20th IMEKO TC4 International Symposium and
18th International Workshop on ADC Modelling and Testing
Research on Electric and Electronic Measurement for the Economic Upturn
Benevento, Italy, September 15-17, 2014

Design of a flexible filter for synchronous resampling in wireless sensor networks

Jürgen Funck and Clemens Gühmann

TU Berlin, Chair of Electronic Measurement and Diagnostic Technology, 10587 Berlin, Germany,
E-Mail: juergen.funck@tu-berlin.de

Abstract – **Many tasks in the context of data acquisition with a wireless sensor network, e.g. jitter compensation or angular resampling, can be solved by reconstructing a uniformly sampled signal from a nonuniformly sampled one. A filter structure that is capable of performing this task is adopted from the literature and a practical design is evaluated. It is verified that its implementation on a resource constrained wireless sensor node is feasible. Two simulation examples demonstrate the effectiveness of the designed filter and a comparison with alternative resampling methods is given.**

I INTRODUCTION

The resampling of data that has been obtained on a nonuniform sampling grid to a uniform one can be used to solve various problems in data acquisition with wireless sensor networks. Firstly, it has been argued in [1] that in order to obtain a set of synchronized samples from a wireless sensor network, it may be beneficial to sample signals in an unsynchronized way first and resample them to a synchronized sampling grid later. This approach has the advantages that it matches better with irregular network communication patterns and helps to reduce interdependencies between different software modules [1]. Due to clock jitter or non-linear drift of the unsynchronized sampling clock, the unsynchronized samples, in general, will have been obtained on a nonuniform sampling grid [1]. Thus, resampling to the usually uniform global sampling grid involves the recovery of a uniformly sampled signal from a nonuniform set of samples. Another application where a nonuniform set of samples has to be transformed to a uniform one, is the resampling of a signal that has been sampled uniformly in time to a signal that is sampled uniformly with reference to the rotation angle of a machine. This is a common procedure in machine diagnosis [2]. Very similar to that is the removal of doppler shifts from acquired signals [2].

This paper describes the design of a filter that can be used to resample nonuniform samples to a uniform sampling grid. Section II presents a filter for the reconstruction of uniform samples from a nonuniform set of samples. Section III gives a design example and shows that it is feasible to implement that filter on a resource constrained wireless sensor node. The simulations presented in section IV show that the designed filter is suitable for the resampling of jittered signals and chirp signals acquired during a speed-sweep of a rotating machine. An evaluation of the filter as well as a comparison to alternative resampling methods is given in section V.

II A FILTER FOR UNIFORM SAMPLE RECOVERY

This section introduces a filter that can be used to recover uniform samples from a set of nonuniform ones. Section II.A outlines the general spectrum of a nonuniformly sampled signal and deduces a reconstruction method from that. An implementation of this method as a digital filter is described in section II.B.

II.A Spectrum of a nonuniformly sampled signal

According to [3] the signal x_s sampled at the nonuniform time instants t_k can be modeled as the product of the original signal x and the comb function x_p:

$$x_s(t) = x(t)x_p(t) \tag{1}$$

$$x_p(t) = \sum_{k=-\infty}^{\infty} \delta(t - t_k) \tag{2}$$

where δ is the dirac delta function. The fourier series expansion of x_p yields [3]:

$$x_p(t) = \frac{\left|1 - \dot{\Theta}(t)\right|}{T} \left[1 + 2\sum_{k=1}^{\infty} \cos\left(\frac{2\pi kt}{T} - \frac{2\pi kt\Theta(t)}{T}\right)\right] \tag{3}$$

$$x_s(t) = x(t)\frac{\left|1 - \dot{\Theta}(t)\right|}{T} \left[1 + 2\sum_{k=1}^{\infty} \cos\left(\frac{2\pi kt}{T} - \frac{2\pi kt\Theta(t)}{T}\right)\right] \tag{4}$$

where T is the uniform sampling interval and $\Theta(t_k) = t_k - kT$ the deviation from the uniform sampling instants.

If the bandwidth of $\Theta(t)$ is less than $\frac{1}{2T}$, lowpass filtering of x_s and x_p yields [3]:

$$x_{p,lp}(t) = \frac{\left|1 - \dot{\Theta}(t)\right|}{T} \tag{5}$$

$$x_{s,lp}(t) = x(t)\frac{\left|1 - \dot{\Theta}(t)\right|}{T} \tag{6}$$

And the original signal can be recovered as [3]:

$$\tilde{x}(t) = \frac{x_{s,lp}(t)}{x_{p,lp}(t)} \approx x(t) \tag{7}$$

ISBN-14: 978-92-990073-2-7

II.B Filter Structure for Recovery

A filter structure that implements the reconstruction method outlined above was proposed in [4]. Its block diagram is shown in Fig. 1. The input signal sampled at the time instants t_k is denoted as $x(t_k) = x[k]$.

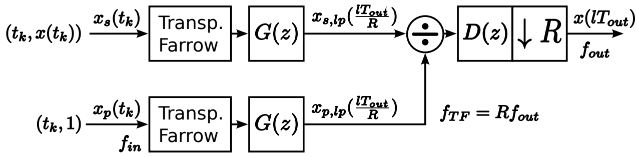

Fig. 1. Reconstruction filter proposed in [4]

Transposed farrow filters are used to do the low-pass filtering of x_s and x_p and to translate the samples to a common uniform sampling grid with the sampling interval T_{out}/R. Two additional FIR lowpass filters with the transfer function $G(z)$ are used to improve the low-pass filtering before the reconstruction with a divisor. Finally, the reconstructed signal is filtered by a FIR decimator with the transfer function $D(z)$ and decimation ratio R.

The transposed farrow filter [5] implements the hybrid filter model shown in Fig. 2. Conceptually, the discrete signal $x[k]$ is used to modulate a delta comb resulting in a continuous signal $x(t_k)$. This signal is then filtered using a continuous filter with the impulse response h_a. Finally, the filter output is sampled at the time instants lT_{out} producing the resampled discrete signal $y[l] = y(lT_{out})$.

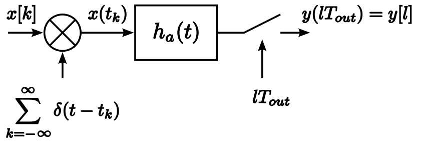

Fig. 2. Hybrid discrete/continuous model implemented by the transposed farrow filter [5]

In a transposed farrow filter the impulse response h_a is a piecewise polynomial. This way the hybrid model shown in Fig. 2 can be implemented by the completely discrete filter structure shown in Fig. 3.

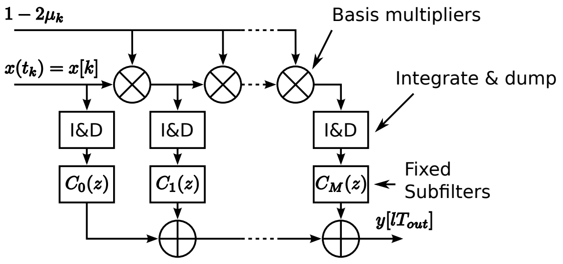

Fig. 3. Structure of the transposed farrow filter[4]

All input sampling instants t_k are expressed as:

$$t_k = (\mu_k + l_k)T_{out}. \tag{8}$$

where l_k is the integer index of the output sample that occurs closest before t_k and μ_k is called the fractional interval [5]. It specifies the position of an input sample relative to the output sampling interval and is an input to the transposed farrow filter.

A transposed farrow filter whose impulse response h_a consists of N polynomial segments of the polynomial order M has $M + 1$ parallel branches. Each is made up of an integrate-and-dump block and a fixed FIR subfilter $C_m(z)$ of the filter order $N - 1$. The filter coefficients of $C_m(z)$ can be obtained directly from the polynomial coefficients of h_a [5]. The input samples $x[k]$ are multiplied with powers of $(1 - 2\mu_k)$ and the results are accumulated in the integrate-and-dump blocks. At every output sampling instant lT_{out} their outputs are fed into the subfilters $C_m(z)$ and the accumulators of the integrate-and-dump blocks are reset. The output of the farrow filter is obtained by summing over the outputs of all subfilters.

The transposed farrow structure is very efficient in resampling signals when the output sampling rate is lower than the input sampling rate. It avoids aliasing in the baseband by having zeros at multiples of the output sampling rate, but can cause significant aliasing in transitions bands. Therefore the FIR lowpass filters $G(z)$ are used to narrow the transition bands and to remove transition band noise [4]. It was observed in [4] that the output sampling rate of the transposed farrow filters should be at least four times the bandwidth of the signal in order to achieve a good reconstruction quality. Thus, the FIR decimator $D(z)$ is used (see Fig. 1).

III FILTER DESIGN FOR A WIRELESS SENSOR NETWORK

In this section a design of the filter presented in the previous section for a wireless sensor network is given. Section III.A explains the network layout and the integration of the filter. The design of filter coefficients is described in section III.B. In section III.C an estimate of the filters computational requirements is given.

III.A Proposed layout for wireless sensor network

Fig. 4 shows the proposed wireless sensor network layout. Nodes sample the input signals independently based on their local clocks. The local clocks of the sensor nodes are also used to assign timestamps t_k to the individual samples. The base node transmits timestamps t_l that specify the global synchronized time grid. They are transferred to the local timescale of the sensor node using a-posteriori synchronization during the transmission process (compare [1]). Alternatively the clocks of the two nodes can also be synchronized beforehand using a-priori synchronization.

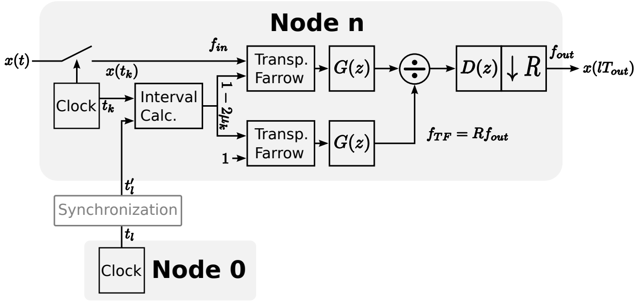

Fig. 4. Proposed layout for wireless sensor network

In case of time synchronized resampling the fractional interval μ_k can be calculated by the following equation if the timestamps t_k have been obtained from an a-priori synchronized clock [5]:

$$\mu_k = \frac{t_k}{T_{out}} - l_k \tag{9}$$

where t_k is the timestamp of the k^{th} input sample, T_{out} the output sampling period and l_k the index of the output sample at or before the k^{th} input sample.

If a-posteriori synchronization is used, the timestamps t_k have to be transformed to the global time grid using the transmitted timestamps t'_l and linear interpolation:

$$t'_k = \frac{T_{out}}{t'_{l_k+1} - t'_{l_k}} \left(t_k - t'_{l_k+1}\right) + l_k T_{out} \tag{10}$$

Then the fractional interval can be expressed as:

$$\begin{aligned}\mu_k &= \frac{t'_k}{T_{out}} - \lfloor \frac{t'_k}{T_{out}} \rfloor && (11)\\ &= \frac{t'_k - t'_{l_k}}{t'_{l_k+1} - t_{l_k}} && (12)\end{aligned}$$

In the case of angular resampling the time instants $t_{\phi,l}$ transmitted by the base node correspond to a uniform angular grid with a spacing of $\Delta\phi$. For the resampling the time instants t_k in the filter structure (Fig. 1) are replaced with angular instants $\phi_k = \phi(t_k)$ which are calculated by linear interpolation:

$$\phi_k = \frac{\Delta\phi}{t_{\phi,l_k+1} - t_{\phi,l_k}}(t_k - t_{\phi,l_k}) + l_k\Delta\phi \tag{13}$$

Then the fractional interval can be expressed as:

$$\begin{aligned}\mu_k &= \frac{\phi_k}{\Delta\phi} - \lfloor \frac{\phi_k}{\Delta\phi} \rfloor && (14)\\ &= \frac{t_k - t_{\phi,l_k}}{t_{\phi,l_k+1} - t_{\phi,l_k}} && (15)\end{aligned}$$

III.B Design of resampling filter

The resampling filter was designed assuming an input frequency $f_{in} \geq 2f_{out}$ and a signal bandwidth of $W_{sig} = f_{out}/4$. Thus the decimation factor at the output stage was $R = 2$.

The coefficients for the transposed farrow filter were taken from [5]. The impulse response of this filter consists of $N = 4$ polynomials of the order $M = 3$. This filter is sufficient for the targeted applications. Yet, better results, i.e. less passband ripple and higher stopband attenuation, could probably be achieved using the method for the calculation of optimal coefficients described in [6]. The FIR filter G at the output of the farrow filter was designed with the help of MATLAB using an optimal least-square design. Its passband characteristic was designed in such a way that it compensates for the passband ripple of the transposed farrow filter. It has a filter order of $N = 61$. The Output FIR decimator D was designed with MATLAB using an optimal equiripple design. It has a filter order of $N = 34$. The magnitude frequency responses of the individual filters are shown in Fig. 5 that of the entire filter cascade in Fig. 6. The latter neglects effects from demodulation by the divisor.

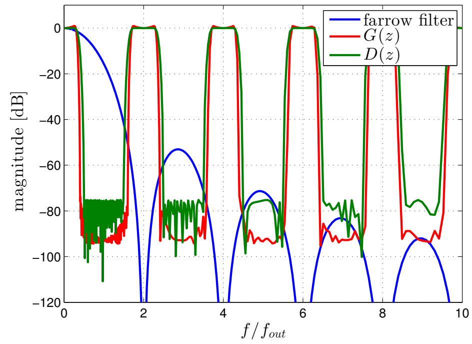

Fig. 5. magnitude frequency responses of individual filters

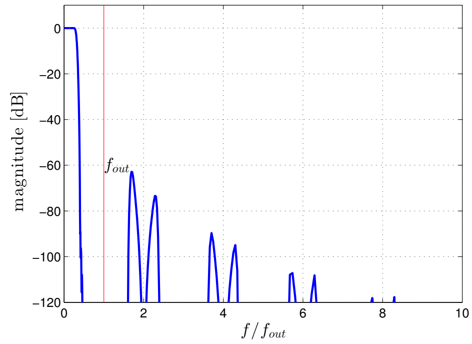

Fig. 6. magnitude frequency response of cascade: transposed farrow, G and D

III.C Computational requirements

To verify that it is feasible to implement the designed filter on a resource constrained wireless sensor node in real-time, its operation rate, i.e. the frequency f_{op} of arithmetic operations, was estimated. The targeted sensor node is a Preon32 by Virtenio GmbH. Its processor is a 32 Bit ARM Cortex-M3 whose maximum clock frequency is $f_{clk} = 72\,\mathrm{MHz}$. Previous experiments with a polyphase FIR decimator implemented using 16 Bit fixed-point arithmetic have shown that on average less than 3 clock cycles are needed for one arithmetic operation [7]. Thus, the processor is able to maintain a maximum operation rate of $f_{op,max} = 24\,\mathrm{MHz}$.

To calculate the fractional interval for angular resampling two subtractions and one division are needed (see equation 15). An additional subtraction and a multiplication are needed to calculate the term $1 - 2\mu_k$, resulting in an operation rate of:

$$f_{op,\mu,\phi} = 5 f_{in} \tag{16}$$

According to [8] the operation rate of a transposed farrow filter is:

$$f_{op,farrow} = (2M+1) f_{in} + \left[\frac{3}{2}N(M+1) - 1\right] f_{out} \tag{17}$$

Where M is the polynomial order and N the number of polynomial segments. The operation rate of a FIR filter with the order N implemented in direct form is [9]:

$$f_{op,fir} = (2N-1) f_{in} \tag{18}$$

For a FIR decimator implemented efficiently in polyphase form the operation rate is [9]:

$$f_{op,dec} = (2N-1) f_{out} \tag{19}$$

Table 1 gives a summary of the operation rates for the filters in the design example in case of an angular resampling. The input sampling rate was set to be $f_{in} = 20\,\mathrm{kHz}$ and the angular sampling rate 25 Samples/revolution. At a rotational speed of $n = 1500\,\mathrm{rpm}$ this results in an output sampling rate of $f_{out} = 625\,\mathrm{Hz}$.

Table 1. Operation rate of filter for angular resampling and CPU-load at $f_{clk} = 72\,\mathrm{MHz}$

filter	M	N	f_{in}	f_{out}	f_{op}	CPU-load
	[−]	[−]	[kHz]	[kHz]	[kHz]	[%]
μ-calc.			20	20	100.0	0.4
2 x farrow	3	4	20	1.25	308.8	1.3
2 x G	-	61	1.25	1.25	302.5	1.3
D	-	34	1.25	0.625	41.9	0.2
total					**753.2**	**3.1**

The CPU-load by the entire proposed filter is only 3.1 % leaving plenty of room for other operations or to decrease the clock frequency in order to save energy.

IV SIMULATIONS AND RESULTS

In this section the effectiveness of the designed filter is demonstrated using two application examples: resampling a jittered signal and resampling a chirp signal. Both examples are oriented towards a real setup at a machine test bench and use an input sampling rate of $f_{in} = 20\,\mathrm{kHz}$.

IV.A Resampling jittered samples

A sine signal with a frequency of 150 Hz and an amplitude of 1 arbitrary unit was simulated over 64 periods. The sampling jitter was created by adding white noise having a gaussian distribution with zero mean and a standard deviation of $\sigma_t = \frac{T_{in}}{10} = 5\,\mu\mathrm{s}$ to the ideal sampling instants. Using the residuals of a sinewave fit its signal-to-noise-and-distortion ratio SINAD was estimated to be $\mathrm{SINAD}_{in} = 46.5\,\mathrm{dB}$.

The signal was fed into the described resampling filter using an output sampling rate of $f_{out} = \frac{f_{in}}{32} = 625\,\mathrm{Hz}$. The SINAD of the output signal was estimated to be $\mathrm{SINAD}_{out} = 63.5\,\mathrm{dB}$. Thus the filter is effective in reducing signal noise introduced by jitter.

IV.B Resampling chirp signals

The resampling of chirp signals is often used to test angular resampling algorithms as it is representative of the data acquired from a rotating machine during speed change [2]. If angular resampling is successful, the chirp signal at the input is transformed into a stationary sine signal at the output. Fig. 7 shows the block diagram of the simulation setup.

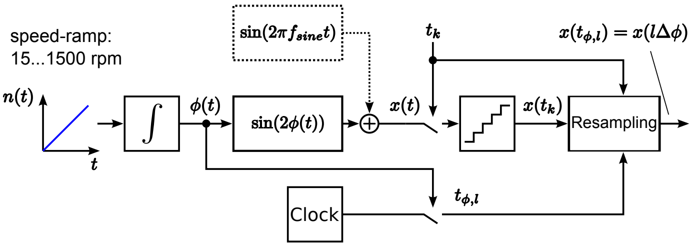

Fig. 7. Setup for simulations on chirp resampling

A linear speed change of $n = 15 \ldots 1500\,\mathrm{rpm}$ within 2 s is generated. Integrating this over time the rotation angle $\phi(t)$ is obtained. This is feed into a sine function so that a linear chirp signal at twice the rotation frequency is generated. The chirp signal is sampled at time instants t_k distributed uniformly over time and quantized with a resolution of 12 Bit. This corresponds to the properties of the analog-to-digital converter of the Preon32. The rotation angle $\phi(t)$ is also used to sample the system clock at 25 equidistant angular intervals per revolution, generating the timestamps $t_{\phi,l} = t(l\Delta\phi)$. In a practical experiment the same could be achieved using a rotary encoder and the internal clock of a sensor node. The timestamps $t_{\phi,l}$ are quantized with a resolution of $\Delta t = 2^{-14}\,\mathrm{s}$. This corresponds to the resolution of the real-time clock used in the Preon32 sensor node. Finally, angular resampling is performed to obtain the signal $x(l\Delta\phi)$ sampled uniformly over the rotation angle. Two resampling methods are used: the first is sample-and-hold resampling which takes the input closest to the desired angular instant, the second is the resampling filter described in this paper.

The SINAD of the output signal was estimated to be $SINAD_{SH,0} = 45.1\,\mathrm{dB}$ for sample-and-hold resampling and $SINAD_{filt,0} = 51.1\,\mathrm{dB}$ for the resampling filter. This shows that already in this simple case the resampling filter helps to reduce the noise caused by the quantization of the input signal $x(t)$ as well as the sampling instants $t_{\phi,l}$.

In another simulation a sine signal with a constant frequency of $f_{sine} = 1.25\,\mathrm{kHz}$ with the same amplitude as the chirp signal was added to the input signal. In this case the SINADs of the output signals were $SINAD_{SH,1} = 0.5\,\mathrm{dB}$ and $SINAD_{filt,1} = 51.2\,\mathrm{dB}$ for the sample-and-hold and filtered case respectively. Fig. 8 shows the results of this experiment. The frequencies of the signals that have been resampled uniformly over the angle are given in terms of orders. An order of m means that the signal has exactly m periods per revolution, i.e. over an angle of 2π. In the time frequency distribution of the input signal the chirp and constant frequency components are both clearly visible. The output from the resampling filter is a constant sinewave at the order two. This shows that the input chirp signal has successfully been resampled at uniform instants of the rotation angle, whereas the constant sinewave component has been completely eliminated. This is the expected behavior because its frequency is too high to be resampled correctly under the nyquist theorem:

$$f_{out,max} = 25 \cdot \frac{1500\,\mathrm{rpm}}{60} = 625\,\mathrm{Hz} < 2 f_{sine} \qquad (20)$$

Without the resampling filter the additive sine component has been aliased multiple times into the bands of the chirp signal resulting in a heavily noised signal.

V EVALUATION OF DESIGNED FILTER AND COMPARISON TO ALTERNATIVE METHODS

In the previous section it was shown that the designed filter is effective for synchronous resampling. Synchronous resampling is a well established technique in machine diagnosis [2] but new in the synchronous data acquisition with wireless sensor networks [1]. Its key advantage is that it poses nearly no additional requirements on other parts of the acquisition system like analog signal processing, analog-to-digital conversion and the communication protocol used in the network. It only relies on the availability of synchronized timestamps for the resampling instants. Any time synchronization protocol described in the literature could be used for this. Errors in the synchronization of timestamps will cause the resampling process not to resample to a truly uniform sampling grid. As a consequence, the output signal will be effected by additional noise that increases in strength with the size of synchronization errors. A detailed discussion of the effects of sampling time errors on the sampled signals can be found in [1].

In [10] various methods for nonuniform reconstruction are reviewed. Amongst those, low-pass filtering and polynomial interpolation seem to be the most widely used. Low-pass filtering (used e.g in [5]) reconstructs a continuous signal out of the nonuniform samples by low-pass filtering a modulated delta comb. However, as can be seen from equation 6, it does not remove the multiplicative noise from the signal [4].

Another method frequently used is polynomial interpolation [10]. It is computationally very simple, especially for low polynomial orders. Yet, at low polynomial orders the usable signal bandwidth and alias attenuation are both very low [11]. At higher polynomial orders the computational complexity quickly increases [10]. Thus, polynomial interpolation is restricted to be used with highly oversampled narrow-band signals. The sample-and-hold resampling that was examined in section IV.B can be referred to as polynomial resampling with a polynomial order of zero. Higher polynomial orders would not have improved performance in this case because the resolution of the timestamps $\Delta t \approx 61\,\mu\mathrm{s}$ was bigger than the sampling period $T_{in} = 50\,\mu\mathrm{s}$.

In [2] two methods for angular resampling are presented. The first uses a FIR interpolation filter to upsample the input signal by a constant integer factor. The upsampled signal is then resampled to the correct angular instants using linear interpolation. Finally, the resampled signal is downsampled using an FIR decimator. The FIR interpolation filter ensures that the signal is sufficiently band-limited to be safely used with the linear interpolator. This method is effective for angular resampling if the supported rpm range is small. At higher rpm-ranges it requires increasingly narrow-band and high order FIR filters leading to a tremendous computational complexity.

The second method presented in [2] applies the hybrid model for resampling (compare Fig. 2) to a continuous filter whose impulse response is a truncated sinc-function. The values of this impulse response are stored in a look-up table which can be of considerable size ($\approx 8192\,\mathrm{Elements}$). Again this method is quite efficient for low rpm-ranges. But at high rpm-ranges the required narrow filter bandwidth resulting in a wide impulse response makes it computationally infeasible, as well.

VI CONCLUSIONS AND OUTLOOK

The designed resampling filter is with only little modifications capable of effectively resampling jittered input samples as well as doing angular resampling. Its com-

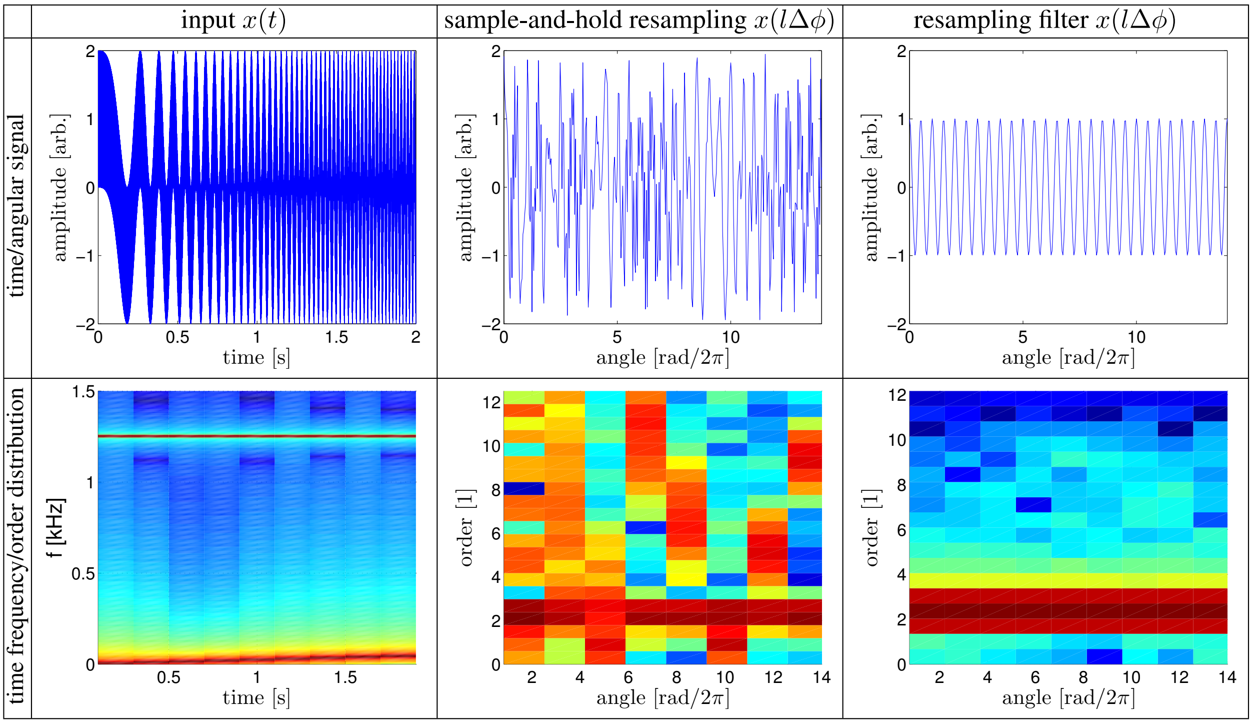

Fig. 8. Results of angular resampling experiment

putational requirements are sufficiently low to be implemented on a commercial wireless sensor node. When the rpm-range of a rotating machine is high, it surpasses other methods in terms of output signal quality and computational efficiency.

Future research will include the implementation of the filter on a wireless sensor node and its testing at a machine test bench. The effects of round-off errors that occur in this implementation will be investigated.

REFERENCES

[1] J. Funck and C. Gühmann, “Comparison of approaches to time-synchronous sampling in wireless sensor networks,” *Measurement*, vol. 56, no. 0, pp. 203 – 214, 2014.

[2] J. Blough, “Adaptive resampling - transforming from the time to the angle domain,” in *24th Conference and Exposition on Structural Dynamics 2006, IMAC-XXIV*, 2006.

[3] F. Marvasti, *Nonuniform sampling: theory and practice*. No. Bd. 1 in Information technology–transmission, processing, and storage, Kluwer Academic/Plenum Publishers, 2001.

[4] V. Lehtinen, E. Maki-Esko, M. Valkama, and M. Renfors, “Error spectrum analysis of time-variant reconstruction from nonuniform sampling,” in *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*, pp. 78–81, 2006.

[5] D. Babic and M. Renfors, “Reconstruction of non-uniformly sampled signal using transposed farrow structure,” in *Circuits and Systems, 2004. ISCAS ’04. Proceedings of the 2004 International Symposium on*, vol. 3, pp. III–221–4 Vol.3, 2004.

[6] J. Vesma and T. Saramaki, “Interpolation filters with arbitrary frequency response for all-digital receivers,” in *Circuits and Systems, 1996. ISCAS ’96., Connecting the World., 1996 IEEE International Symposium on*, vol. 2, pp. 568–571 vol.2, 1996.

[7] O. Peters, “Diagnose von Unwuchten mit einem drahtlosen intelligenten Körperschallsensor,” Diplomarbeit, Technische Universität Berlin, 2013.

[8] F. M. Klingler and H. G. Göckler, “Conversion between arbitrary sampling rates: An implementation cost trade-off study for the family of farrow structures,” in *Proc. 5th Karlsruhe Workshop on Software Radios*, pp. 1–8, 2008.

[9] H. G. Glöckler and A. Groth, *Multiratensysteme*. J. Schlembach Fachverlag, 2004.

[10] Y.-R. Sun and S. Signell, “Algorithms for nonuniform bandpass sampling in radio receiver,” in *Circuits and Systems, 2003. ISCAS ’03. Proceedings of the 2003 International Symposium on*, vol. 1, pp. I–1–I–4 vol.1, 2003.

[11] P. McFadden, “Interpolation techniques for time domain averaging of gear vibration,” *Mechanical Systems and Signal Processing*, vol. 3, no. 1, pp. 87 – 97, 1989.