

LOW LATENCY AUDIO FILTERBANK HAVING IMPROVED FREQUENCY RESOLUTION

CROSS REFERENCE TO RELATED APPLICATIONS

5

[0001] This application claims priority to United States Provisional Patent Application No. 63/028,966, filed May 22, 2020 and United States Provisional Patent Application No. 62/866,823, filed June 26, 2019, each of which is hereby incorporated by reference in its entirety.

10

FIELD

[0002] The present disclosure relates to audio processing, and in particular, to dynamic audio filters.

15

BACKGROUND

[0003] Unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

20

[0004] When an input audio signal is being processed, it is often desirable to use a filter bank to alter the audio signal such that the gain at various frequencies is specified by a set of dynamic coefficients. A dynamic audio filter may be implemented by making use a number of pre-computed band-pass filters, with the audio filter response being formed from the weighted sum of the band-pass filter responses. The weights may be varied dynamically.

25

[0005] In general, the frequency bands do not have the same bandwidth, but instead the lower frequency filters are narrower (and more closely spaced) than the higher frequency filters. As a consequence, the impulse responses of the higher frequency filters will be more compact (having most of their energy spread over a smaller number of samples) than the impulse responses of the lower frequency filters (having their energy spread over a larger number of samples).

30

SUMMARY

[0006] One issue with existing filter banks is the latency. For the lower frequency filters, in order to accurately represent the impulse response, a defined number of samples must be used, which corresponds to the latency of the filter. The higher frequency filters need not use
5 as many samples as the lower frequency filters, and hence have a lower latency. However, in a filter bank that contains both low and high frequency filters, in order to recombine the filtered bands, the high frequency filters need to be delayed to match the latency of the low frequency filters. Thus the latency of the low frequency filters constrains the overall latency of the filter bank.

10 [0007] Given the above, there is a need to reduce the latency at low frequencies while preserving the impulse response of the filter. Described herein are techniques related to low-latency filter design.

[0008] According to an embodiment, a method of audio processing includes generating a plurality of modified impulse responses from a plurality of ideal impulse responses, wherein
15 the plurality of ideal impulse responses respectively correspond to a plurality of frequencies, wherein generating the plurality of modified impulse responses includes performing a fade operation and a time reverse operation on at least one of the plurality of ideal impulse responses. The method further includes filtering an input signal with the plurality of modified impulse responses to generate an output signal.

20 [0009] Generating the plurality of modified impulse responses may include generating a pre-ripple response based on a first ideal impulse response; generating a post-ripple response based on the pre-ripple response; and adding the first ideal impulse response, subtracting the pre-ripple response, and adding the post-ripple response to generate a first modified impulse response.

25 [0010] Generating the plurality of modified impulse responses may include generating a first filter response based on a first ideal impulse response; generating a pre-ripple response based on the first filter response; generating an intermediate response based on the pre-ripple response; generating a second filter response based on the intermediate response; generating a post-ripple response based on the second filter response; and adding the first filter response,
30 adding the post-ripple response, and subtracting the pre-ripple response to generate a first modified impulse response.

[0011] According to another embodiment, an apparatus includes a processor and a memory. The processor is configured to control the apparatus to generate a plurality of modified impulse responses from a plurality of ideal impulse responses, wherein the plurality of ideal impulse responses respectively correspond to a plurality of frequencies, wherein generating
5 the plurality of modified impulse responses includes performing a fade operation and a time reverse operation on at least one of the plurality of ideal impulse responses. The processor is further configured to control the apparatus to filter an input signal with the plurality of modified impulse responses to generate an output signal. The apparatus may additionally include similar details to those of one or more of the methods described herein.

10 [0012] According to another embodiment, a non-transitory computer readable medium stores a computer program that, when executed by a processor, controls an apparatus to execute processing including one or more of the methods described herein.

[0013] The following detailed description and accompanying drawings provide a further understanding of the nature and advantages of various implementations.

15

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is a block diagram of a filter bank 100.

[0015] FIG. 2 is a graph 200 showing a set of filter bank frequency responses.

[0016] FIG. 3 is a graph 300 showing an example combined filter response 302.

20 [0017] FIG. 4 is a block diagram of a filter bank 400.

[0018] FIG. 5 is a block diagram of a filter bank 500.

[0019] FIG. 6 is a graph of an impulse response 600.

[0020] FIG. 7 is a block diagram of a filter array 700.

25 [0021] FIG. 8 is a graph 800 showing an example set of time-domain filter bank impulse responses.

[0022] FIG. 9 is a graph 900 of signals showing an improved method of modifying impulse responses.

[0023] FIG. 10 is a block diagram of a method 1000 of generating a filter response.

[0024] FIG. 11 is a graph 1100 showing various signals related to FIG. 10.

[0025] FIG. 12 is a set of graphs showing a phase distortion 1202 and a group delay 1204.

[0026] FIG. 13 is a graph 1300 showing various impulse responses.

[0027] FIG. 14 is a block diagram of a method 1400 of generating a filter response.

5 [0028] FIG. 15 is a graph 1500 showing various signals related to FIG. 14.

[0029] FIG. 16 is a flow diagram of a method 1600 of audio processing.

[0030] FIG. 17 is a flow diagram of a method 1700 of audio processing.

[0031] FIG. 18 is a flow diagram of a method 1800 of audio processing.

10

DETAILED DESCRIPTION

[0032] Described herein are techniques related to audio filters. In the following description, for purposes of explanation, numerous examples and specific details are set forth in order to provide a thorough understanding of the present disclosure. It will be evident, however, to one skilled in the art that the present disclosure as defined by the claims may include some or
15 all of the features in these examples alone or in combination with other features described below, and may further include modifications and equivalents of the features and concepts described herein.

20

[0033] In the following description, various methods, processes and procedures are detailed. Although particular steps may be described in a certain order, such order is mainly for convenience and clarity. A particular step may be repeated more than once, may occur before or after other steps (even if those steps are otherwise described in another order), and may occur in parallel with other steps. A second step is required to follow a first step only when the first step must be completed before the second step is begun. Such a situation will be specifically pointed out when not clear from the context.

25

[0034] In this document, the terms “and”, “or” and “and/or” are used. Such terms are to be read as having an inclusive meaning. For example, “A and B” may mean at least the following: “both A and B”, “at least both A and B”. As another example, “A or B” may mean at least the following: “at least A”, “at least B”, “both A and B”, “at least both A and B”. As another example, “A and/or B” may mean at least the following: “A and B”, “A or B”. When

an exclusive-or is intended, such will be specifically noted (e.g., “either A or B”, “at most one of A and B”).

[0035] This document describes various processing functions that are associated with structures such as blocks, elements, components, circuits, etc. In general, these structures may
 5 be implemented by a processor that is controlled by one or more computer programs.

[0036] FIG. 1 is a block diagram of a filter bank 100. The filter bank 100 includes a number of filters that are not individually shown. The filter bank 100 is configured with a number of weights 102 (also referred to as weighting coefficients), where each of the weights
 10 102 corresponds to a gain for a particular frequency band. The filter bank 100 receives an input signal 104, applies the weights 102 to the input signal 104, and generates an output signal 106. The weights 102 may vary over time.

[0037] The input signal 104 (and the output signal 106) may be a single channel signal, in which case the filter bank 100 includes a number of filters, where each filter corresponds to
 15 one of the weights 102 and a particular frequency band. The input signal 104 (and the output signal 106) may be a multichannel signal, in which case the filter bank 100 includes a number of arrays of filters, where each array of filters corresponds to one of the channels, and each filter in a given array corresponds to one of the weights 102 and a particular frequency band. The input signal 104 and the output signal 106 may be multichannel signals with differing
 20 number of channels, where each array of filters corresponds to one of the input channels and one of the output channels.

[0038] In general, the filtering techniques described herein may be applied to each individual filter in the filter array 100.

[0039] FIG. 2 is a graph 200 showing a set of filter bank frequency responses. In the graph
 25 200, the x-axis is the frequency and the y-axis is the gain. By way of example, a set of B band-pass impulse responses, $h_1(n)$, $h_2(n)$, ..., $h_B(n)$ may be pre-computed, with band-pass center frequencies f_{c1} , f_{c2} , ..., f_{cB} . Ideally, the filter $h_b(n)$ will have a frequency response, $H_b(f)$ that (approximately) satisfies:

30

Equation (1):

$$|H_b(f c_k)| = \begin{cases} 1 & \text{when } k = b \\ 0 & \text{otherwise} \end{cases}$$

[0040] The frequency response 202 of filter $H_2(f)$ is shown having a gain of 1 at $f = f c_2$ and a gain of 0 at all other values of $f = f c_b, b \neq 2$. The other filters $H_b(f)$ have a gain of 1 at other of the frequencies $f c_b$. The frequency $f s/2$ is the frequency of half of the sampling rate (the Nyquist rate).

[0041] A set of weighting coefficients, $w_1, w_2, \dots, w_B$ may then be used to form the combined filter response. The time-domain and frequency-domain versions of the equation are as follows:

Equation (2):

$$\text{Time domain: } h(n) = \sum_{b=1}^B h_b(n) w_b$$

Equation (3):

$$\text{Frequency domain: } H(f) = \sum_{b=1}^B H_b(f) w_b$$

[0042] FIG. 3 is a graph 300 showing an example combined filter response 302. The x-axis is the frequency (five frequency values shown) and the y-axis is the gain (five corresponding weights shown). The gain 304 at $f = f c_2$ is equal to w_2 , as an example. The other frequencies $f c_b$ have gains that correspond to their respective weights w_b . In a practical application, this combined filter response may be implemented in a variety of ways, as shown in FIGS. 4-5.

[0043] FIG. 4 is a block diagram of a filter bank 400. The filter bank 400 includes a number of convolution blocks 402, a number of multiplication blocks 404, and an addition block 406. The filter bank 400 receives an input signal 410 (shown as X and also referred to

as $x(n)$, which references the input signal for a given sample n , and generates an output signal 412 (shown as Y and also referred to as $y(n)$).

[0044] Each of the convolution blocks 402 generally corresponds to a given frequency, frequency band or frequency range associated with an impulse response 420 (shown as h_b). A given convolution block 402 convolves the input signal 410 (e.g., $x(n)$) with the impulse response 420 (e.g., $h_b(n)$) to generate a filtered signal 422 (e.g., $x_b(n)$).

[0045] Each of the multiplication blocks 404 is associated with a weight 430 (shown as w_b) and is associated with one of the convolution blocks 402 (so likewise associated with a frequency band). A given multiplication block 404 multiplies the filtered signal 422 (e.g., $x_b(n)$) with the weight 430 (e.g., w_b) to generate a weighted signal 432.

[0046] The addition block 406 adds the weighted signal 432 from each of the multiplication blocks 404 to generate the output signal 412.

[0047] The equations representing these operations are as follows:

Equation (4):

$$\text{Filtered sub-band signals: } x_b(n) = \{h \otimes x\}(n)$$

Equation (5):

$$= \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

Equation (6):

$$\text{Filtered output: } y(n) = \sum_{b=1}^B x_b(n)w_b$$

[0048] The $\{\dots \otimes \dots\}$ operator used in Equation (4) indicates the convolution of $h(n)$ with $x(n)$.

[0049] Equation (5) shows the details of the convolution operation, resulting in the creation of sub-band signals $x_1(n)$, $x_2(n)$, ..., $x_B(n)$. The output signal 412 (e.g., $y(n)$) is then formed as the sum of these sub-band signals multiplied by their respective weights, as shown in Equation (6).

- 5 [0050] According to this method, the weights may be time-varied, so w_b may also be replaced by $w_b(n)$ in Equation (6).

[0051] FIG. 5 is a block diagram of a filter bank 500. The filter bank 500 includes a number of multiplication blocks 502, an addition block 504, and a convolution block 506.

- 10 The filter bank 500 receives an input signal 510 (shown as X and also referred to as $x(n)$, which references the input signal for a given sample n), and generates an output signal 512 (shown as Y and also referred to as $y(n)$).

[0052] Each of the multiplication blocks 502 generally corresponds to a given frequency, frequency band or frequency range associated with an impulse response 520 (shown as h_b),

- 15 and is also associated with a weight 530 (shown as w_b). A given multiplication block 502 multiplies the impulse response 520 (e.g., $h_b(n)$) and the weighting coefficient 530 (e.g., $w_b(n)$) to generate one of the weighted responses 532.

[0053] The addition block 504 adds the weighted responses 532 from each of the multiplication blocks 502 to generate a combined filter response 540.

- 20 [0054] The convolution block 506 convolves the input signal 510 (e.g., $x(n)$) with the combined filter response 540 to generate the output signal 512 (e.g., $y(n)$).

[0055] In general, the filter bank 500 implements a dynamic filter that performs a periodic calculation of the filter response. For example, the input audio may be processed in (overlapped) blocks of audio, and at the time of block# k , the filter response (in the frequency

- 25 domain) may be computed as per Equation (3).

[0056] In one embodiment, we may implement the actual convolution operation in the frequency domain, using overlapped audio blocks, with smoothed cross-fades between audio block, which may allow the filter response to be changed on a block-by-block basis.

- [0057] In one common application, the filter bank responses are causal filters, in which the
30 filter output depends only on past and present inputs (so that for each $b \in \{1, 2, \dots, B\}$ the

impulse response $h_b(n) = 0, \forall n < 0$). If all filter bank responses are causal, then the combined response ($h(n)$, as per Equation (2)) will also be causal.

[0058] FIG. 6 is a graph of an impulse response 600. The impulse response 600 results from applying a filter such that for each $b \in \{1, 2, \dots, B\}$ the impulse response $h_b(n) = 0, \forall n < -L$, where L is a positive integer. Filters with this property may be implemented in practice by adding an additional delay of L samples in the filter implementation (so that the impulse responses are shifted by L samples in time to allow them to be implemented as causal filters).

[0059] FIG. 7 is a block diagram of a filter array 700. The filter array 700 includes a number of filters 702 (shown as $\text{Filter}_{ny, nx}$) and a number of addition blocks 704. The filter array 700 receives a number of input signals 710 (shown as X_1 to X_{nx}) and generates a number of output signals 712 (shown as Y_1 to Y_{ny}). In general, the filter array 700 illustrates the operation of a multichannel-input, multichannel-output system, where a filter joins every input to every output. The filter 700 may also be conceptualized as a multichannel filter bank where, for each frequency band, there is a $[ny \text{ by } nx]$ two-dimensional array that defines the mixing of inputs to form the outputs.

[0060] The filters 702 are arranged in banks, where each filter bank is associated with one of the input signals 710 (e.g., a channel). Each of the filters 702 is generally associated with a frequency band, a filter response, and a weight similar to the other filters discussed herein. For example, each of the filters 702 may be a filter such as the filter bank 400 (see FIG. 4), the filter bank 500 (see FIG. 5), etc.

[0061] In terms of frequency domain filter responses, we may describe the processing of the filter array 700 in terms of the following equation:

Equation (7):

$$Y_j(f) = \sum_{i=1}^{nx} \text{Filter}_{j,i}(f) \times X_i(f)$$

[0062] The embodiments described herein relate to methods used to implement one or more of the filters 702. It will be appreciated that embodiments are equally applicable to arrays of filters as shown in FIG. 7.

[0063] Referring back to FIG. 3, the desired frequency response of an example filter is shown, where the gain of the filter response 302, as a function of frequency, is defined according to a pre-defined set of control frequencies, $f_{c_1}, f_{c_2}, \dots$ and corresponding gain values, $w_1, w_2, \dots$

[0064] In the example of FIG. 3, the gain of the filter at frequency f_{c_2} is set by w_2 , as shown as the gain 204.

[0065] In an embodiment, the frequency response of FIG. 3 is achieved by the weighted summation of a number of pre-defined filter bank responses. Example responses are shown in FIG. 2, with the frequency response 202 of band 2 being shown.

[0066] We will refer to the response of each of these pre-defined filter bank responses as $H_b(f)$ (in the frequency domain) or alternately as $h_b(n)$ (in the time domain).

[0067] A desired filter response (such as 302 in FIG. 3) may be formed from a weighted sum of pre-defined filter bank responses. This may be expressed as a time-domain or frequency-domain summation:

Equation (8):

$$\text{Time domain: } h(n) = \sum_{b=1}^B h_b(n)w_b$$

Equation (9):

$$\text{Frequency domain: } H(f) = \sum_{b=1}^B H_b(f)w_b$$

[0068] We may choose to insist that each of the pre-defined filter bank responses should represent a causal filter (so that $h(n) = 0, \forall n < 0$). However, as a matter of convenience, we

instead insist that $h(n) = 0, \forall n < -L$ and we therefore will be required to add L samples of latency to the implementation of our filters, so that the final (realizable) filter output, $y'(n)$, will be computed as a delayed version of the ideal response, $y(n)$, according to:

5 Equation (10):

$$y(n) = \sum_{k=-L}^{\infty} h(k)x(n-k)$$

Equation (11):

$$y'(n) = y(n-L)$$

10

Equation (12):

$$= \sum_{k=-L}^{\infty} h(k)x(n-L-k)$$

[0069] Note that the calculation of output sample n of the ideal signal, $y(n)$, is computed using input samples up to $x(n+L)$ (including L future input samples). In contrast, the calculation of output sample n of the delayed signal, $y'(n)$, is computed only using input samples up to $x(n)$.

[0070] An example impulse response, with the property that $h(n) = 0, \forall n < -L$, is shown in FIG. 6.

20 [0071] In an embodiment, the frequency resolution of the filter bank (the spacing between adjacent filter bank center frequencies, for example $f_{c_{b+1}} - f_{c_b}$) is small, whilst the latency L is also small.

[0072] In one particular embodiment, a low-latency filter bank is used to provide low latency L and high resolution. In a further embodiment, an all-pass low-latency filter bank is used to provide low perceived latency L and high resolution.

25

[0073] Ideal Filter Bank Responses

[0074] A filter bank may be constructed by a variety of means, including a method whereby ideal frequency responses are created as shown in FIG. 2, and then converted to
 5 linear-phase ideal time-domain impulse responses.

[0075] FIG. 8 is a graph 800 showing an example set of time-domain filter bank impulse responses. In the graph 800, the x-axis is time (in samples) and the y-axis is frequency. Numerous filter responses are shown in the graph 800, with each filter response
 10 corresponding to a given frequency as indicated by the names $F_1, F_2, \dots, F_B$, where the number of filters in the filter bank, in this example is $B = 10$. The highest number (e.g. 10) corresponds to the highest frequency and the lowest number (e.g. 1) corresponds to the lowest frequency. The ideal impulse response for filter F_b is $h'_b(n)$ (for $b \in \{1, 2, \dots, B\}$). Two filter responses 802 and 804 are indicated for further discussion below; the filter response 802
 15 corresponds to the 3rd filter ($h'_3(n)$) and the filter response 804 corresponds to the lowest frequency filter response ($h'_1(n)$).

[0076] According to a common practice, the ideal filter bank responses exhibit a perfect summation property, as follows:

20 Equation (13):

$$\sum_{b=1}^B h'_b(n) = \delta(n)$$

Equation (14):

$$\text{where } \delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$$

25

[0077] Preferably, we may make use of filter banks where the lower frequency filters are narrower (and more closely spaced) than the higher frequency filters, as is illustrated in FIG.

2, where the separation between f_{c_1} and f_{c_2} is less than the separation between f_{c_4} and f_{c_5} , for example. As a consequence, the impulse responses of the higher frequency filters will be more compact (having most of their energy spread over a smaller number of samples) than the impulse responses of the lower frequency filters, as may be observed in FIG. 8.

- 5 [0078] Theoretically, the impulse responses of the ideal filter banks may be infinite in length, and in particular it may be seen, in FIG. 8, that the impulse response 802 of the filter F_3 does not satisfy the desired low-latency property: $h'_3(n) = 0, \forall n < -L$. Hence, the filters shown in FIG. 8 are not suitable for implementing a low-latency filter bank.

10 [0079] Techniques for Low-Latency Filter Responses

[0080] According to one method, a low-latency filter bank (with latency L) may be created by simply truncating the ideal impulse responses, as follows:

Equation (15):

$$15 \quad h_b(n) = \begin{cases} h'_b(n) & n \geq -L \\ 0 & n < -L \end{cases}$$

[0081] However, such truncation will impact the resulting responses at low frequencies, and may result in auditory artifacts if used in a real world system.

- [0082] FIG. 9 is a graph 900 of signals showing an improved method of modifying impulse responses. In the graph 900, the x-axis is time (in samples). The y-axis is magnitude; note that each signal is independent of each other on the y-axis (e.g., the y-axis shows the magnitude range of each signal independently, not any sort of relative magnitude or comparison between signals). The signal 802 corresponds to the ideal impulse response of the filter F_3 (e.g., $h'_3(n)$), as also shown in FIG. 8. The signal 904 corresponds to a fade function. The signal 906 corresponds to the signal 802 multiplied by the signal 904 to generate a low-latency filter response, as follows:
- 25

Equation (16):

$$h_b(n) = h'_b(n)q(n)$$

Equation (17):

$$\text{where: } q(n) = \begin{cases} 0 & n < -L \\ 1 & n \geq 0 \\ \cos^2\left(\frac{n\pi}{2(L+1)}\right) & -L \leq n < 0 \end{cases}$$

5

[0083] It will be appreciated by those skilled in the art that according to both techniques (Equation (15) and Equation (16)), the resulting impulse response $h_b(n)$ has the desirable property that $h_b(0) = h'_b(0)$, and this guarantees that the perfect-reconstruction criteria (Equation (13)) will be satisfied.

- 10 **[0084]** However, the frequency response of these filters (as produced by Equation (15) or Equation (16)) will not necessarily match the original (ideal) responses, $h'_b(n)$ as well as may be possible with improved techniques, as discussed below.

[0085] Improved Method for Low-Latency Filter Responses

- 15 **[0086]** FIG. 10 is a block diagram of a method 1000 of generating a filter response. The method 1000 may be used to generate the filter response for one or more of the filters in any of the filter banks discussed herein (e.g., the filter bank 400 of FIG. 4, the filter bank 500 of FIG. 5, the filter 702 in the filter array 700 of FIG. 7, etc.). The method 1000 includes a fade block 1002, a time reverse block 1004, and a summation block 1006. The blocks of the
- 20 method 1000 may be implemented in various ways, e.g. by circuit elements, by a processor executing one or more computer programs, etc.

- [0087]** The fade block 1002 receives an ideal impulse response 1010, applies a fade function, and generates a pre-ripple response 1012. The ideal impulse response 1010 may correspond to one or more of the ideal impulses responses discussed above (e.g., $h'_b(n)$ as discussed in FIG. 8 or FIG. 9, such as the signal 802). An example of the fade function is
- 25 shown in FIG. 11.

[0088] The time reverse block 1004 performs a time reverse operation on the pre-ripple response 1012 to generate a post-ripple response 1014. The time reverse operation generally corresponds to mirroring the pre-ripple response 1012 around the zero sample of the signal. An example of the time reverse operation is shown in FIG. 11.

- 5 [0089] The summation block 1006 adds the ideal impulse response 1010, subtracts the pre-ripple response 1012, and adds the post-ripple response 1014, to generate a filter response 1016.

[0090] FIG. 11 is a graph 1100 showing various signals related to FIG. 10. In the graph 1100, the x-axis is time (in samples) and the y-axis is magnitude; as with FIG. 9, note that the
10 magnitude of each signal is independent of each other signal. The signal 802 corresponds to the ideal impulse response of the filter F_3 (e.g., $h'_3(n)$), as also shown in FIGS. 8-9. See also the ideal impulse response 1010 in FIG. 10.

[0091] The signal 1102 corresponds to a fade function. See also the signal 904 in FIG. 9. The signal 1104 corresponds to the signal 802 multiplied by the signal 1102 to generate a pre-
15 ripple response. See also the pre-ripple response 1012 in FIG. 10.

[0092] The signal 1106 corresponds to a time reverse of the signal 1104, referred to as the post-ripple response. See also the post-ripple response 1014 in FIG. 10.

[0093] The signal 1108 corresponds to the filter bank response, and is generated by adding the signal 802 (the ideal impulse response 1010), subtracting the signal 1104 (the pre-ripple
20 response 1012), and adding the signal 1106 (the post-ripple response 1014). See also the filter response 1016 in FIG. 10.

[0094] The following equations show this process:

Equation (18):

25 pre-ripple: $r_b^{pre}(n) = h'_b(n)s(n)$

Equation (19):

$$\text{fade-function: } s(n) = \begin{cases} 1 & n < -L \\ 0 & n > L \\ \cos^2\left(\frac{(n+L+1)\pi}{4(L+1)}\right) & -L \leq n \leq L \end{cases}$$

Equation (20):

$$\text{post-ripple: } r_b^{\text{post}}(n) = r_b^{\text{pre}}(-n)$$

5

Equation (21):

$$\text{low-latency response: } h_b(n) = h'_b(n) - r_b^{\text{pre}}(n) + r_b^{\text{post}}(n)$$

[0095] Note that, according to Equation (20), $r_b^{\text{post}}(0) = r_b^{\text{pre}}(0)$ and hence, the low-latency impulse response $h_b(n)$ (as per Equation (21)) will have the desirable property that $h_b(0) = h'_b(0)$, and this guarantees that the perfect-reconstruction criteria (Equation (13)) will be satisfied.

[0096] Low-Latency All-Pass Filter Responses

15 [0097] Various types of audio filtering may be used to modify an audio signal without the group delay of the filtering being perceptible to a listener. As an example, a typical high-pass filter, such as that used to remove unwanted low-frequency noise from recorded sounds, will introduce phase distortion at low frequencies that will not be considered by a listener to be detrimental. Another common type of filtering that will be imperceptible to a listener is an all-pass filter.

[0098] FIG. 12 is a set of graphs showing a phase distortion 1202 and a group delay 1204. The phase distortion 1202 and the group delay 1204 result from using the transfer function as shown in Equation (22) to process an audio signal sampled at 48kHz.

25

Equation (22):

$$G_{ap} = \frac{0.994986 - 1.000000z^{-1}}{1.000000 - 0.994986z^{-1}}$$

[0099] The constant 0.994986 in Equation (22) is a typical value that corresponds to a transition frequency of approximately 37Hz (where the phase shift of a first-order all-pass filter is 90 degrees), for a filter operating at a sample rate of 48000 samples per second. Alternative values for this constant may be chosen, and for filters operating at 48000 samples per second, the constant may vary between 0.9490 and 0.9993, which values correspond to transitions frequencies of 400Hz and 5Hz, respectively.

[0100] Given that this phase distortion 1202 produces artefacts in the output audio signal that are acceptable to a listener, it is therefore acceptable to apply the same phase distortion to the ideal impulse responses, to produce a new set of all-pass filter responses, as shown in FIG. 13.

[0101] FIG. 13 is a graph 1300 showing various impulse responses. In the graph 1300, the x-axis is time (in samples) and the y-axis arranges the impulse responses at various frequencies F_n . The impulse response 1302 of filter F_3 appears to be similar to original (ideal) response 802 of filter F_3 in FIG. 8. In contrast, the impulse response 1304 of filter F_1 appears to be different from the original (ideal) response 804 of filter F_1 in FIG. 8. This difference is due to the effect of an all-pass filter, which affects the lower frequency filters, whilst having a negligible impact on the higher frequency filters. The all-pass filter is discussed in more detail below.

[0102] It can be seen that the impulse response 1304 of the filter F_1 has lower amplitude in the region 1306 (more than 100 samples prior to time zero) compared to the region 1308 (more than 100 samples after time zero). The shifting of the energy of the impulse response to a later time in the impulse response is a desirable attribute of the all-pass filter, as it produces a filter bank that more closely approximates the latency constraint (where $L = 100$ in the example of FIG. 13).

[0103] FIG. 14 is a block diagram of a method 1400 of generating a filter response. The method 1400 may be used to generate the filter response for one or more of the filters in any of the filter banks discussed herein (e.g., the filter bank 400 of FIG. 4, the filter bank 500 of

FIG. 5, the filter 702 in the filter array 700 of FIG. 7, etc.). The method 1400 includes an all-pass filter 1402, a fade block 1404, a time reverse block 1406, an all-pass filter 1408, an all-pass filter 1410, and a summation block 1412. The blocks of the method 1400 may be implemented in various ways, e.g. by circuit elements, by a processor executing one or more computer programs, etc.

[0104] The all-pass filter 1402 receives an ideal impulse response 1420, performs all-pass filtering, and generates a filter response 1422. The ideal impulse response 1420 may correspond to one or more of the ideal impulses responses discussed above (e.g., $h'_b(n)$ as discussed in FIG. 8 or FIG. 9, such as the signal 802). The impulse response of the all-pass filter is discussed in more detail in the equations below.

[0105] The fade block 1404 applies a fade function to the filter response 1422 to generate a pre-ripple response 1424. An example of the fade function is shown in FIG. 15.

[0106] The time reverse block 1406 performs a time reverse operation on the pre-ripple response 1424 to generate an intermediate response 1426. An example of the time reverse operation is shown in FIG. 15.

[0107] The all-pass filter 1408 performs all-pass filtering on the intermediate response 1426 to generate a filter response 1428.

[0108] The all-pass filter 1410 performs all-pass filtering on the filter response 1428 to generate a post-ripple response 1430.

[0109] The summation block 1412 adds the filter response 1422, adds the post-ripple response 1430, and subtracts the pre-ripple response 1424, to generate a filter response 1432.

[0110] In an embodiment, in order to correctly satisfy the latency constraint, we apply the process of Equation (21), with an adaptation to the calculation, to allow for the introduction of the all-pass filter. In the following equations, the notation $\{a \otimes b\}(n)$ is used to indicate the convolution of the impulse responses $a(n)$ and $b(n)$. The generation of the low-latency all-pass filter responses is carried out as follows:

Equation (23):

$$\text{all-pass: } h''_b(n) = \{g \otimes h'_b\}(n)$$

Equation (24):

$$\text{pre-ripple: } r_b^{pre}(n) = h''_b(n)s(n)$$

5 Equation (25):

$$\text{fade-function: } s(n) = \begin{cases} 1 & n < -L \\ 0 & n > L \\ \cos^2\left(\frac{(n+L+1)\pi}{4(L+1)}\right) & -L \leq n < 0 \end{cases}$$

Equation (26):

$$\text{post-ripple: } r_b^{post}(n) = r_b^{pre}(-n)$$

10

Equation (27):

$$\text{low-latency response: } h_b(n) = h''_b(n) - r_b^{pre}(n) + \{g \otimes g \otimes r_b^{post}\}(n)$$

[0111] In the above equations, $g(n)$ is the impulse response of the all-pass filter (see also

15 Equation (22)).

[0112] In an embodiment, the all-pass filter is a first-order filter with a single real pole at a frequency between 10Hz and 200Hz. For a filter bank that is operating on audio sampled at a rate of F_s sample per second, with a latency L , the frequency of the all-pass pole may be set to approximately:

20

Equation (28):

$$\text{all-pass pole freq: } f = \frac{F_s}{12L} \text{ Hz}$$

[0113] The calculation of the pole frequency as shown in Equation (28) is an example. It will be appreciated that other pole frequencies will also provide the benefit of an improved latency/accuracy trade-off in the design of a filter bank.

[0114] When the impulse responses of the set of B filters in the low-latency all-pass filter bank are summed together, the end result (the reconstruction impulse response) is approximately equal to the all-pass response:

Equation (29):

$$h(n) = \sum_{b=1}^B h_b(n)$$

10

Equation (30):

$$\approx g(n)$$

[0115] The procedure of FIG. 14 may be applied to each filter in the filter bank, so for example, for $b \in \{1, 2, \dots, B\}$, the ideal impulse response $h'_b(n)$ may be processed according to FIG. 14 to produce the low-latency all-pass filter $h_b(n)$.

[0116] FIG. 15 is a graph 1500 showing various signals related to FIG. 14. In the graph 1500, the x-axis is time (in samples) and the y-axis is magnitude; as with FIGS. 9 and 11, note that the magnitude of each signal is independent of each other signal. The signal 1304 corresponds to the all-pass filtered impulse response of the filter F_1 (e.g., $h'_1(n)$), as also shown in FIG. 13. See also the filter response 1422 in FIG. 14, and Equation (23).

[0117] The signal 1502 corresponds to a fade function. See also the signal 904 in FIG. 9 and the signal 1102 in FIG. 11, and Equation (25). The signal 1504 corresponds to the signal 1304 multiplied by the signal 1502 to generate a pre-ripple response. See also the pre-ripple response 1424 in FIG. 14, and Equation (24).

[0118] The signal 1506 corresponds to a time reverse of the signal 1504, referred to as the intermediate response. See also the intermediate response 1426 in FIG. 14. The signal 1508

corresponds to applying an all-pass filter twice to the signal 1506, referred to as the post-ripple response. See also the post-ripple response 1430 of FIG. 14, and Equation (26).

[0119] The signal 1510 corresponds to the filter bank response, and is generated by adding the signal 1304 (the filter response 1422), adding the signal 1508 (the post-ripple response
5 1430), and subtracting the signal 1504 (the pre-ripple response 1424). See also Equation (27).

[0120] The differences between the method 1000 of FIG. 10 and the method 1400 of FIG. 14 may be conceptualized as follows. First, note that the method 1000 operates on the filter response $h'_b(n)$, and the method 1400 operates on the all-pass filtered version $g \otimes h'_b(n)$. The process of the method 1000 relies on the fact that the pre-ripple and post-ripple
10 components are made by mirroring around $t=0$, and in conjunction we know that the ideal impulse response 1010 is symmetric about time zero.

[0121] Extending this knowledge to the method 1400, imagine we can make a new filter component: $\text{preA} = g^{-1} \otimes \text{pre_ripple}$. This means we take the original pre_ripple (e.g., the intermediate response 1426 or the signal 1506) and apply the inverse of the all-pass function
15 g .

[0122] We then make another new filter component: $\text{postA}(t) = \text{preA}(-t)$. This is simply the mirrored version of preA.

[0123] Now, we can make another new filter component: $\text{finalA} = h'_b(n) + \text{postA} - \text{preA}$. This step is very much like the process used in FIG. 10 and, most importantly, this step obeys
20 our mantra: the pre-ripple and post-ripple components are made by mirroring around $t=0$, and in conjunction we know that the original filter is symmetric about time zero.

[0124] We can then build the final filter: $\text{final_filter} = g \otimes \text{finalA} = g \otimes (h'_b(n) + \text{postA} - \text{preA}) = g \otimes h'_b(n) + g \otimes \text{postA} - g \otimes \text{preA} = \text{orig_filter} + g \otimes \text{postA} - \text{pre_ripple}$.

[0125] Now, we know that: $\text{postA} = \text{mirror}(\text{preA}) = \text{mirror}(g^{-1} \otimes \text{pre_ripple}) = g \otimes \text{mirror}(\text{pre_ripple}) = g \otimes \text{post_ripple}$. (This is true because the time-mirror of an all-pass filter is the same as the inverse of the all-pass filter.)
25

[0126] Hence, we get: $\text{final_filter} = \text{orig_filter} + g \otimes \text{postA} - \text{pre_ripple} = \text{orig_filter} + g \otimes g \otimes \text{post_ripple} - \text{pre_ripple}$.

[0127] FIG. 16 is a flow diagram of a method 1600 of audio processing. The method 1600 may be implemented by a processor executing instructions e.g. according to one or more computer programs.

[0128] At 1602, modified impulse responses are generated from ideal impulse responses.

5 The ideal impulse responses respectively correspond to a number of frequencies. Generating the modified impulse responses includes performing a fade operation and a time reverse operation on at least one of the ideal impulse responses. For example, the method 1000 (see FIG. 10) may perform a fade operation using the fade block 1002 and a time reverse operation using the time reverse block 1004 on the ideal impulse response 1010. As another
10 example, the method 1400 (see FIG. 14) may perform a fade operation using the fade block 1404 and a time reverse operation using the time reverse block 1406 on the ideal impulse response 1420.

[0129] At 1604, an input signal is filtered with the modified impulse responses (see 1602) to generate an output signal. For example, the filter bank 100 (see FIG. 1) may include a
15 number of filters with the modified impulse responses generated at 1602, and may generate the output signal 106 from the input signal 104.

[0130] The general steps of the method 1600 are further detailed in FIGS. 17-18.

[0131] FIG. 17 is a flow diagram of a method 1700 of audio processing. The method 1700 may be implemented by a processor executing instructions e.g. according to one or more
20 computer programs. The method 1700 is similar to the method 1600 (see FIG. 16) with more details specific to the operation of the method 1000 (see FIG. 10).

[0132] At 1702, modified impulse responses are generated from ideal impulse responses. This includes substeps 1704-1708.

[0133] At 1704, a pre-ripple response is generated based on a first ideal impulse response.

25 For example, the fade block 1002 (see FIG. 10) may perform a fade operation on the ideal impulse response 1010 to generate the pre-ripple response 1012. See also Equation (18).

[0134] At 1706, a post-ripple response is generated based on the pre-ripple response. For example, the time reverse block 1004 (see FIG. 10) may perform a time reverse operation on the pre-ripple response 1012 to generate the post-ripple response 1014. See also Equation

30 (20).

- [0135] At 1708, a first modified impulse response is generated by adding the first ideal impulse response, subtracting the pre-ripple response, and adding the post-ripple response. For example, the summation block 1006 (see FIG. 10) may add the ideal impulse response 1010, subtract the pre-ripple response 1012, and add the post-ripple response 1014 to
5 generate the filter response 1016. See also Equation (21).
- [0136] At 1710, an input signal is filtered with the modified impulse responses (see 1702-1708) to generate an output signal. For example, the filter bank 100 (see FIG. 1) may include a number of filters with the modified impulse responses generated at 1702, and may generate the output signal 106 from the input signal 104.
- 10 [0137] FIG. 18 is a flow diagram of a method 1800 of audio processing. The method 1800 may be implemented by a processor executing instructions e.g. according to one or more computer programs. The method 1800 is similar to the method 1600 (see FIG. 16) with more details specific to the operation of the method 1400 (see FIG. 14).
- [0138] At 1802, modified impulse responses are generated from ideal impulse responses.
15 This includes substeps 1804-1814.
- [0139] At 1804, a first filter response is generated based on a first ideal impulse response. For example, the all-pass filter 1402 (see FIG. 14) may perform an all-pass filter operation on the ideal impulse response 1420 to generate the filter response 1422. See also Equation (23).
- [0140] At 1806, a pre-ripple response is generated based on the first filter response. For
20 example, the fade block 1404 (see FIG. 14) may perform a fade operation on the filter response 1422 to generate the pre-ripple response 1424. See also Equation (24).
- [0141] At 1808, an intermediate response is generated based on the pre-ripple response. For example, the time reverse block 1406 (see FIG. 14) may perform a time reverse operation on the pre-ripple response 1424 to generate the intermediate response 1426. See also Equation
25 (26).
- [0142] At 1810, a second filter response is generated based on the intermediate response. For example, the all-pass filter 1408 (see FIG. 14) may perform an all-pass filter operation on the intermediate response 1426 to generate the filter response 1428. See also Equation (27).

[0143] At 1812, a post-ripple response is generated based on the second filter response. For example, the all-pass filter 1410 (see FIG. 14) may perform an all-pass filter operation on the filter response 1428 to generate the post-ripple response 1430. See also Equation (27).

5 [0144] At 1814, a first modified impulse response is generated by adding the first filter response, adding the post-ripple response, and subtracting the pre-ripple response. For example, the summation block 1412 (see FIG. 14) may add the filter response 1422, add the post-ripple response 1430, and subtract the pre-ripple response 1424 to generate the filter response 1432. See also Equation (27).

10 [0145] At 1816, an input signal is filtered with the modified impulse responses (see 1802-1814) to generate an output signal. For example, the filter bank 100 (see FIG. 1) may include a number of filters with the modified impulse responses generated at 1802, and may generate the output signal 106 from the input signal 104.

15 [0146] In an embodiment, an input audio signal may be processed to produce an output audio signal where the gain of the processing at a number of predetermined frequencies is approximately equal to a corresponding set of gain coefficients. The processing may be implemented according to a filter response that is the weighted sum of a corresponding number of low-latency all-pass filter responses with the weightings being determined by the corresponding gain coefficients. The low-latency all-pass filter responses have been determined according to the method of FIG. 18.

20 [0147] In an alternative embodiment, an input audio signal is processed to produce an output audio signal, where the said output audio signal is formed by filtering said input audio signal by a total impulse response that is a weighted sum of a set of filter bank impulse responses. The sum of said filter bank impulse responses is an all-pass filter. The all-pass filter may have approximately constant phase response above a frequency of approximately
25 500Hz, and a group-delay that rises at lower frequencies.

[0148] In an alternative embodiment, an input audio signal is processed to produce an output audio signal, where the said output audio signal is formed by filtering said input audio signal by a total impulse response that is a weighted sum of a set of filter bank impulse response. A reconstruction impulse response may be formed by the sum of the filter bank
30 impulse responses such that the group delay of the reconstruction impulse response is approximately constant over a frequency range above 500Hz and increases below 500Hz. For example, the group delay may be perceptually constant above 500Hz (varying by no more

than 0.1ms), and at lower frequencies, e.g., less than 500Hz, may vary by less than 0.5/f seconds. The bandwidth of one or more impulse responses from said set of filter bank impulse responses may be narrower according to the increased group delay of the reconstruction impulse response.

5

[0149] Implementation Details

[0150] An embodiment may be implemented in hardware, executable modules stored on a computer readable medium, or a combination of both (e.g., programmable logic arrays).

Unless otherwise specified, the steps executed by embodiments need not inherently be related to any particular computer or other apparatus, although they may be in certain embodiments. In particular, various general-purpose machines may be used with programs written in accordance with the teachings herein, or it may be more convenient to construct more specialized apparatus (e.g., integrated circuits) to perform the required method steps. Thus, embodiments may be implemented in one or more computer programs executing on one or more programmable computer systems each comprising at least one processor, at least one data storage system (including volatile and non-volatile memory and/or storage elements), at least one input device or port, and at least one output device or port. Program code is applied to input data to perform the functions described herein and generate output information. The output information is applied to one or more output devices, in known fashion.

[0151] Each such computer program is preferably stored on or downloaded to a storage media or device (e.g., solid state memory or media, or magnetic or optical media) readable by a general or special purpose programmable computer, for configuring and operating the computer when the storage media or device is read by the computer system to perform the procedures described herein. The inventive system may also be considered to be implemented as a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer system to operate in a specific and predefined manner to perform the functions described herein. (Software per se and intangible or transitory signals are excluded to the extent that they are unpatentable subject matter.)

[0152] The above description illustrates various embodiments of the present disclosure along with examples of how aspects of the present disclosure may be implemented. The above examples and embodiments should not be deemed to be the only embodiments, and are presented to illustrate the flexibility and advantages of the present disclosure as defined by

the following claims. Based on the above disclosure and the following claims, other arrangements, embodiments, implementations and equivalents will be evident to those skilled in the art and may be employed without departing from the spirit and scope of the disclosure as defined by the claims.