MARCH 24 2022

Optimized design of windowed-sinc anti-aliasing filters for phase-preserving decimation of hydrophone data

Yanwu Zhang; Paul R. McGill; John P. Ryan

J. Acoust. Soc. Am. 151, 2077–2084 (2022)
https://doi.org/10.1121/10.0009823

JASA ARTICLE

Optimized design of windowed-sinc anti-aliasing filters for phase-preserving decimation of hydrophone data

Yanwu Zhang,[a] Paul R. McGill,[b] and John P. Ryan[c]
Monterey Bay Aquarium Research Institute, Moss Landing, California 95039, USA

ABSTRACT:
Passive acoustic monitoring generates large data sets for which decimation is beneficial to analysis and portability for data sharing. Among the goals for effective decimation are avoidance of aliasing in the passband, accurate and complete control of the attenuation profile, phase preservation, and high efficiency in processing. We present an approach to decimator design that addresses each of these goals, and we demonstrate its application to ocean audio recordings. Anti-aliasing is achieved by windowed-sinc filters that also preserve phase. Control of the passband attenuation profile is based on the specification of the maximum allowed attenuation at a certain percentage of the final output Nyquist frequency. The window type is selected to meet the stopband attenuation requirement. Efficiency is achieved through optimization of the anti-aliasing filters applied in each decimation stage, and through parallelization of processing. The best combination of the sinc function's cutoff frequency and the mainlobe bandwidth of the window function generates the shortest qualifying filter, optimizing the trade-off between filter performance and computational load. Parallelization is enabled by applying the overlap-add method to contiguous segments of audio data, consistent with the commonly used storage of contiguous audio data in files of limited duration. Beyond addressing common goals for effective decimation of audio data, the approach presented is deployable in open-source environments.
https://doi.org/10.1121/10.0009823

(Received 18 November 2021; revised 18 February 2022; accepted 27 February 2022; published online 24 March 2022)
[Editor: Karim G. Sabra] Pages: 2077–2084

I. INTRODUCTION

A common need in the analysis of ocean passive acoustic monitoring data is decimation. A reduced sample rate that meets requirements for analysis of many sound sources can greatly reduce data volume, expedite data analysis, and facilitate data sharing. Desirable attributes in a decimation procedure include high-fidelity signal retention approaching the Nyquist frequency (i.e., a narrow transition band), strong attenuation across the stopband to prevent spectral leakage (Oppenheim and Schafer, 1989), and preservation of signal phase. More effective decimation procedures are computationally expensive, thus introducing the need to optimize filter design in order to achieve target performance metrics with maximum computational efficiency. Here, we describe and demonstrate an optimization approach for this purpose.

The data for this demonstration are from a seabed-mounted hydrophone connected to a cabled ocean observatory installed at 890 m depth off central California (Ryan *et al.*, 2016). The cabled observatory supplies continuous power to the hydrophone and transmits the hydrophone data stream to shore in real time. The instrument sampling rate is 256 kHz. An example one-day spectrogram spanning from 5 Hz to 100 kHz (Fig. 1) represents a variety of sound sources including dolphins, three baleen whale species, and transiting vessels. Many research topics concerning biological and anthropogenic sound in the ocean can be studied using audio data with a much lower sample rate of 2 kHz (Helble *et al.*, 2012; Širović *et al.*, 2015; Oestreich *et al.*, 2020; Simonis *et al.*, 2020; Ryan *et al.*, 2021). In order to reduce data storage and expedite data analysis, we constructed a decimator that decreases the data sample rate from 256 to 2 kHz, while retaining spectral content below 1 kHz with high fidelity. The method presented in this paper is applicable to any decimation factor, and it can be implemented with parallel architecture for rapid processing of large data archives.

The decimation process is composed of an anti-aliasing low-pass filter followed by down-sampling. The anti-aliasing filter sufficiently attenuates spectral content above the Nyquist frequency (Oppenheim and Schafer, 1989) to prevent aliasing in the down-sampling step. To preserve signal phase, we use symmetric finite impulse response (FIR) filters to conduct low-pass filtering that provides linear-phase responses in the frequency domain. In the time domain, the same amount of delay is introduced for all frequencies, thus retaining the temporal sequence of different frequency components. In contrast, the nonlinear-phase response of an infinite impulse response (IIR) filter

[a]Electronic mail: yzhang@mbari.org, ORCID: 0000-0002-8773-9275.
[b]ORCID: 0000-0001-7707-2529.
[c]ORCID: 0000-0001-7954-5369.

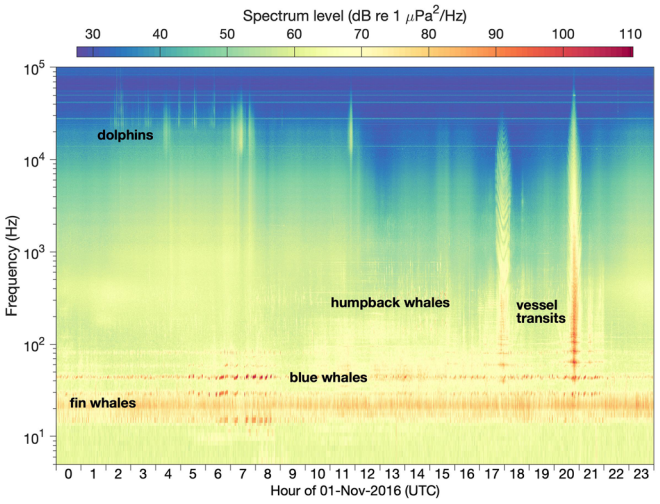

FIG. 1. (Color online) Spectrogram of one day of data from the seabed-mounted hydrophone.

introduces different delays for different frequencies, which alters the temporal sequence of different frequency components (Oppenheim and Schafer, 1989).

Among FIR filters, an often-used type is windowed-sinc filters (Crochiere and Rabiner, 1981). They are formulated by using closed-form expressions of the sinc and the window functions as shown below. Compared with the structure of cascaded filters (Saramaki, 1984), windowed-sinc filters are straightforward to design and easy to implement. A windowed-sinc low-pass filter $h(n)$ is the product of the normalized sinc function and a certain window (Lyons, 2011; Smith, 1997), expressed as follows:

$$h(n) = s(n) \cdot w(n), \tag{1}$$

where $s(n) = sinc(2f_{c\_sinc}n) = \sin(2\pi f_{c\_sinc}n)/(2\pi f_{c\_sinc}n)$ is the normalized sinc function (Oppenheim *et al.*, 1983) and f_{c_sinc} is the cutoff frequency; $w(n)$ is the window function with a finite length. Popular windows include the Hamming window and the Blackman window (Oppenheim and Schafer, 1989). In the frequency domain, the frequency response of the windowed-sinc filter is the convolution of the Fourier transform of the sinc function and that of the window function:

$$H(f) = S(f) * W(f), \tag{2}$$

where $S(f) = \mathcal{F}[s(n)]$ and $W(f) = \mathcal{F}[w(n)]$; $\mathcal{F}[\cdot]$ stands for Fourier transform; "$*$" stands for convolution. $S(f)$ has a boxcar shape with cutoff frequency f_{c_sinc}. $W(f)$ has a mainlobe centered at zero frequency, and side lobes falling with increasing frequency. The mainlobe bandwidth (MLW) and the sidelobe heights depend on the type and length of the window.

The convolution result, $H(f)$, is low-pass. When the shapes of $S(f)$ and $W(f)$ change, the properties of the transition band of $H(f)$ change accordingly. The cutoff frequency of the passband of $H(f)$ is determined by f_{c_sinc} of $S(f)$; the steepness of the transition band of $H(f)$ is determined by the MLW of $W(f)$. For example, in Fig. 2, three pairs of different Blackman windows and sinc functions produce three different Blackman-sinc low-pass filters (see Sec. II). A shorter $w(n)$ corresponds to a larger MLW of $W(f)$, and consequently a wider transition band (i.e., a wider interval between the passband cutoff frequency and the stopband start frequency) of $H(f)$. To minimize the computational load of low-pass filtering, we desire the shortest filter whose transition band is the widest allowed yet still meets the passband and stopband requirements.

The common practice in designing a windowed-sinc low-pass filter (Smith, 1997) is: (i) set f_{c_sinc} of $S(f)$ to the desired passband cutoff frequency; (ii) calculate MLW of $W(f)$ based on the desired transition bandwidth. Ad hoc adjustments of f_{c_sinc} and MLW settings are made to ensure $H(f)$ meets the passband and stopband requirements. In this paper, we present a systematic way to find the shortest filter that meets the requirements.

II. METHODS

A. Two-stage decimator design

When the decimation factor is large, multi-stage decimation is more efficient than that of one stage (Crochiere and Rabiner, 1975). Each stage achieves a fraction of the desired factor. The gradual reduction of sample rate imposes less-demanding filtering requirements resulting in a shorter anti-aliasing low-pass filter in each stage. Computational efficiencies with different numbers of stages are analyzed in Crochiere and Rabiner (1975). The analysis showed that generally the largest gains in efficiency are obtained in going from a one-stage to a two-stage design, and the benefit of using more stages is very small.

Our filter optimization technique is applied to each stage of the decimator, hence not restricted by the number of stages chosen. In this paper, we consider decimating data from 256 kHz sample rate to 2 kHz using a two-stage decimator. We partition the decimation factor $256/2 = 128$ into two stages of decimation using a partition formula (Crochiere and Rabiner, 1975): the 1st stage from 256 to 8 kHz (at a decimation factor of 32), and the 2nd stage from 8 to 2 kHz (at a decimation factor of 4), as shown in Fig. 3.

B. Designing the optimized windowed-sinc filter in each decimation stage

The Blackman window produces windowed-sinc filters that provide strong stopband attenuation and low passband ripple (Lyons, 2011). In this paper, we use Blackman windows to present the method, although the method is equally valid for other window functions. A Blackman window of length L is

https://doi.org/10.1121/10.0009823

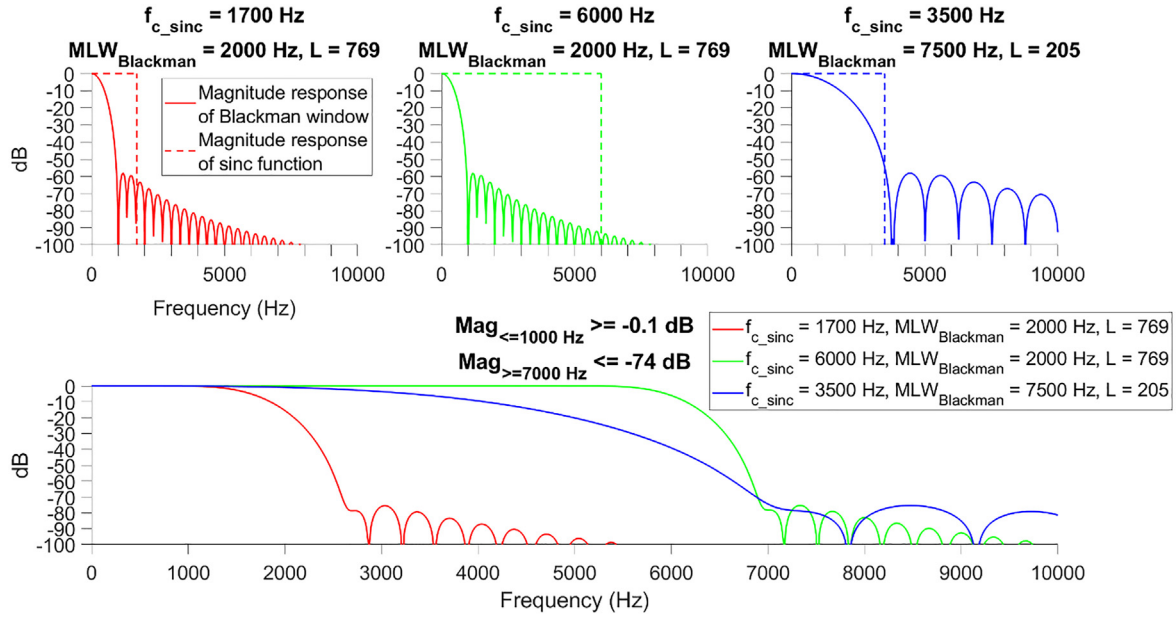

FIG. 2. (Color online) Design of the 1st-stage filter. Three pairs of sinc functions and Blackman windows (magnitude responses shown in different colors in upper panels) produce three filters of different lengths (in corresponding colors in lower panel) that all meet the passband (zero to 1 kHz) and stopband (from 7 to 128 kHz) requirements. Sample rate is 256 kHz.

$$w(n) = \begin{cases} 0.42 - 0.5\cos\left(\dfrac{2\pi n}{L-1}\right) \\ \quad +0.08\cos\left(\dfrac{4\pi n}{L-1}\right) & \text{for } 0 \le n \le L-1, \\ 0 & \text{otherwise.} \end{cases}$$

The mainlobe bandwidth $MLW_{Blackman}$ of $W(f)$ is $6/L$ (typically $L \gg 6$), and the peak sidelobe amplitude is -57 dB (Oppenheim and Schafer, 1989). L is also the length of the windowed-sinc filter [Eq. (1)]. A Blackman-sinc low-pass filter provides a stopband attenuation of 74 dB.

We intend to find the best combination of the sinc function's cutoff frequency and the mainlobe bandwidth of the window function to generate the shortest filter that meets the passband and stopband requirements, and we call this process "optimization." We design the 1st-stage Blackman-sinc low-pass filter for raw input data sampled at 256 kHz, with the following requirements: in the passband from zero to 1 kHz, attenuation is less than 0.1 dB; in the stopband from 7 to 128 kHz, attenuation is greater than 74 dB. We want the filter to be as short as possible to reduce the computational load in decimation. When the data are down-sampled to 8 kHz, the passband spectrum from 4 to 7 kHz will be aliased and folded back to the band from 1 to 4 kHz. Nonetheless, this aliased spectral section (1–4 kHz) will be filtered out in the 2nd-stage decimation. We allow a wide transition band (from 1 to 7 kHz) in order to shorten the length of the 1st-stage filter.

When $S(f)$ is convolved with $W(f)$ [Eq. (2)], a range of different combinations of f_{c_sinc} and $MLW_{Blackman}$ can produce filters all meeting the passband and stopband requirements, three of which are shown in Fig. 2. For filter No. 1 (upper left panel), the combination of $f_{c\_sinc} = 1700$ Hz and $MLW_{Blackman} = 2000$ Hz (corresponding to $L = 769$) produces a filter whose magnitude response starts to fall off at 1000 Hz. For filter No. 2 (upper middle panel), $MLW_{Blackman}$ remains the same as that for filter No. 1 (hence the same L), but $f_{c\_sinc} = 6000$ Hz is much higher than that for filter No. 1. Consequently, the filter's magnitude response starts to fall off at a much higher frequency of 5500 Hz. Magnitude responses of filters No. 1 and 2 fall off with the same steepness (because of the same L), but the falloff starts at a lower and a higher frequency respectively. We note that the falloff does not need to be steep but can be a gradual fall from 1 to 7 kHz, i.e., $MLW_{Blackman}$ is allowed to be much larger (corresponding to a much smaller L). We adopt this strategy to design Filter No. 3 (upper right panel) by using a much larger $MLW_{Blackman} = 7500$ Hz (corresponding to a much smaller $L = 205$). However, the setting $f_{c\_sinc} = 3500$ Hz

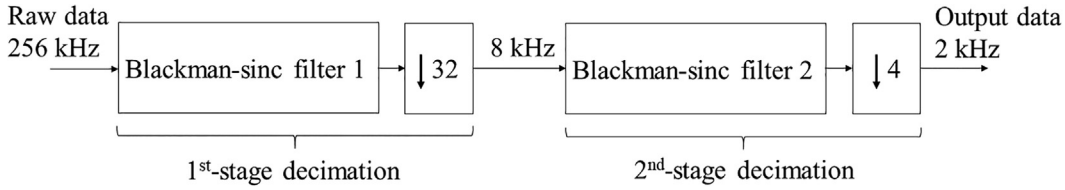

FIG. 3. Diagram of the two-stage decimator. Each stage comprises a Blackman-sinc anti-aliasing low-pass filter followed by a down-sampler.

has to be just right to produce a filter that still satisfies the passband and stopband requirements.

We search through a range of $[MLW_{Blackman}, f_{c\_sinc}]$ combinations to find the shortest qualifying filter, as shown in Fig. 4. At a small $MLW_{Blackman}$ (corresponding to a large L), a wide range of f_{c_sinc} settings is available to be combined with this $MLW_{Blackman}$ to produce qualifying filters, such as filters No. 1 and 2. When $MLW_{Blackman}$ increases (corresponding to L decreasing), the range of usable f_{c_sinc} settings shrinks, giving a wedge shape of the usable $[MLW_{Blackman}, f_{c\_sinc}]$ combinations. At the rightmost tip of the wedge where $MLW_{Blackman}$ is the largest (corresponding to the smallest L), there is only one usable f_{c_sinc} setting (given the search grid resolution). This $[MLW_{Blackman}, f_{c\_sinc}]$ combination produces the shortest qualifying filter ($L = 205$, i.e., filter No. 3 in Fig. 2).

When the passband cutoff frequency gets closer to the stopband start frequency, there is less room for expanding $MLW_{Blackman}$ (corresponding to reducing L). Now we design the 2nd-stage Blackman-sinc low-pass filter for input data sampled at 8 kHz, with the following requirements: in the passband from zero to 900 Hz, attenuation is less than 1 dB; in the stopband from 1000 to 4000 Hz, attenuation is greater than 74 dB. The interval between the passband cutoff frequency and the stopband start frequency is only 100 Hz. Three filters are compared in Fig. 5. Filters No. 1 and 2 (upper left and middle panels) are of the same length but different passband cutoff frequencies. Filter No. 3 (upper right panel) takes a more gradual falloff in the transition band by using a Blackman window of a larger $MLW_{Blackman}$ (corresponding to a smaller L).

The wedge-shaped $[MLW_{Blackman}, f_{c\_sinc}]$ combinations that produce qualifying filters are shown in Fig. 6. The $[MLW_{Blackman}, f_{c\_Sinc}]$ combination at the rightmost tip of the wedge produces the shortest filter ($L = 321$, i.e., filter No. 3 in Fig. 5).

C. Comparison with filters designed by the Parks-McClellan algorithm

We compare the 321-tap Blackman-sinc filter with two filters (218-tap and 327-tap, respectively) designed by the Parks-McClellan algorithm (Oppenheim and Schafer, 1989), as shown in Fig. 7. The passband ripple of the three filters is compared in the right panel. Both Parks-McClellan filters meet the stopband requirement and have a steeper transition band than that of the Blackman-sinc filter. However, the Blackman-sinc filter is advantageous in providing a very flat magnitude response (i.e., high fidelity) in the passband. The 218-tap Parks-McClellan filter's passband ripple (± 0.2 dB) is much higher than that of the 321-tap Blackman-sinc filter ($\pm 10^{-4}$ dB); the 327-tap Parks-McClellan filter's passband ripple ($\pm 10^{-3}$ dB) is comparable with but still higher than the Blackman-sinc filter's passband ripple. In applications where high fidelity in the passband is required, Blackman-sinc filters cause very little amplitude distortion.

III. APPLICATION TO HYDROPHONE DATA DECIMATION

A. Decimation using the optimized Blackman-sinc filters

We use the 2-stage decimator designed in Sec. II[1] to decimate the hydrophone data stream from the raw sample rate of 256 kHz down to 2 kHz. The 1st-stage decimation uses the 205-tap Blackman-sinc filter (colored blue in Fig. 2) with the passband from zero to 1 kHz and the stopband from 7 to 128 kHz. The 2nd-stage decimation uses the 321-tap Blackman-sinc filter (colored blue in Fig. 5) with the passband from zero to 900 Hz and the stopband from 1 to 4 kHz.

We apply the decimator to a 10-min section of the hydrophone data shown in Fig. 1, during the second vessel transit beginning 20:41:26. The power spectral densities

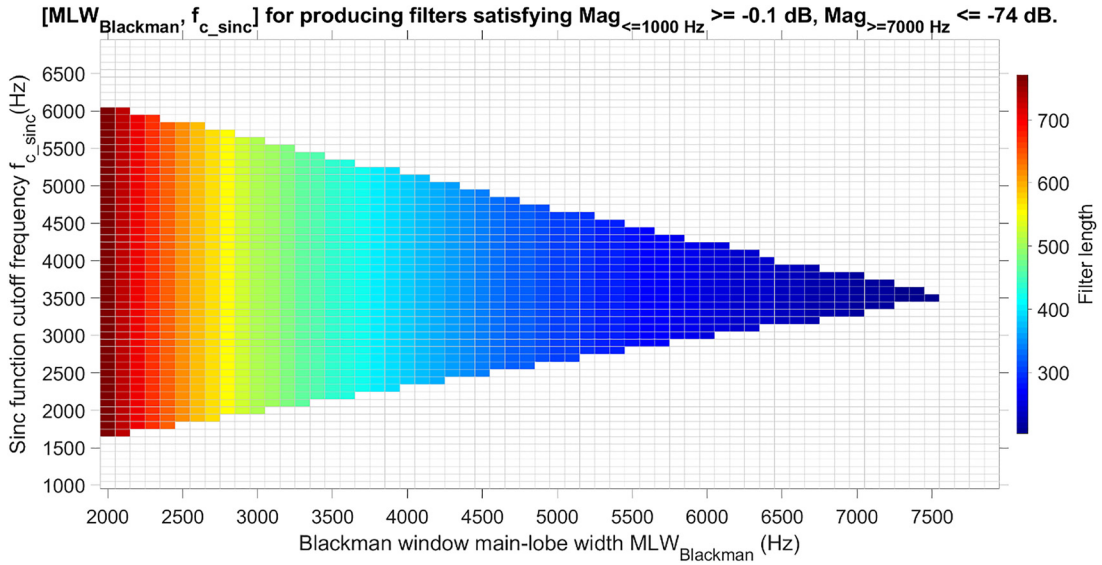

FIG. 4. (Color online) For designing the 1st-stage filter, the usable $[MLW_{Blackman}, f_{c\_sinc}]$ combinations that produce filters meeting the passband and stopband requirements. The rightmost tip of the wedge corresponds to the shortest filter.

https://doi.org/10.1121/10.0009823

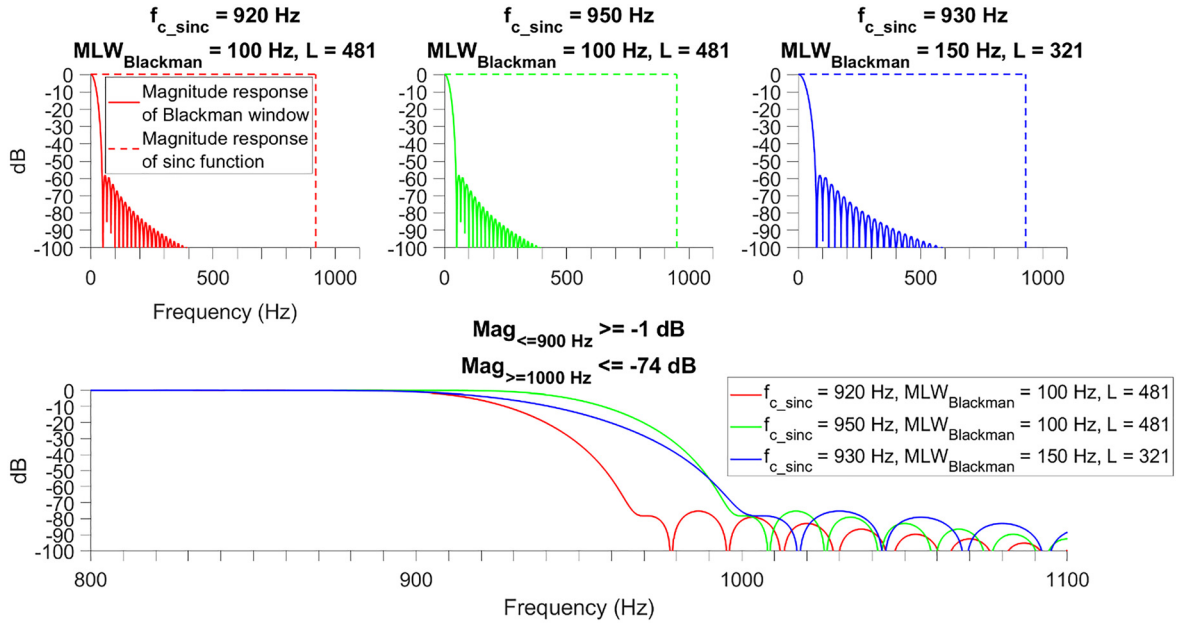

FIG. 5. (Color online) Design of the 2nd-stage filter. Three pairs of sinc functions and Blackman windows (in different colors in upper panels) produce three filters of different lengths (in corresponding colors in lower panel) that all meet the passband (zero to 900 Hz) and stopband (from 1000 to 4000 Hz) requirements. Sample rate is 8 kHz.

(PSDs) of the raw data (256 kHz sample rate), the 1st-stage decimated data (8 kHz sample rate), and the 2nd-stage decimated data (i.e., the final output data; 2 kHz sample rate) are compared in Fig. 8. PSDs are computed by the Welch's method (Stoica and Moses, 1997) (segment length = 15 s using the Blackman window; 50% overlap between segments).

In the frequency range from zero to 900 Hz, the PSD of the final output data overlaps with that of the raw data, thus meeting the requirement of high-fidelity retention of the raw hydrophone data's spectral content up to 900 Hz. The two-stage decimation significantly reduces the amount of data for expedited data analysis, while satisfactorily retaining the desired signal band and excluding the unwanted band.

B. Computational efficiency

When an N-point data sequence of 256 kHz sample rate passes the 1st-stage filter of length L_1, the computational load of convolution is $N \cdot L_1$ multiply-adds. The output data sequence is then down-sampled by a factor of 32, producing an $N/32$-point data sequence of 8 kHz sample rate. When this sequence passes the 2nd-stage filter of length L_2, the computational load is $N/32 \cdot L_2$ multiply-adds. Hence the

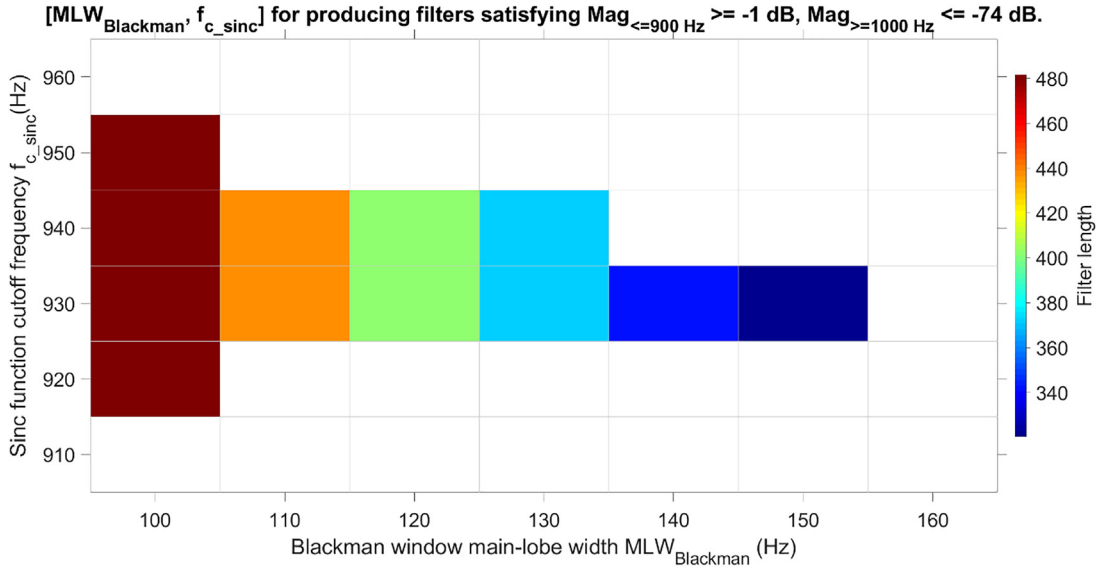

FIG. 6. (Color online) For designing the 2nd-stage filter, the usable $[MLW_{Blackman}, f_{c\_sinc}]$ combinations that produce filters meeting the passband and stopband requirements.

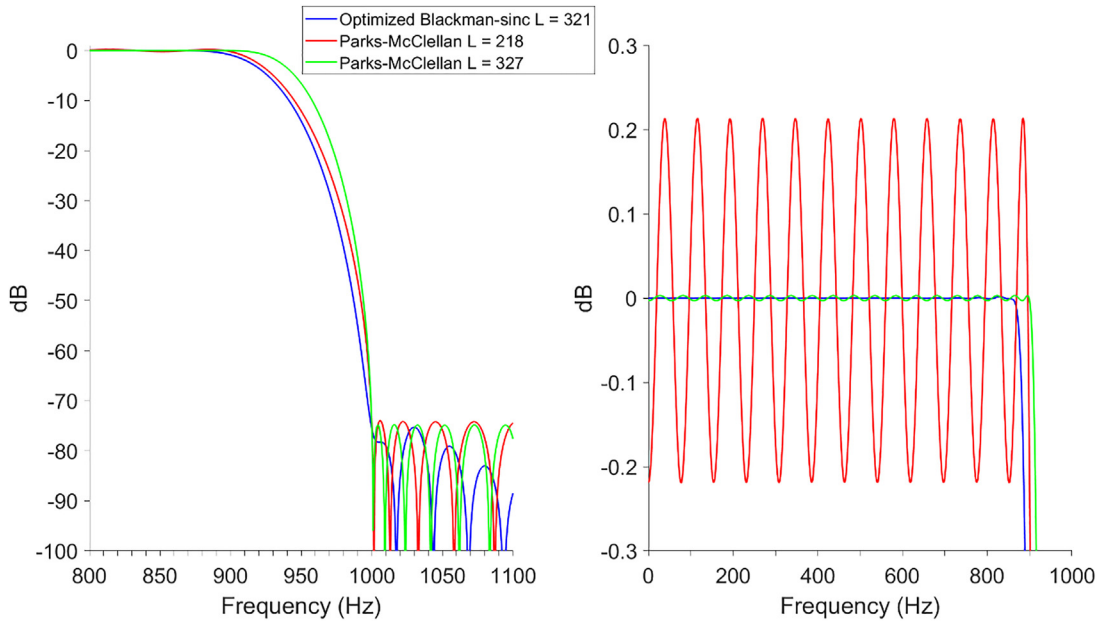

FIG. 7. (Color online) Comparison between the magnitude responses of the Blackman-sinc filter and two Parks-McClellan filters (left panel: from passband to stopband between 800 and 1100 Hz; right panel: close-up view of passband ripple between zero and 900 Hz).

total computational load is $(N \cdot L_1) + (\mathrm{N}/32 \cdot L_2) = N[L_1 + (L_2/32)]$ multiply-adds. The 1st-stage filter length has a much heavier weight than the 2nd-stage filter length in the total computational load because the 1st-stage filter is applied to data sampled at a much higher sample rate. Plugging in $L_1 = 205$ and $L_2 = 321$, the total computational load is $215N$ multiply-adds.

In each stage of decimation, in the down-sampling step only every Mth data point of the anti-aliasing filter output is retained (where M is the decimation factor) and the other output data points are thrown away. In place of direct implementation where filtering is followed by down-sampling, polyphase implementation can reduce the computational load by a factor of M by using a bank of M polyphase sub-filters (Oppenheim and Schafer, 1989). Each sub-filter has a length of L/M (where L is the length of the anti-aliasing filter), and only processes every Mth data point of the input data sequence. The computation savings come from down-sampling first and then filtering.

For long filters, "fast Fourier transform (FFT) convolution" can run faster than the standard convolution (Smith, 1997; Oppenheim and Schafer, 1989). "FFT convolution" uses the principle that the Fourier transform of a time-domain convolution equals a multiplication in the

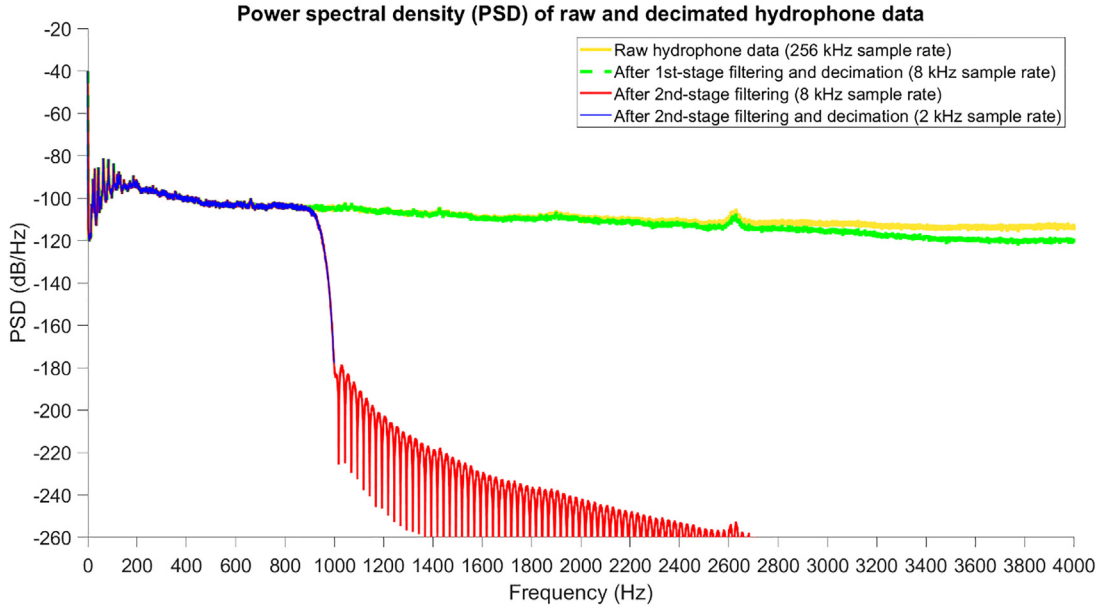

FIG. 8. (Color online) Performance of the two-stage decimation.

https://doi.org/10.1121/10.0009823

frequency domain, and takes advantage of the computational efficiency of the FFT and inverse fast Fourier transform (IFFT) algorithms. The procedure is: the input signal is transformed into the frequency domain using FFT, multiplied by the frequency response of the filter which is also computed by FFT, and then transformed back into the time domain using IFFT (Smith, 1997).

A long data sequence can be divided into shorter segments. Each segment passes the 2-stage decimator, and then the processed segments can be seamlessly recombined using the "overlap-add" method (i.e., overlapping and adding the tails and heads of successive segments) (Smith, 1997; Oppenheim and Schafer, 1989). Using this method, the decimation job can be allocated to multiple processors for parallel processing. Note that the "overlap-add" method is applicable to FIR filters but not to IIR filters; this is yet another consideration when we choose to use FIR anti-aliasing filters.

C. Comparison with some signal processing software packages

Some signal processing software packages offer "canned" decimation functions. In this section, we compare our approach with "*decimate*" functions in MATLAB and Python in terms of anti-aliasing performance and suitability for parallel processing.

1. Anti-aliasing performance

MATLAB and Python both have a decimation function giving the option of using an FIR anti-aliasing filter. However, the user can only specify the filter length but not the desired passband and stopband attributes. To facilitate the assessment of anti-aliasing performance, we add two strong interference tones of 1010 and 1100 Hz to the 10-min hydrophone data to form the raw input. If the added interference tones were weak, their aliased tones in the passband would be buried in the hydrophone signal and hard to assess. We use the 205-tap Blackman-sinc filter to implement the 1st-stage decimation of factor 32. For the 2nd-stage decimation of factor 4, we compare the performance of the 321-tap Blackman-sinc filter, the MATLAB Signal Processing Toolbox function *decimate(X, 4, 321, 'fir')*, and the Python SciPy package function *scipy.signal.decimate(X*, 4, $n = 320$, *ftype = "fir")*, where X is the 1st-stage decimation output. The lengths of the anti-aliasing FIR filters in MATLAB and Python *decimate* functions are set to the same as the Blackman-sinc filter (the length setting in the Python *decimate* function is one less than the filter length).

The anti-aliasing performance is compared in Fig. 9. The 1010-Hz interference (just above the Nyquist frequency 1 kHz) makes a sensitive test because any unfiltered remnant will stand out as an aliased mirror tone (about the Nyquist frequency) at 990 Hz, where the hydrophone data PSD is suppressed in the transition band of the 2nd-stage low-pass filter. The introduction of the 1100-Hz tone is aimed at evaluating aliasing of higher-frequency interference. As shown

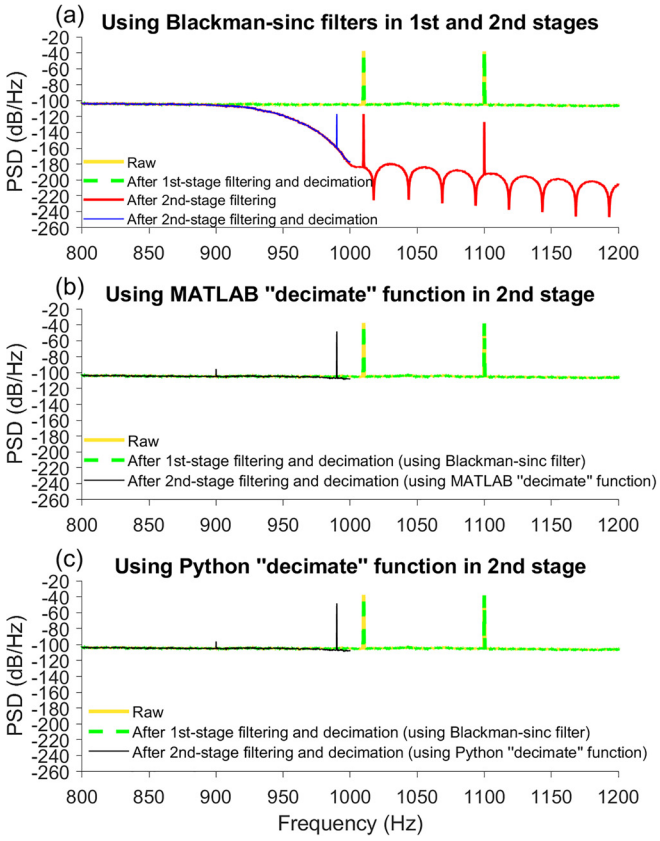

FIG. 9. (Color online) Comparison of anti-aliasing performance using the Blackman-sinc anti-aliasing filters (a) or MATLAB and Python *decimate* functions (b) an (c).

in Fig. 9(a), the 2nd-stage Blackman-sinc filter attenuates the 1010-Hz interference by 79 dB. After decimation, the aliased mirror tone at 990 Hz is 79 dB weaker than the 1010-Hz interference tone. The 1100-Hz interference is attenuated by 89 dB so that the aliased mirror tone at 900 Hz is 89 dB weaker than the 1100-Hz interference tone. These strong attenuations effectively suppress aliasing.

As shown in Fig. 9(b), when the MATLAB *decimate* function is used for the 2nd-stage filtering and decimation, the aliased mirror tone at 990 Hz is only 11 dB weaker than the 1010-Hz interference tone (i.e., 8% of the interference tone's power is leaked into the passband). This is caused by the anti-aliasing filter's passband cutoff frequency being set too high, so that the 1010-Hz tone interference is only slightly attenuated. In practice, if there are strong acoustic sources (e.g., instrument noise or ambient sound) just above the Nyquist frequency (1 kHz in this case), the insufficient anti-aliasing will lead to aliased spectral content and possible errors in the analysis of the decimated data. The aliased mirror tone at 900 Hz is 57 dB weaker than the 1100-Hz interference tone, but the attenuation is not as strong as that provided by the 2nd-stage Blackman-sinc filter. The anti-aliasing performance of the Python *decimate* function is very close to that of the MATLAB *decimate* function, as shown in Fig. 9(c). The aliasing problem could be mitigated if the "*decimate*" functions are redefined to allow the user to

specify passband and stopband attributes of the anti-aliasing low-pass filter.

2. Concatenation and parallel processing

Audio recordings having a high sample rate are commonly stored in files of relatively short duration. This introduces a requirement to seamlessly concatenate output from decimation of data contained in separate original files, as well as an opportunity to accelerate processing of large data sets through parallel processing. An effective approach to concatenation is the "overlap-add" method (Smith, 1997; Oppenheim and Schafer, 1989), which allows separately processed segments to be seamlessly combined. This method requires that when each segment is convolved with the filter, the entire output of convolution be retained. For the sake of clarity, let us only consider filtering but not down-sampling. Suppose each segment has a length of K, and the FIR filter has L taps. Then the entire output of length K + L-1 must be retained. When the successive filtered segments are recombined, the length L-1 "tail" of one filtered segment is overlapped and added with the length L-1 "head" of the next filtered segment, and so on. Canned decimation functions cannot be applied in this way because they cut off the "tail" of the convolution output. If one simply concatenates the successive decimated segments in an attempt to reconstruct the decimated output of the original long data sequence, discontinuity errors will be introduced at the segment junctions because of the missing "tails" of the segments. Application of the overlap-add method has the further advantage of enabling effective parallelization of decimation processing.

IV. CONCLUSION

Effective and efficient decimation is essential to research using audio data. This contribution encompasses the major elements of decimator design required to achieve high quality and high efficiency. The quality of decimation output is defined by prevention of spectral leakage, exact control of the attenuation profile, and phase preservation. The efficiency of decimation processing relies on both the design of filters and the scalability of processing. By finding the optimal combination of the sinc function's cutoff frequency and the mainlobe bandwidth of the window function, we can design the shortest windowed-sinc low-pass filter that meets the signal retention and exclusion needs. Using the optimized Blackman-sinc filters as the anti-aliasing filters, we constructed a two-stage decimator for decimating hydrophone data. In contrast to the "canned" decimation functions in some software packages, the approach presented allows for complete control of the filters and a thorough understanding of their effects on the data. Further, application of the overlap-add method enables this approach to take full advantage of parallelization to accelerate the processing of voluminous passive acoustic monitoring data sets.

ACKNOWLEDGMENTS

This work was supported by the David and Lucile Packard Foundation. The authors appreciate the reviewer comments for improving the paper, and we thank Danelle Cline for support of the open-source implementation of the presented method.

[1]An open-source implementation of hydrophone data decimation is publicly accessible at https://docs.mbari.org/pacific-sound/notebooks/data/PacificSound256kHzTo2kHzDecimate/.

Crochiere, R., and Rabiner, L. (**1975**). "Optimum FIR digital implementations for decimation, interpolation, and narrow-band filtering," IEEE Trans. Acoust. Speech Signal Process. **23**(5), 444–456.

Crochiere, R. E., and Rabiner, L. R. (**1981**). "Interpolation and decimation of digital signals—A tutorial review," Proc. IEEE **69**(3), 300–331.

Helble, T., Ierley, G., D'Spain, G., Roch, M., and Hildebrand, J. (**2012**). "A generalized power-law detection algorithm for humpback whale vocalizations," J. Acoust. Soc. Am. **131**, 2682–2699.

Lyons, R. G. (**2011**). *Understanding Digital Signal Processing*, 3rd ed. (Pearson Education, Inc., Upper Saddle River, NJ).

Oestreich, W. K., Fahlbusch, J. A., Cade, D. E., Calambokidis, J., Margolina, T., Joseph, J., Friedlaender, A. S., McKenna, M. F., Stimpert, A. K., Southall, B. L., Goldbogen, J. A., and Ryan, J. P. (**2020**). "Animal-borne metrics enable acoustic detection of blue whale migration," Curr. Biol. **30**(23), 4773–4779.

Oppenheim, A. V., and Schafer, R. W. (**1989**). *Discrete-Time Signal Processing* (Prentice-Hall, Inc., Englewood Cliffs, NJ).

Oppenheim, A. V., Willsky, A. S., and Young, I. T. (**1983**). *Signals and Systems* (Prentice-Hall, Inc., Englewood Cliffs, NJ).

Ryan, J., Cline, D., Dawe, C., McGill, P., Zhang, Y., Joseph, J., Margolina, T., Caillat, M., DeVogelare, A., Stimpert, A., and Southall, B. (**2016**). "New passive acoustic monitoring in Monterey Bay National Marine Sanctuary: Exploring natural and anthropogenic sounds in a deep soundscape," in *Proceedings of MTS/IEEE Oceans 2016*, September 19–23, Monterey, CA, pp. 1–8.

Ryan, J. P., Joseph, J. E., Margolina, T., Hatch, L. T., Azzara, A., Reyes, A., Southall, B. L., DeVogelare, A., Peavey Reeves, L. E., Zhang, Y., Cline, D. E., Jones, B., McGill, P., Baumann-Pickering, S., and Stimpert, A. K. (**2021**). "Reduction of low-frequency vessel noise in Monterey Bay National Marine Sanctuary during the COVID-19 pandemic," Front. Mar. Sci. **8**, 1–13.

Saramaki, T. (**1984**). "A class of linear-phase FIR filters for decimation, interpolation, and narrow-band filtering," IEEE Trans. Acoust. Speech Signal Process. **32**(5), 1023–1036.

Simonis, A. E., Forney, K. A., Rankin, S., Ryan, J., Zhang, Y., DeVogelaere, A., Margolina, T., Joseph, J., Krumpel, A., and Baumann-Pickering, S. (**2020**). "Seal bomb noise as a potential threat to Monterey Bay harbor porpoise," Front. Mar. Sci. **7**(142), 1–9.

Širović, A., Rice, A., Chou, E., Hildebrand, J., Wiggins, S. M., and Roch, M. A. (**2015**). "Seven years of blue and fin whale call abundance in the Southern California Bight," Endangered Species Res. **28**, 61–76.

Smith, S. W. (**1997**). *The Scientist and Engineer's Guide to Digital Signal Processing* (California Technical Publishing, San Diego, CA).

Stoica, P., and Moses, R. (**1997**). *Introduction to Spectral Analysis* (Prentice-Hall, Upper Saddle River, NJ).