Fast and Accurate Audio Resampling for Acoustic Sensor Networks by Polyphase-Farrow Filters with FFT Realization

Aleksej Chinaev[1], *Gerald Enzner*[1], *Joerg Schmalenstroeer*[2]

[1]Institute of Communication Acoustics, Ruhr-Universität Bochum, 44780 Bochum, Germany
Email: {aleksej.chinaev, gerald.enzner}@rub.de
[2]Department of Communications Engineering, Paderborn University, 33100 Paderborn, Germany
Email: schmalen@nt.uni-paderborn.de

Abstract

Arbitrary sampling rate conversion has already received considerable attention in the past, but still lacks an equivalent representation of the effective time-dilation process in the block frequency domain. Good sampling rate converters in the time domain have been known, for instance, in terms of time-varying 'Sinc' or fixed 'Farrow' polynomial filters. The former can deliver nearly exact conversion at high complexity, while the latter has pronounced computational efficiency with limited accuracy. Only recently, it was shown that a composite 'polyphase Farrow' form with high resampling precision can be implemented with quasi-fixed filters that operate at the input sampling rate. We therefore propose to capitalize from that fixed-filter architecture in that we translate the polyphase-Farrow filters into an equivalent FFT-based overlap-save form. Experimental evaluation and comparison with other state-of-the art frequency-domain approaches then proves currently the best price-performance ratio of the proposed algorithm. It is thus an ideal candidate for the new framework of acoustic sensor networks that critically rests upon fast and accurate alignment of autonomous sampling processes.

1 Introduction

The notion of discrete-time in digital signal processing (DSP) fundamentally involves a demand for multirate signal processing (MSP) and conversion between the multiple rates. The MSP has received a lot of attention in the past, which is evident from a number of textbooks in the field [1–5]. New applications and infrastructure, such as acoustic sensor networks (ASNs) for distributed sensing and classification of sound [6], however, require new attention in order to meet application specific requirements of acoustic signal enhancement [7–9], sound source localization [10–12], acoustic scene classification [13–15] and network self-calibration [16, 17]. Here, acquisition of the sound field at multiple autonomous nodes specifically assumes low-power consumption and asynchronous treatment of the audio input in the first stage. Advanced sensing and classification performance is then expected from additional cooperative treatment of the individual audio inputs in a second stage. This structured approach to ASNs thus requires alignment of the asynchronous inputs with various time bases to a virtually common clock, even if the same nominal sampling frequencies are assumed in all nodes for the audio acquisition. It has been shown that sampling rate offset (SRO) as small as 50-100 ppm of the nominal sampling rate are sufficient for drift and deterioration of the audio signals in terms of spatial cues, correlation factors and processed quality [18–20]. Therefore, a synchronization of ASN to a common clock is absolutely necessary [19–22]. The required arbitrary sampling rate conversion (ASRC), however, needs to be accomplished with minimum computational load for maintaining the aforementioned low-power requirement.

Asynchronous sampling of analog audio input virtually corresponds to different dilations of the single continuous time scale before synchronous sampling of the signals. Or, from the digital viewpoint, the sampling times of asynchronous sampling processes actually drift along the continuous time to different sampling time instances. While the phenomenon of continuous-time dilation is frequently considered in the context of linear signals and systems, it is essentially highly nonlinear and cannot be compensated by means of linear filtering processes. In DSP, for instance, the lowpass interpolation of discrete-time series to arbitrary new sampling time instances requires a time-varying 'Sinc' interpolator which has to be updated at the output rate of the resampler [23]. This is initially a bad prerequisite for accomplishing a frame-wise conversion with FFT-support and thus low computational load. Other interpolators rely on polynomial representations for modeling a smooth transition of continuous waveforms [1–3]. The polynomial model may have nothing to do with the physical lowpass characteristic of the signal at hand, however, it provides a good computational basis. As an example, we here emphasize on the Farrow polynomial model to represent the celebrated Waring-Lagrange interpolation polynomials as FIR linear filters with just additional memoryless nonlinearity [24–27].

In this paper we reflect and rival three recent developments for advanced audio resampling in ASNs. The methods from [28], [29] and [30] are based on entirely different ideas, while claiming advances in terms of low complexity, high performance, and utility each. Owing to their entirely different motivation, structure, and parameters, this paper comprehensively compares those new techniques on the same data to work out their performances. Note, the required SRO estimation can be accomplished by several methods [18, 21, 22, 28, 31–33], but we assume an exact knowledge of SRO for the scope of this contribution.

The paper[1] is organized as follows: in Sec. 2 a time-domain Polyphase-Farrow resampler from [30] is recapitulated and its FFT-based implementation is newly introduced. Two further FFT-based resamplers presented in [28, 29] are briefly described in Sec. 3, before results of experimental comparison are presented in Sec. 4 and conclusions are drawn in Sec. 5.

2 Polyphase Farrow (PolyFar) ASRC

In this section, a PolyFar resampling method from [30] is recapitulated with its main components and involved signals. Based on this, an accurate and computationally efficient FFT-based PolyFar approach for ASRC is introduced.

2.1 PolyFar resampling in time domain

The key task of a digital-to-digital ASRC is the reconstruction of a discrete-time signal $y(n) = s(nT_y)$, sampled at a nominal sampling rate $f_y = 1/T_y$, from a discrete-time signal $x(m) = s(mT_x)$, sampled at another sampling rate $f_x = 1/T_x \neq f_y$, where $s(t)$ is often assumed to be a continuous-time bandlimited signal. The conversion ratio $f_x/f_y = \epsilon + 1$ defines an SRO ϵ that belongs to the set of irrational numbers [34].

A PolyFar resampler recently developed in [30] is a time-domain approach for solving such a task via timing alignment, during which sample estimates $\hat{y}(n)$ at the required time points $n \cdot T_y = (n + \delta(n)) \cdot T_x$ are calculated from the input signal $x(m)$, where $\delta(n) \in \mathbb{R}$ is an accumulating time drift between axis m and axis n. As depicted in Fig. 1, a PolyFar resampler consists of three main components: an input buffer, polyphase-Farrow FIR filters, and a polynomial chain. Every system component is controlled by a respective control signal $\mu(n) \in \mathbb{Z}$,

[1]This work was in part supported by *Deutsche Forschungsgemeinschaft* (DFG) under grant no. EN 869/3-1 and SCHM 3301/1-1 within the framework of the Research Unit FOR2457 "Acoustic Sensor Networks".

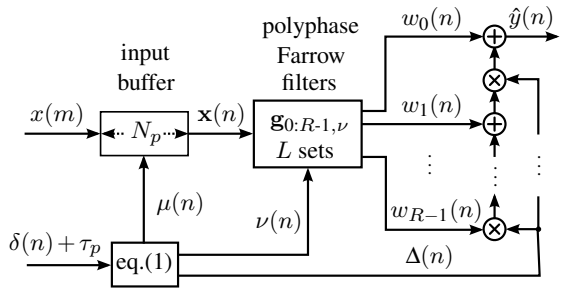

Figure 1: Block diagram of PolyFar resampler in time domain.

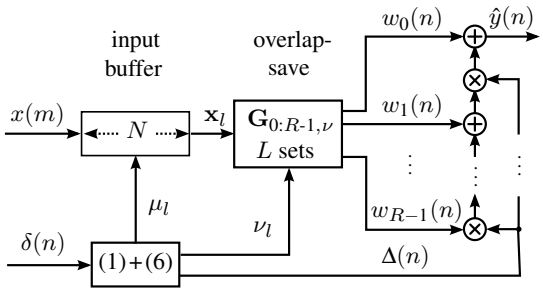

Figure 3: Block diagram of FFT-based PolyFar resampler.

$\nu(n) \in \{0, \dots L-1\}$ or $\Delta(n) \in \mathbb{R}$ on $[0,1)$, where L is the number of polyphase filter sets. The signals $\mu(n)$, $\nu(n)$ and $\Delta(n)$ are calculated from $\delta(n)$ via:

$$\mu(n) = \lfloor \delta(n) \rfloor, \;\; \nu(n) = \lfloor L \cdot \{\delta(n)\} \rfloor, \;\; \Delta(n) = \{L \cdot \{\delta(n)\}\}, \quad (1)$$

where $\lfloor \cdot \rfloor$ and $\{\cdot\}$ are the floor and fractional part functions. For a constant SRO ϵ, $\delta(n)$ can be easily calculated via $\delta(n) = \epsilon \cdot n$, when perfect synchronization at the beginning is assumed.

The input buffer of length N_p is moved across the input stream by the integer $\mu(n)$ taking care of a rough (down to one sampling point) synchronization of $\hat{y}(n)$ w.r.t. $x(m)$. It operates as a common delay line of every single polyphase filter and delivers on its output a column vector of length N_p:

$$\mathbf{x}(n) = [x(m-\mu(n)), \dots, x(m-\mu(n)-(N_p-1))]^{\mathrm{T}}. \quad (2)$$

For the FIR filtering, L polyphase-Farrow filter sets $\mathbf{g}_{0:R\text{-}1,\nu}$ for $\nu \in \{0, \dots, L-1\}$ are available. Which polyphase-Farrow filter set has to be used for the current time instance n is controlled by $\nu(n)$. One polyphase-Farrow filter set $\mathbf{g}_{0:R\text{-}1,\nu}$ consists of R subfilters with fixed filter coefficients $\mathbf{g}_{r,\nu}$ for $r \in \{0, \dots, R-1\}$, where R is both the number of branches in the underlying Farrow structure and the number of sampling points of the Lagrange polynomial interpolation part of the system. Polyphase subfilters of every branch $\mathbf{g}_{r,\nu}$ for a certain index r are calculated by polyphase decomposition of an FIR filter of length $N_p \cdot L$ resulting from a convolution of a lowpass filter with a correspondent branch of a Farrow-structured asynchronous polynomial interpolator [30]. The outputs $w_r(n) = \mathbf{g}_{r,\nu}^{\mathrm{T}} \cdot \mathbf{x}(n)$ of every subfilter are then used in the memoryless polynomial chain controlled by $\Delta(n)$ to calculate the output:

$$\hat{y}(n) = \sum_{r=0}^{R-1} w_r(n) \cdot \Delta^r(n). \quad (3)$$

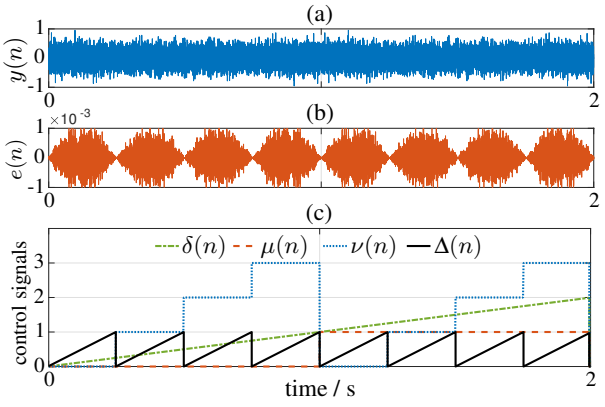

Figure 2: Time-domain PolyFar resampler with $N_p = 80$ and $L = R = 4$ in work for $f_y = 16\,\mathrm{kHz}$ and $\epsilon = 62.5\,\mathrm{ppm}$: (a) pseudo-random signal $y(n)$, (b) resampling error $e(n) = \hat{y}(n) - y(n)$ resulting in SINR $\approx 60\,\mathrm{dB}$, and (c) control signals from (1).

Note, an important implementation aspect of PolyFar resampler is considering the FIR filter delay $\tau_p = N_p/2$, which can be easily compensated by using $\delta(n) + \tau_p$ instead of $\delta(n)$.

Fig. 2 shows a PolyFar resampler in work and illustrates a correspondent course of its control signals. Here, two multi-tone pseudo-random signals $y(n)$ and $x(m)$ with frequencies in the range $[0;7]\,\mathrm{kHz}$ are generated for $f_y = 16\,\mathrm{kHz}$ and $\epsilon = 62.5\,\mathrm{ppm}$. Applying an equiripple FIR lowpass filter ($L = 4$, $N_p = 80$) and Waring-Lagrange coefficients ($R = 4$), a high signal-to-interpolation noise ratio (SINR) defined as in [29] is achieved. Fig. 2 reveals that interpolation errors are especially small in the vicinity of values $\Delta(n) \approx 0$ and $\Delta(n) \approx 1$, where a change of polyphase is taking place. This observation will be useful for switching polyphases in the FFT-based PolyFar resampler.

2.2 Implementation of PolyFar resampling in the block frequency domain

The linear FIR filtering of the PolyFar resampler in Fig. 1 can be efficiently implemented using FFT-based signal processing. For this, three requirements of FFT-based filtering should be met: 1) block-based signal processing, 2) appropriate FIR filter delay τ_p, and 3) fixed filter coefficients for one signal block.

For block-based filtering, the input buffer has to be increased up to a desired FFT-length $N > N_p$, as it is shown in Fig. 3. Its output vector can then be defined as

$$\mathbf{x}_l = [x(m-\mu_l-(N-1)), \dots, x(m-\mu_l)]^{\mathrm{T}}|_{m=l\cdot B}, \quad (4)$$

where l, μ_l and $B = N - N_p + 1$ are a block index, a control signal and the desired size of one output signal block, respectively. Next, $\mathbf{x}_l$ is filtered by an appropriate set of polyphase-Farrow filters $\mathbf{G}_{0:R-1,\nu}$ with $\mathbf{g}_{r,\nu} \overset{N\text{-FFT}}{\circ\!\!-\!\!\bullet} \mathbf{G}_{r,\nu}$ using the overlap-save technique [5]. For the data block l, R output vectors

$$\mathbf{w}_{r,l} = [w_r(n-(B-1)), \dots, w_r(n)]^{\mathrm{T}}|_{n=l\cdot B} \quad (5)$$

of length B for $r \in \{0, \dots, R-1\}$ can be defined as a result of FFT-based filtering. Components of $\mathbf{w}_{r,l}$ denoted in the following as $w_{r,l,b} = w_r(n-(B-b))|_{n=l\cdot B}$ with a subindex $b \in \{1, \dots, B\}$ are used as $w_r(n)$ in the polynomial chain, which is still controlled by the fractional delay signal $\Delta(n)$ at time n (i.e., not block-wise as μ_l and ν_l), as shown in Fig. 3. In our realization, μ_l and ν_l are calculated from $\mu(n)$ and $\nu(n)$ via:

$$\mu_l = \mu((l-1)\cdot B) \quad \text{and} \quad \nu_l = \nu((l-1)\cdot B), \quad (6)$$

with $\mu_1 = \nu_1 = 0$ for perfect synchronization at the beginning.

Assuming $\tau_p < B$, a filter delay τ_p can be taken into account by introducing R delayed vectors $\mathbf{w}_{r,l}^{\mathrm{del}}$ defined as:

$$\mathbf{w}_{r,l}^{\mathrm{del}} = [\underbrace{w_{r,l-1,\tau_p+1}, \dots, w_{r,l-1,B}}_{b_p = B-\tau_p}, \underbrace{w_{r,l,1}, \dots, w_{r,l,\tau_p}}_{\tau_p}]^{\mathrm{T}} \quad (7)$$

comprising of the last $b_p = B - \tau_p$ components of $\mathbf{w}_{r,l-1}$ from the previous block and of the first τ_p components of $\mathbf{w}_{r,l}$ from the

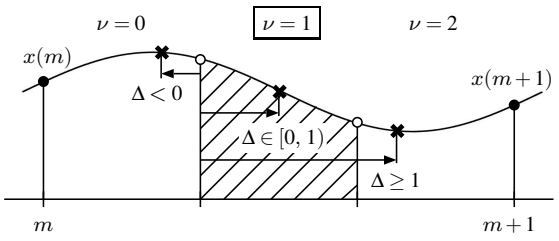

Figure 4: Illustration of Δ-extension for FFT-based implementation of PolyFar with $L=3$, $R=4$ and $\nu_l=1$: for $\nu(n)=0$ use $\Delta<0$; for $\nu(n)=\nu_l=1$ use $\Delta\in[0,1)$; for $\nu(n)=2$ use $\Delta\geq 1$.

current block. Now, a vector $\mathbf{\Delta}_l$ is defined in the same fashion as (5) consisting of the fractional delays for l-th block. In this case, the definition (7) allows us to use components $w^{\text{del}}_{r,l,b}$ of $\mathbf{w}^{\text{del}}_{r,l}$ for different r values and a component $\Delta_{l-1,b}$ of $\mathbf{\Delta}_{l-1}$ with the same index b for calculation of resampled output $\hat{y}(n)$ via (3).

In order to maintain a high precision of time-domain PolyFar resampling, we have also to deal appropriately with the fact that a change of the polyphase filter indicated by $\nu(n)$ can happen for any values of n. This means that signal samples at time instances with $\nu(n)\neq\nu_l$ are filtered with the adjacent (previous or next) polyphase filter instead of the target one. In this case, we propose to adjust some components of vector $\mathbf{\Delta}_{l-1}$, before they can be used in (3). For such components, we allow the fractional delay values in an adjusted vector $\mathbf{\Delta}^{\text{adj}}_{l-1}$ to leave their definition interval $[0,1)$ taking for a while a) values slightly bigger than 1 or b) slightly negative values, as shown in Fig. 4.

Which of both cases a) or b) should be applied for calculating the adjusted fractional delays, depends on the exact position of a polyphase change denoted in the following by b_c:

$$a)\ \text{if } b_c\leq b_p:\ \Delta^{\text{adj}}_{l-1,b}=\begin{cases}\Delta_{l-1,b}+1 \text{ for } b\in\{b_c,\ldots,b_p\}\\ \Delta_{l\text{-}1,b}, \quad \text{else}\end{cases} \tag{8}$$

resulting in values of adjusted components $\Delta^{\text{adj}}_{l-1,b}\geq 1$, or

$$b)\ \text{if } b_c>b_p{+}1\text{: }\Delta^{\text{adj}}_{l\text{-}1,b}=\begin{cases}\Delta_{l\text{-}1,b}-1,\ b\in\{b_p{+}1,\ldots,b_c{\text{-}}1\}\\ \Delta_{l\text{-}1,b}, \quad \text{else}\end{cases} \tag{9}$$

leading to values of adjusted components $\Delta^{\text{adj}}_{l-1,b}<0$. Both cases of proposed adjustment are visualized in Fig. 5. Note, the suggested corrections (8) and (9) do not lead to noteworthy resampling errors of the FFT-based PolyFar implementation, because a time-domain PolyFar approach exhibited a high resampling accuracy in time regions where a change of polyphase filter occurs, as shown in Fig. 2. The proposed solution is developed under the condition, that a change of the polyphase filter set happens only once within one block. This can be easily fulfilled by appropriate choice of the number of polyphase filter sets $L<1/(\epsilon\cdot B)$.

3 Reference methods

We utilize the STFT and the Overlap-Save method (OSM) method as reference methods, since both solve the issue of sampling rate offset compensation within the block frequency domain. In the following each method is briefly sketched and the applied approximations are discussed.

3.1 STFT method for ASRC

In [28] the authors proposed to handle the SRO as an effect which introduces a decreasing/increasing delay between two asynchronously sampled audio streams. If a signal $s(t)$ is sampled at two slightly different rates f_x and f_y, the STFTs $X(l,k)$ and $Y(l,k)$ of the two sampled signals $x(m)$ and $y(n)$ would be linked by

$$Y(l,k)\approx X(l,k)\cdot e^{-j\frac{2\pi}{N}(\frac{B}{2}+lB)k\epsilon}, \tag{10}$$

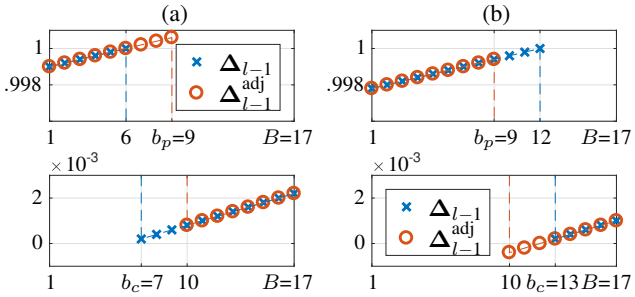

Figure 5: Extension of Δ-intervals in signal blocks with a polyphase change: (a) for $b_c\leq b_p$ with (8) and (b) for $b_c>b_p+1$ with (9). Examples for $\epsilon=50\,\text{ppm}$, $N=2^5$ and $N_p=16$ resulting in $B=17$, $\tau_p=8$ and $b_p=9$: (a) $b_c=7$ and (b) $b_c=13$.

where k the frequency bin, B the block size, and N the FFT size. The approximation quality degrades with increasing values of ϵ. In order to compensate for the SRO the STFT method applies an analysis window to the input stream (e.g., a Hann window), transforms the windowed data to the STFT domain, multiplies it with the complex conjugate version of the exponential function from (10), and subsequently transforms it back to the time domain for synthesis. Since time shifting by multiplication with phase terms in the STFT domain always introduces cyclic wrap around effects, the implementation has to split the temporal delay $(B/2+lB)\cdot\epsilon$ into two parts. The integer part of the delay has to be compensated by a buffer shift while the remaining fractional part is used in the exponential function [28, 29].

3.2 Overlap-save method for ASRC

The OSM follows the idea of signal reconstruction and subsequent sampling [29]. This "resampling-by-reconstruction"-method is also used by the Sinc method and well-known for its precision as well as its computational demands. The signal $x(m)$ has been sampled at time instances $t=m\cdot T_x$ and should be resampled at time instances $t=n\cdot T_y$ using

$$y(n)=\sum_{m=\tilde{m}-M}^{\tilde{m}+M} x(m)\ \text{sinc}\left((1+\epsilon)\,n-m\right), \tag{11}$$

where $\text{sinc}(z)=\sin(z)/z$, M is the window size parameter and $\tilde{m}$ is the sample in $x(m)$ which is temporally closest to the n-th sample (i.e., time $n\cdot T_y$).

In order to speed up the approach and in parallel keep the precision, (11) has to be transformed into a linear convolution such that an efficient frequency-domain implementation is possible. To this end, a zero addition $\epsilon\cdot m-\epsilon\cdot m=0$ is introduced into the argument of the sinc function and the expression $\epsilon\cdot m$ is approximated by its middle value in the block as $\epsilon\cdot m\approx\epsilon\cdot\tilde{m}$:

$$y(n)\approx\sum_{m=\tilde{m}-M}^{\tilde{m}+M} x(m)\ \text{sinc}\left((1+\epsilon)(n-m)+\epsilon\cdot\tilde{m}\right). \tag{12}$$

Equation (12) states a linear convolution between the input block samples $x(m)$ and an interpolation function $h(m)=\text{sinc}\left((1+\epsilon)m+\epsilon\cdot\tilde{m}\right)$. It can be efficiently handled by an Overlap-save technique in the frequency domain.

The OSM benefits from the broadband signal handling capabilities of the ideal reconstruction approach (11), as the approximation errors of the sinc function decreases with an increasing window size M, which in case of the OSM results in an increased FFT size. However, in order to achieve precise results the OSM has to keep the block size B small, such that the SRO effects within a block remains negligible. Hence, the block size B has to be chosen with respect to the SRO and the targeted precision. An elaborated description of the OSM and its implementation details can be found in [29].

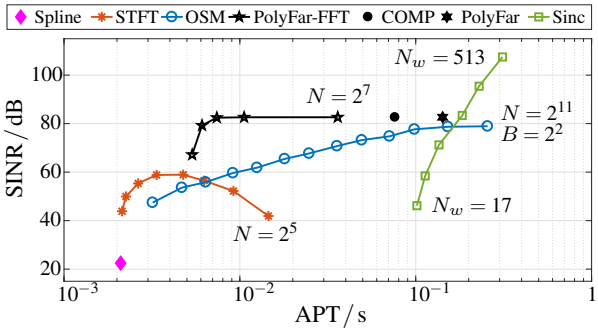

Figure 6: Resampling precision in terms of SINR values plotted over an average processing time per one second of input data averaged over all simulated SRO values for speech-band data set.

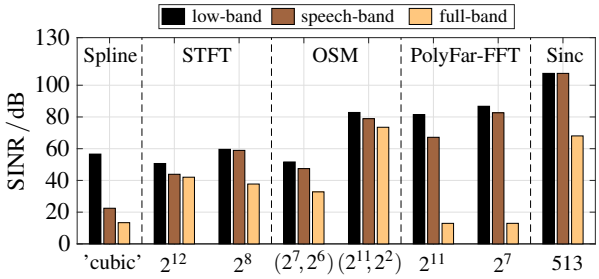

Figure 7: Resampling precision in terms of SINR values averaged over all simulated SRO values for low-band, speech-band and full-band data sets. Parameterization of resamplers is given on x-axis: FFT size N for STFT and PolyFar-FFT; parameters (N, B) for OSM with block size B; window size N_w for Sinc.

4 Experimental investigation

We consider band-limited discrete-time multi-tone pseudo-random signals artificially generated for constant SRO values $\epsilon \in \{5, 25, 50, 75, 100, 150, 200\}$ ppm at a nominal sampling rate of $f_y = 16$ kHz to provide input data for resampling and reference data for performance assessment. Three data sets are produced: low-band, speech-band and full-band. Signals of every data set are composed of components in a frequency range $[50\,\text{Hz}; f_u]$ for the upper frequencies $f_u \in \{3\,\text{kHz}, 6.5\,\text{kHz}, 7.95\,\text{kHz}\}$, respectively, and are perfectly synchronized at the beginning. Every data set contains 424 signals of duration between 10 and 30 seconds resulting in a total length of approximately 142 minutes per data set. Resampling precision is measured using a signal-to-interpolation noise ratio defined as in [29]. Further, an averaged processing time (APT) per one second of input data is calculated to compare computational efficiency of the resamplers under investigation[2].

In addition to the PolyFar methods from Sec. 2 and to STFT and OSM approaches from Sec. 3, two time-domain resamplers, Spline interpolation and Sinc method, are included in the experimental evaluation. While the Spline approach implements a simple piecewise cubic spline interpolation, the Sinc method applies the Hann windowed sinc function of length $N_w = 2M + 1$ for $M \in \{2^3, \ldots, 2^8\}$. The STFT resampler is implemented for FFT size $N \in \{2^5, \ldots, 2^{12}\}$ with a block length $B = N/2$ and block shift $N/4$. The OSM resampler is realized for different (N, B) combinations of FFT size N and block length B chosen from values $N \in \{2^7, \ldots, 2^{11}\}$ and $B \in \{2^2, \ldots, 2^6\}$. The time-domain PolyFar is realized with $N_p = 80$, $L = 8$ and the lowpass filter designed using the Parks-McClellan method for equiripple FIR filter design with a passband cutoff frequency $f_p = 7$ kHz and a stopband cutoff frequency $f_s = 8$ kHz. For asynchronous interpolation, the Waring-Lagrange coefficients for $R = 4$ are used. For the same configuration as PolyFar, its ancestor the conventional (two-stage) composite 'COMP' resampler [2, 35] and the proposed FFT-based PolyFar for FFT size $N \in \{2^7, \ldots, 2^{11}\}$ and for the number of polyphase filter sets $L = \min(\lfloor 1/(\epsilon \cdot B) \rfloor, L_{\max})$ with $L_{\max} = 8$ are evaluated.

Resulting SINR and APT values averaged over all simulated SRO values are presented in Fig. 6 for speech-band data. The Spline resampler is admittedly very fast, however, it shows a lack of resampling precision. In contrast, the Sinc method has resampling accuracy growing quickly with an increasing window size N_w. Nevertheless, compared to the Spline approach, it is very slow, because of its high computational demand. The two time-domain approaches COMP and PolyFar still require relatively large APT and show equal precision as expected. In order to speed up a resampling process, FFT-based approaches should be used. One of them is the STFT resampler, which is almost as fast as the Spline method. While increasing of FFT size up to $N = 2^{12}$ here leads to a successive reduction of processing time, the best resampling performance of 60 dB is achieved for values N between 2^8 and 2^9. Better precision can be gained by the OSM resampler. A high SINR is obtained here for large FFT sizes (leading to longer sinc-filters) accompanied by shorter output block lengths (for smaller deviations of true fractional delay values from its middle value averaged over one block). In this way, the OSM resampler serves as a flexible link between STFT and Sinc interpolations. By the choice of the block size it can trade SINR performance against processing time APT in proportional terms. The diagram finally shows that the proposed FFT-based PolyFar resampler starts a new dimension in terms of the price-performance ratio of the resampling process w.o. loss of precision of its time-domain realization for $N \leq 2^9$. The PolyFar-FFT reaches out furthest to the top left of the diagram which implies both high SINR of 80 dB and still moderate processing time. It does, however, not cover the extreme points of operation in terms of lowest processing time or highest accuracy.

Fig. 7 presents SINR values of the different sampling rate converters with additional attention to the bandwidth of the input signals. Here, for each FFT-based resampling method (OSM, STFT, PolyFar-FFT) only two representatives (i.e., with different parameters) are depicted: the first with the fastest processing time and the second with the best resampling precision for speech-band data, as shown in Fig. 6. The band-limitation as of the low-band signal obviously supports all resamplers. Consistently also, the speech-band still supports higher SINR than the full-band, as expected. Amongst the FFT-based methods, the diagram once more confirms the currently best balance between computational cost and interpolation precision of the FFT-based PolyFar approach for speech-band and low-band signals. Low SINR of the PolyFar-FFT method with full-band signals here indicates the particular configuration of its discrete-time lowpass filter to speech-band signals, specifically, the fixed 1 kHz transition bandwidth. This is, however, not a fundamental limitation, since the monolithic PolyFar system can be tailored to the bandwidth at hand by appropriate design of the discrete-time lowpass filter of the underlying composite interpolation architecture [30].

5 Conclusions

In this contribution, a fast implementation of the time-domain approach from [30] for arbitrary sampling rate conversion has been derived using an efficient FFT-based realization of polyphase-Farrow filters with quasi-fixed filter coefficients. Maintaining a high resampling precision of the original time-domain method of up to 80 dB for speech-band signals, the new FFT-based resampler has low to moderate computational load in order of the STFT approach. An experimental comparison of the proposed resampler, baptized with the name FFT-based PolyFar, with some other state-of-the-art FFT-based resampling techniques confirms its superiority for processing speech-band signals - a task typically encountered in ad-hoc acoustic sensor networks.

[2] An Intel CPU i5-4590 @ 3.30 GHz computer was used and all resamplers are implemented and executed in Matlab.

References

[1] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice Hall, 1983.

[2] H. G. Göckler and A. Groth, *Multiratensysteme: Abtastratenumsetzung und digitale Filterbänke*. Schlembach Fachverlag, 2004.

[3] P. Prandoni and M. Vetterli, *Signal Processing for Communications*. EPFL Press, 2008.

[4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Pearson Education, India, 1993.

[5] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications (3. ed.)*. Prentice-Hall Int. Corp., 1996.

[6] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. of IEEE Symp. on Commun. and Vehicular Technology*, pp. 1–6, Nov. 2011.

[7] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Advances in Signal Process.*, 2009.

[8] M. Taseska, S. Markovich-Golan, E. Habets, and S. Gannot, "Near-field source extraction using speech presence probabilities for ad hoc microphone arrays," in *Proc. of Int. Workshop on Acoustic Signal Enhanc.*, pp. 170–174, Sept. 2014.

[9] S. M. Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Process.*, vol. 107, pp. 4–20, 2015.

[10] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, and A. Mouchtaris, "Localizing multiple audio sources in a wireless acoustic sensor network," *Signal Process.*, vol. 107, pp. 54–67, 2015.

[11] A. Brendel and W. Kellermann, "Localization of multiple simultaneously active sources in acoustic sensor networks using ADP," in *Proc. of Int. Workshop on Comput. Advances in Multi-Sensor Adapt. Process.*, pp. 1–5, Dec. 2017.

[12] A. Brendel and W. Kellermann, "Learning-based acoustic source-microphone distance estimation using the coherent-to-diffuse power ratio," in *Proc. of IEEE Int. Conf. on Acoustic, Speech and Signal Process.*, Apr. 2018.

[13] S. Gergen, A. M. Nagathil, and R. Martin, "Classification of reverberant audio signals using clustered ad hoc distributed microphones," *Signal Process.*, vol. 107, pp. 21–32, 2015.

[14] P. Arora and R. Haeb-Umbach, "A study on transfer learning for acoustic event detection in a real life scenario," in *Proc. of IEEE Int. Workshop on Multimedia Signal Process.*, Oct. 2017.

[15] J. Ebbers, A. Nelus, R. Martin, and R. Haeb-Umbach, "Evaluation of modulation-MFCC features and DNN classification for acoustic event detection," in *Proc. of Jahrestagung für Akustik*, Mar. 2018.

[16] P. Pertilä, M. S. Hämäläinen, and M. Mieskolainen, "Passive temporal offset estimation of multichannel recordings of an ad-hoc microphone array," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 21, pp. 2393–2402, Nov 2013.

[17] M. Parviainen, P. Pertilä, and M. S. Hämäläinen, "Self-localization of wireless acoustic sensors in meeting rooms," in *Proc. of IEEE Joint Workshop on Hands-free Speech Commun. and Microphone Arrays*, pp. 152–156, 2014.

[18] D. Cherkassky, S. Markovich-Golan, and S. Gannot, "Performance analysis of MVDR beamformer in WASN with sampling rate offsets and blind synchronization," in *Proc. of European Signal Process. Conf.*, pp. 245–249, Aug. 2015.

[19] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Process. Magazine*, vol. 28, pp. 124–138, Jan. 2011.

[20] S. Wehr, I. Kozintsev, R. Lienhart, and W. Kellermann, "Synchronization of acoustic sensors for distributed ad-hoc audio networks and its use for blind source separation," in *Proc. of IEEE Int. Symp. on Multimedia Software Engineering*, pp. 18–25, Dec. 2004.

[21] J. Schmalenstroeer, P. Jebramcik, and R. Haeb-Umbach, "A combined hardware-software approach for acoustic sensor network synchronization," *Signal Process.*, vol. 107, pp. 171–184, July 2015.

[22] D. Cherkassky and S. Gannot, "Blind synchronization in wireless acoustic sensor networks," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 25, pp. 651–661, Mar. 2017.

[23] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, N.J., Prentice-Hall, 1975.

[24] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *in Proc. of the IEEE*, vol. 90, pp. 319–342, May 2002.

[25] C. W. Farrow, "A continuously variable digital delay element," in *Proc. of IEEE Int. Symp. on Circuits and Systems*, pp. 2641–2645, June 1988.

[26] E. Waring, "Problems concerning interpolations," *Philosophical Trans. of the Royal Society of London*, vol. 69, pp. 59–67, Jan. 1779.

[27] J. L. Lagrange, "Leçons élémentaires sur les mathématiques données à l'école normale en 1795," in *Œuvres complètes*, vol. 7, pp. 183–287, Paris, France: Gauthier-Villars, 1877.

[28] S. Miyabe, N. Ono, and S. Makino, "Blind compensation of interchannel sampling frequency mismatch for ad hoc microphone array based on maximum likelihood estimation," *Signal Process.*, vol. 107, pp. 185 – 196, Sept. 2015.

[29] J. Schmalenstroeer and R. Haeb-Umbach, "Efficient sampling rate offset compensation - an Overlap-Save based approach," in *Proc. of European Signal Process. Conf.*, Sept. 2018.

[30] A. Chinaev, P. Thuene, and G. Enzner, "Low-rate Farrow structure with discrete-lowpass and polynomial support for audio resampling," in *Proc. of European Signal Process. Conf.*, Sept. 2018.

[31] L. Wang and S. Doclo, "Correlation maximization-based sampling rate offset estimation for distributed microphone arrays," *IEEE Trans. on Speech and Language Process.*, vol. 24, pp. 571–582, Mar. 2016.

[32] M. H. Bahari, A. Bertrand, and M. Moonen, "Blind sampling rate offset estimation for wireless acoustic sensor networks through weighted least-squares coherence drift estimation," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 25, no. 3, pp. 674–686, 2017.

[33] J. Schmalenstroeer, J. Heymann, L. Drude, C. Boeddecker, and R. Haeb-Umbach, "Multi-stage coherence drift based sampling rate synchronization for acoustic beamforming," in *Proc. of IEEE Int. Workshop on Multimedia Signal Process.*, Oct. 2017.

[34] T. A. Ramstad, "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Trans. on Acoustic, Speech, and Signal Process.*, vol. 32, pp. 577–591, June 1984.

[35] G. Evangelista, "Design of digital systems for arbitrary sampling rate conversion," *Signal Process.*, vol. 83, pp. 377–387, Feb. 2003.