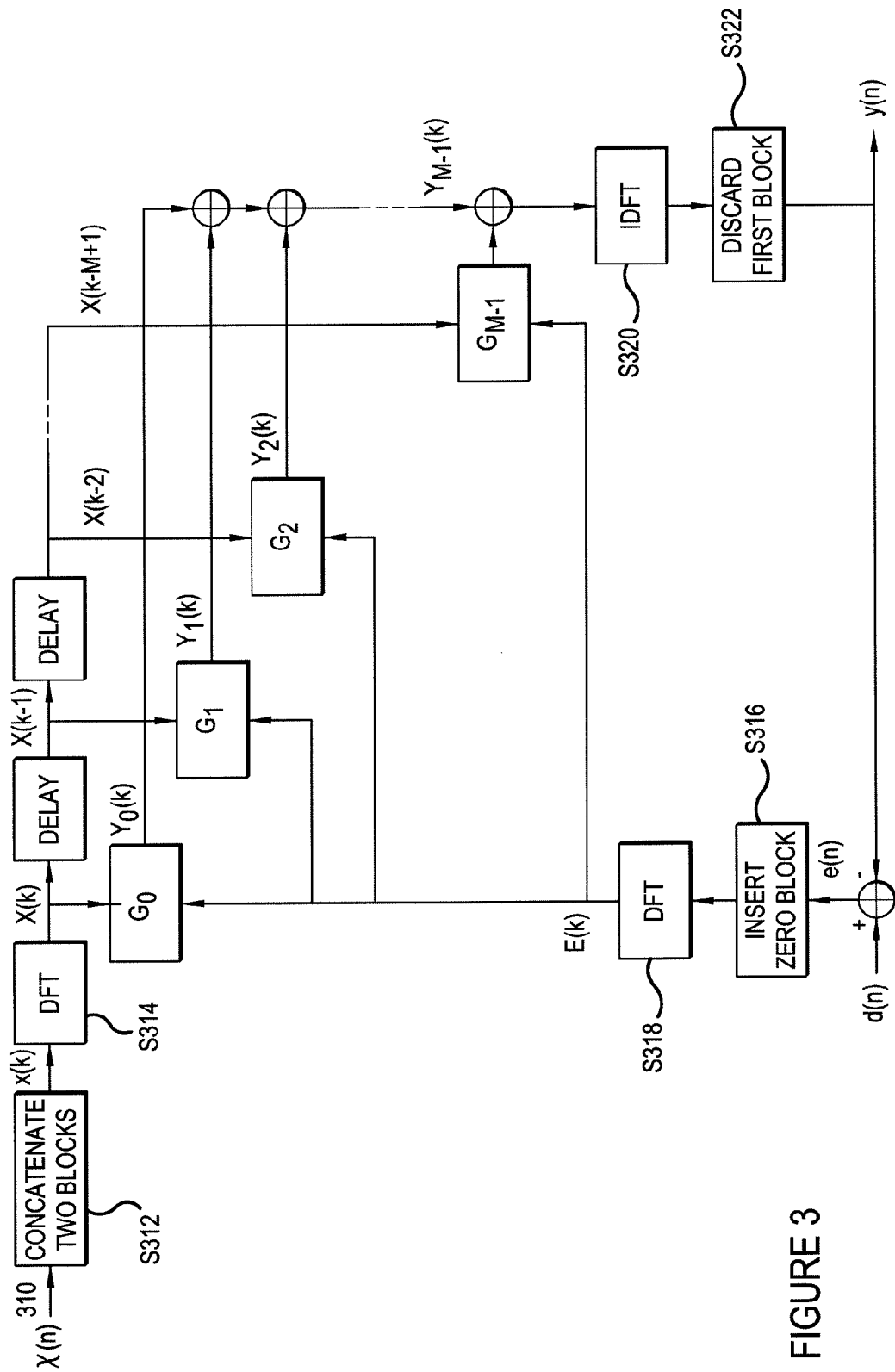


FIGURE 3

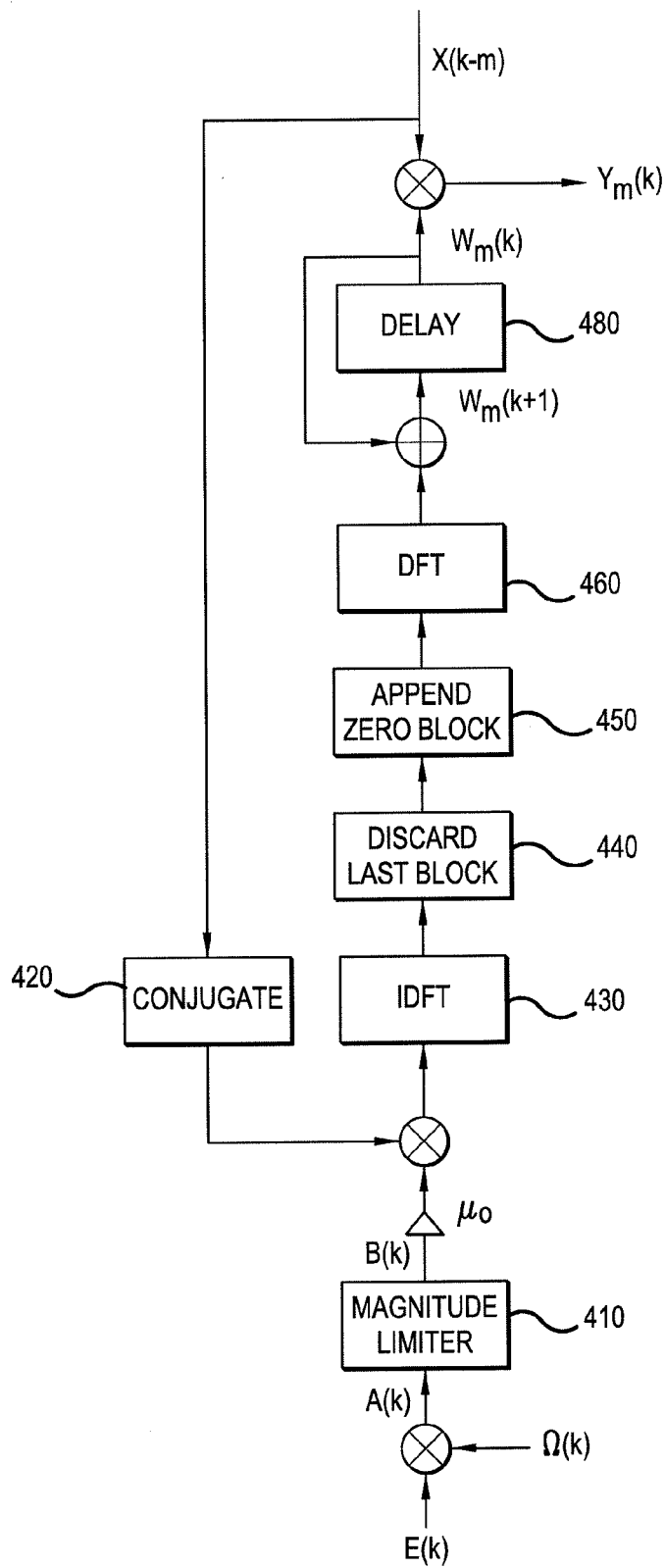


FIGURE 4

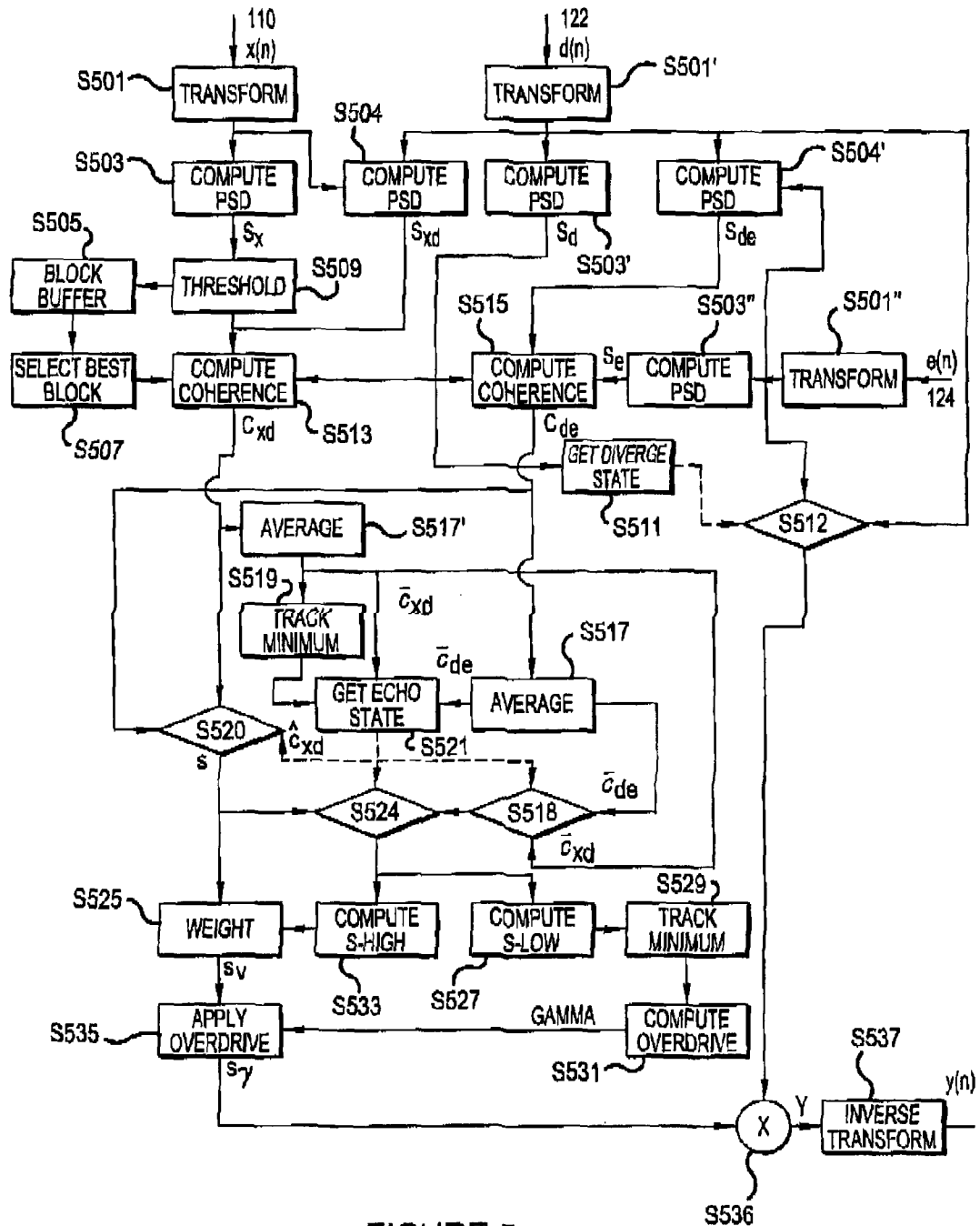


FIGURE 5

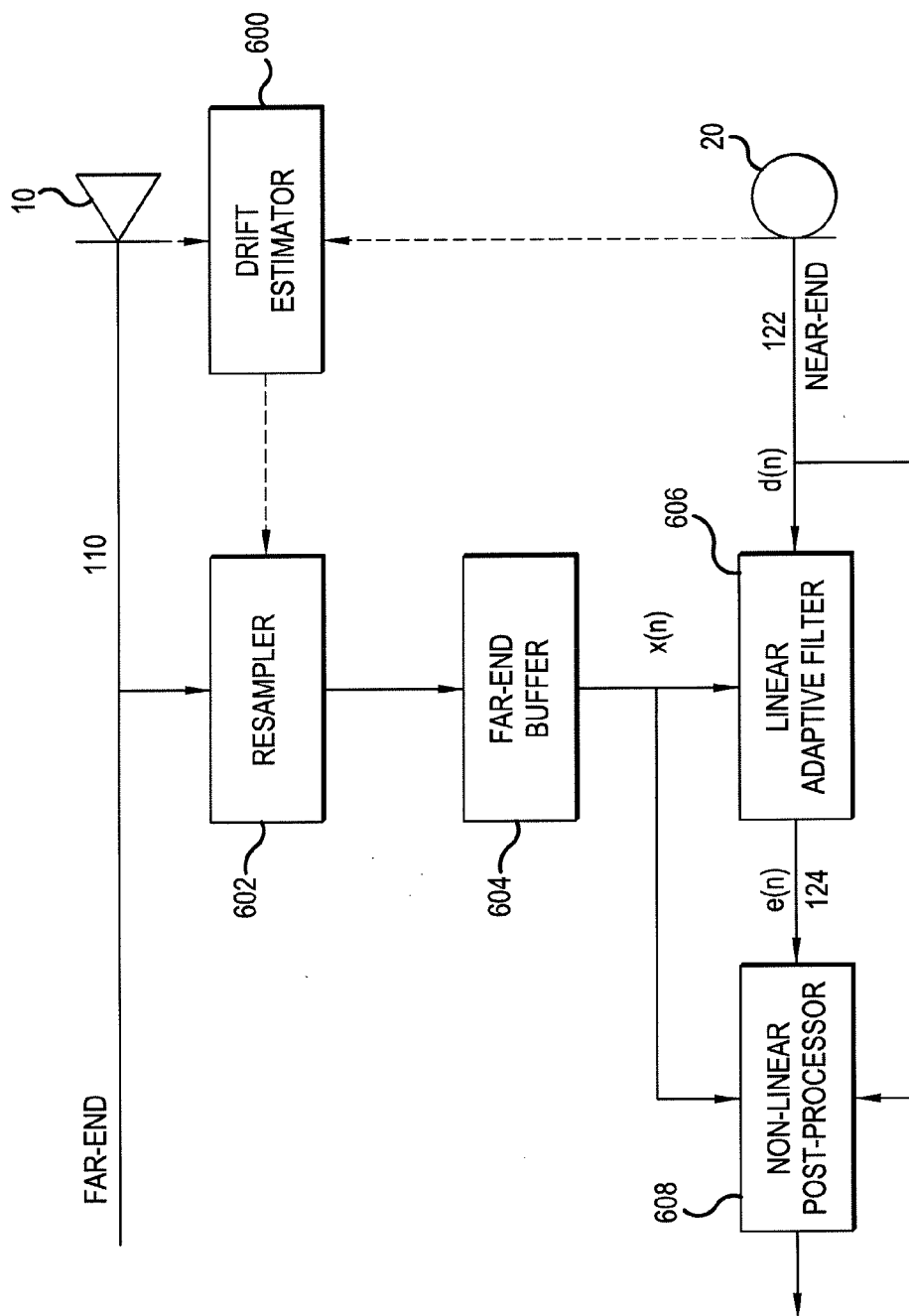


FIGURE 6

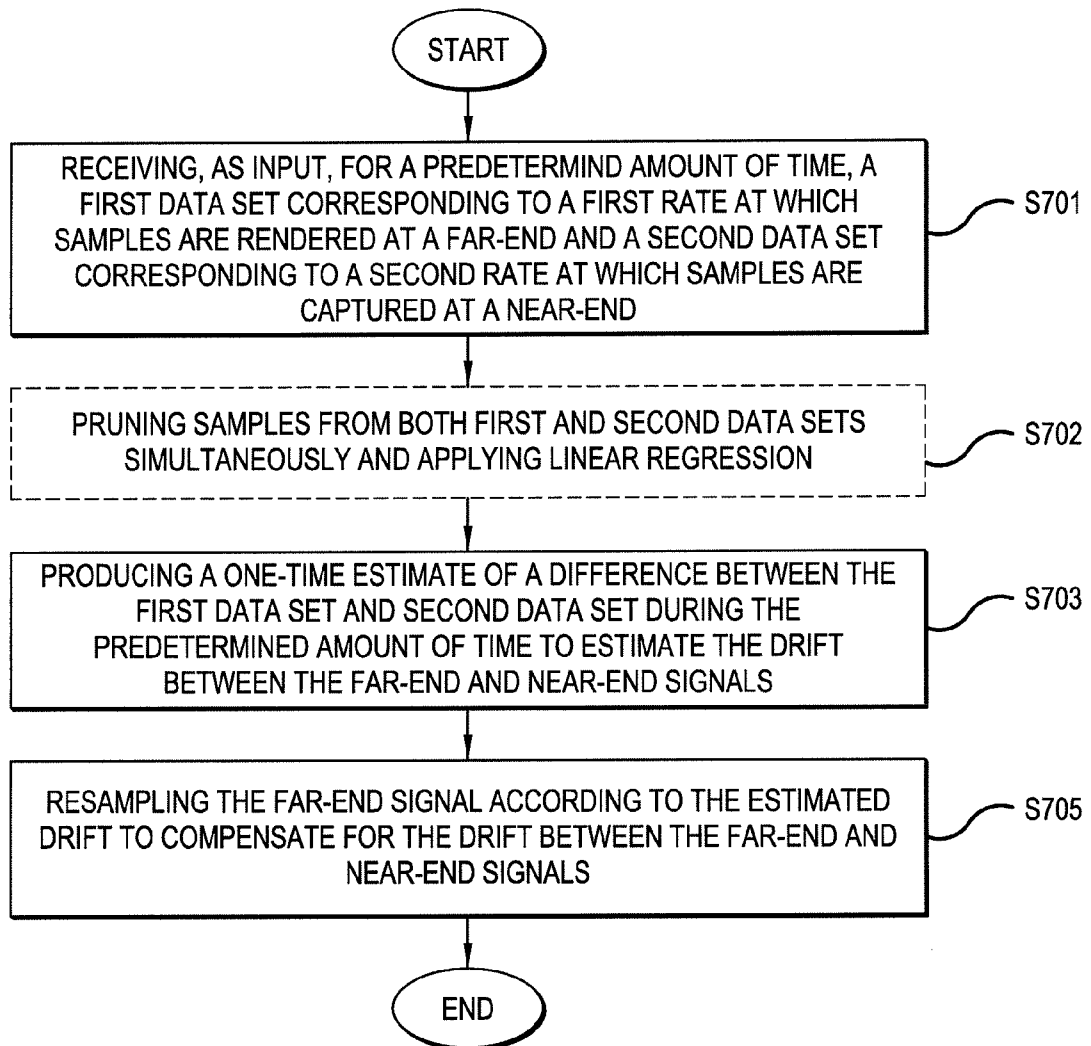


FIGURE 7

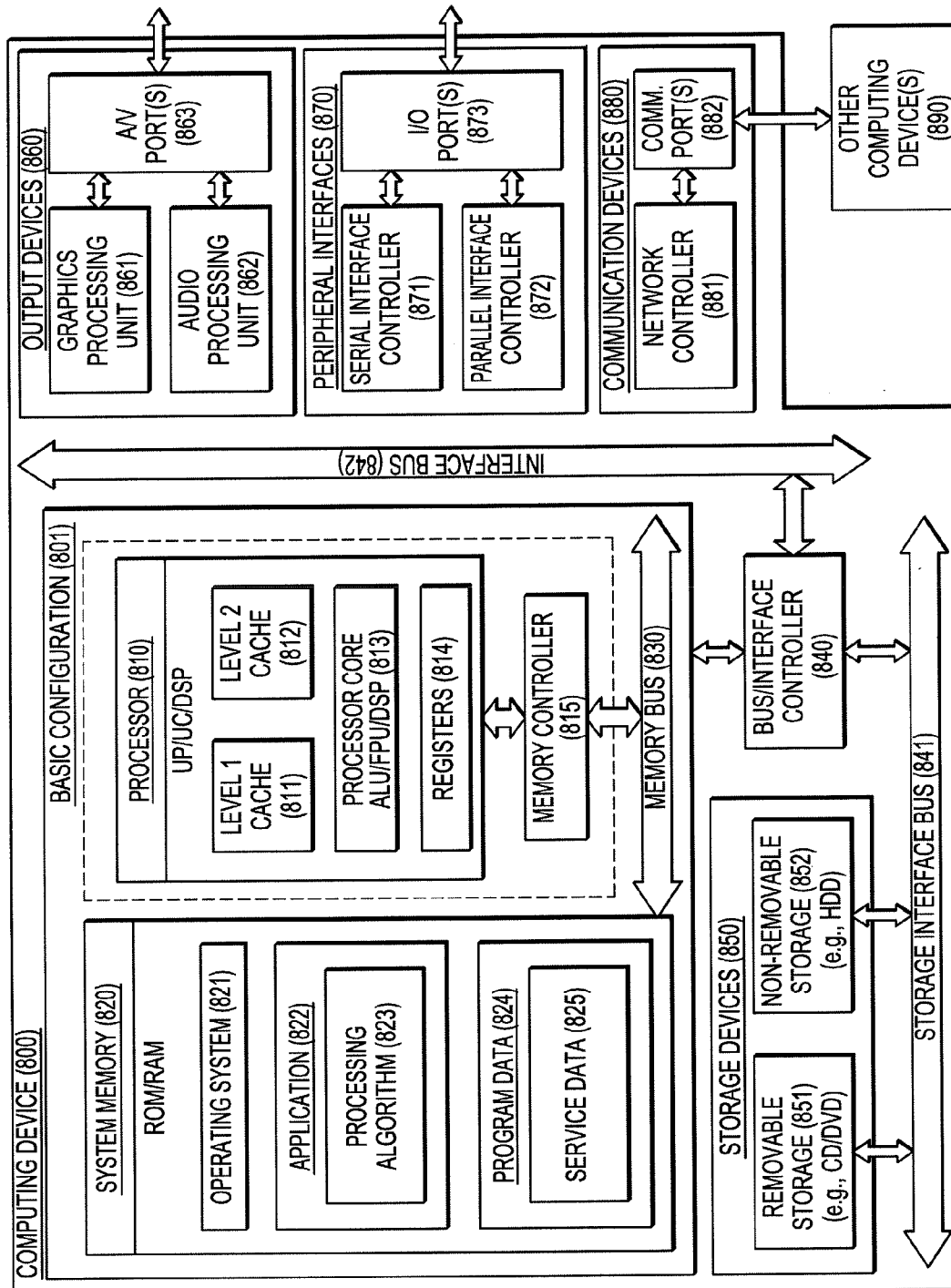


FIGURE 8

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2011/036995

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04M9/08 G10L19/00
ADD.

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)

H04M G10L

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007/165838 A1 (LI QIN [US] ET AL) 19 July 2007 (2007-07-19)	1-6,14, 15, 19-26, 32,33, 37-44, 50,51
A	paragraphs [0004] - [0007], [0012] - [0015], [0026] - [0065], [0085] - [0087] ----- WO 2009/078733 A1 (TANDBERG TELECOM AS [NO]; MARTON TRYGVE FREDRIK [NO]; LIEN TORGEIR GRO) 25 June 2009 (2009-06-25) page 21, line 3 - page 24, line 14 -----	1-6,14, 15, 19-26, 32,33, 37-44, 50,51



Further documents are listed in the continuation of Box C.



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

6 February 2012

Date of mailing of the international search report

18/06/2012

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Sorrentino, Andrea

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/036995

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of 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. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims 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. ☐ Claims 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 additional 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 an 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 claims Nos.:
4. ☒ 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 claims Nos.:

1-6, 14, 15, 19-26, 32, 33, 37-44, 50, 51

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

Information on patent family members

International application No

PCT/US2011/036995

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007165838	A1	19-07-2007	NONE

WO 2009078733	A1	25-06-2009	CN 101843082 A 22-09-2010
			EP 2235928 A1 06-10-2010
			JP 2011512698 A 21-04-2011
			NO 327377 B1 22-06-2009
			US 2009185695 A1 23-07-2009
			WO 2009078733 A1 25-06-2009

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-6, 14, 15, 19-26, 32, 33, 37-44, 50, 51

Method for drift compensation between first and second signal, including estimating a drift between the two signal, resampling one of the two or both to compensate for the drift, using timestamps on both signals.

2. claims: 1, 7-13, 19, 27-31, 37, 45-49

Method for drift compensation between first and second signal, including estimating a drift between the two signal, resampling one of the two or both to compensate for the drift, comprising pruning outliers before estimating the drift.

3. claims: 1, 16-19, 34-37, 52-54

Method for drift compensation between first and second signal, including estimating a drift between the two signal, resampling one of the two or both to compensate for the drift, using coherency measures to compute echo cancellation or suppression.
