A Calibration Scheme for a Nonuniform Sampling Driver Architecture

F. Papenfuß
University of Rostock
Richard Wagner Str. 31
18119 Rostock, Germany
+49 (0)381 498 3536
frank.papenfuss@uni-rostock.de

F. Olbricht
University of Rostock
Richard Wagner Str. 31
18119 Rostock, Germany
+49 (0)381 498 3537
frank.olbricht@uni-rostock.de

Y. Artyukh
Institute of Electronics and CS
14 Dzerbenes Str.
LV 1006 Riga, Latvia
+371 755 1771
artyukh@edi.lv

D. Timmermann, University of Rostock, Richard Wagner Str. 13, 18119 Rostock, Germany,
+49 (0)381 498 3528, **dirk.timmermann@uni-rostock.de**

Abstract

In this paper the authors propose a calibration scheme for a nonuniform sampling driver (SD) architecture introduced in [10]. The employed sampling scheme of the SD is matched to the ADCs analogue bandwidth. The bandwidth of our nonuniform sampling system exceeds 16 times the limit set by the classical sampling theorem (bandwidth is 800 MHz instead of 50 MHz at a mean sampling rate of 66 MHz). This is achieved by deliberate randomisation of the sampling points on a fine time grid having 625 ps spacing. Delay steps are realised using a digitally controllable delay line (DCDL). Compared to that a traditional system operates the same ADC at its maximum conversion rate of 100 MSPS.

The paper demonstrates the application of additive random sampling (ARS) to wideband signals. It is shown how the noise floor of spectral analysis can be reduced through sequential component extraction. Results are presented that show how the accuracy of time instance determination enhances the component extraction process.

The presented calibration scheme enables high quality spectral analysis of wideband signals. This in turn is very vital property for subsystems like the environment characterisation function in a software defined radio environment (see [9]).

1. Introduction

A sampling driver is a device connected directly to the encode input of an ADC. In contrast to traditional sampling systems applying a fixed frequency clock to the ADC encode input our approach applies a randomised train of sampling pulses to the ADC. Assuming that the SD system clock T_{clk} is matched to the maximum ADC conversion rate the bandwidth gain is given by

$$G_{BW} = M = \frac{T_{clk}}{T_Q}. \tag{1}$$

Where T_Q is the time quantum realised using a DCDL. T_Q is the least significant delay step (see [10] for details). Earlier publications [10,11] showed efficient architectures to construct a suitable sampling scheme $x_s(t)$, given in equation (2), when taking K nonuniform samples. The statistical properties (i. e. pdf of sampling instances and intersample intervals) of a randomised sampling scheme must be carefully chosen.

$$x_s(t) = \sum_{k=0}^{K-1} \delta(t - t_k). \tag{2}$$

The sampling instances are placed on a grid with spacing T_Q. The main goal is to generate a suitable sampling point density function (SPDF). The SPDF indicates the probability that a particular moment in time will become a sampling instance. A suitably constructed random sampling scheme will produce a SPDF having a constant value given by T_Q/T_{mean} (where T_{mean} is the mean sampling period). For a more complete coverage of nonuniform sampling and the details on how to construct a suitable nonuniform sampling scheme for alias suppression see [3-7,12]. One possible sampling scheme producing a constant SPDF is additive random sampling (ARS) explained in [4,13]. It is implemented by the architecture referred to in this paper.

In a real system an ARS scheme will be applied to a signal a limited amount of discrete spectral components. The spectrum of the signal will convolute itself with the spectrum of the randomised sampling process. This results in a high noise floor of around 25dB to 30dB (depending on K) relative to the component carrying the most power. This is a well-known, unavoidable property of the nonuniform sampling approach.

In order to reveal components buried in noise identified components can be subtracted in time domain from the original signal. The difference is fed back to the nonuniform spectral analysis leading to a recursive procedure proposed in [8]. To be able to subtract identified spectral components from the time discrete signal $x_d(t_k)$:

$$\begin{aligned} x_d(t) &= x(t)x_s(t) = x(t)\sum_{k=0}^{K-1}\delta(t-t_k) \\ &= \sum_{k=0}^{K-1} x(t_k)\delta(t-t_k) \end{aligned} \tag{3}$$

The sampling time instants $\{t_k\}$ have to be known a priori as precise as possible. The precision of estimation of the sampling time instants will have a vital effect on the quality with which the sequential component extraction algorithm will converge. This will be shown in the paper.

2. Spectral Analysis

In this Section simulation results are presented. To obtain a near to realistic scenario we define a signal with ten discrete spectral components which we pretend not to know. Properties of the components are given in Table 1:

Nr.	*Amplitude [%FS]*	*Frequency [MHz]*	*Level [dB]*	*Phase [rad]*
1	3.1	2.4	-18	0.0
2	1.6	10.0	-24	3.142
3	12.5	43.9	-6	5.027
4	25.0	65.9	0	1.571
5	6.3	165.9	-12	4.712
6	25.0	232.9	0	0.031
7	6.3	314.9	-12	3.770
8	0.4	432.9	-36	2.827
9	6.3	500.0	-12	0.283
10	25.0	665.9	0	2.199

Table 1: ADC test signal constituents

Obviously, when samples are taken nonuniformly, the classical FFT algorithm cannot be applied. However, as Figure 1 illustrates, samples are placed on a regular grid. This regular grid has spacing T_Q.

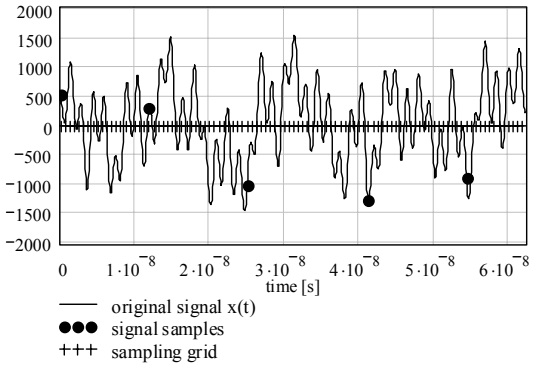

Figure 1: Continuous signal and its samples.

The ARS scheme used by the SD described in [11] is given by:

$$t_k = t_{k-1} - \varepsilon_{k-1}T_Q + T_{clk} + \left\{ \begin{matrix} 0 & if \quad \varepsilon_{k-1} < \frac{M}{2} \\ T_{clk} & otherwise \end{matrix} \right\} + \varepsilon_k T_Q \,. \qquad (4)$$

$$k, \varepsilon_k \in \mathbb{N}; \quad 0 \le \varepsilon_k < M$$

Where ε_k represents the k-th pseudo random number (maximum length LFSR). Thus, the theoretical sampling time instants $\{t_k\}$ are determined. They can be calculated a priori for a particular sampling run given the seed ε_0 and the recursive production formula (4):

$$t_k = e_k + \begin{cases} 0 & if \quad k = 0 \\ \sum_{l=1}^{k} \Delta t_l & otherwise \end{cases}$$

$$where \quad \sum_{l=1}^{k} \Delta t_l = T_Q \sum_{l=1}^{k} \Delta n_l = T_Q n_k \qquad . \qquad (5)$$

$$\Delta n_l = M - \varepsilon_{l-1} + \left\{ \begin{matrix} 0 & if \quad \varepsilon_{l-1} < \frac{M}{2} \\ M & otherwise \end{matrix} \right\} + \varepsilon_l$$

Where n_k is the k-th index into the sampling grid ($n_0 = 0$). The value e_k is a model of the error when taking the k-th sample. The calculation of the spectrum follows as:

$$\begin{aligned} X(f) &= \sum_{k=0}^{K-1} x(t_k) \exp^{-j2\pi f t_k} \\ &= \sum_{k=0}^{K-1} x(n_k T_Q + e_k) \exp^{-j2\pi f (n_k T_Q + e_k)} \end{aligned} \qquad . \qquad (6)$$

From formula (6) it is clear that the FFT algorithm can be applied to the sampling set given that the unoccupied places of the sampling grid are set to zero (as opposed to zero padding we call this technique "zero stuffing").

$$x_{FFT}(mT_Q) = \begin{cases} 0 & if \quad m \notin \{n_k\} \\ x(n_k T_Q + e_k) & otherwise \end{cases} \qquad (7)$$

$$m, n_k, k \in \mathbb{Z}; \quad 0 \le k < K$$

Note that this means that the unavoidable sampling time errors are ignored for FFT processing. Usually these errors increase the noise floor however they are assumed to be an order of magnitude lower than the time quantum and are neglected.

As can be seen in Figure 3 the FFT is already very early in favour compared to direct computation of the DFT when calculating $(T_{mean}/T_Q)K$ different equally spaced frequency values ranging from DC to $f_Q = 1/T_Q$ (where $1/(2\ T_Q)$ is the equivalent bandwidth of the system).

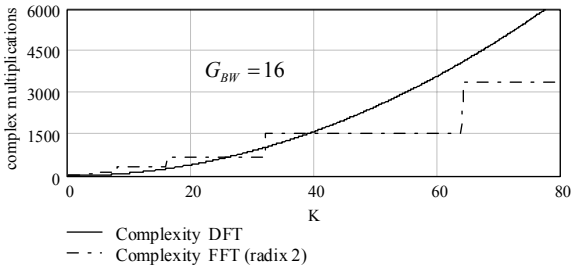

Figure 2: Complexity DFT vs. zero filled FFT

For the calculations presented in this paper a number of K=680 (N_{FFT}=16384) nonuniform samples were taken. This provides a balance between a sufficient number of

samples (information) taken from the signal and the effort to perform spectral analysis. The resulting signal spectrum characterised by Table 1 is depicted in Figure 3. Only components 3, 4, 6 and 10 are clearly identified. The remaining components are buried in noise. Therefore, the effective spectral dynamic range is around 10 dB. According to theory [13] noise suppression for real valued AR sampled signals can be approximated by

$$NS_{ARS} = 10\log_{10}\left(\frac{2}{K}\right). \qquad (8)$$

In case of our system (8) delivers a noise figure of -25 dB (relative to most powerful component). This theoretical value has to be interpreted as a mean value of the noise observed in Figure 3. Deviation from the theoretical value delivered by (8) in the zero stuffed FFT result destructs from the achievable dynamic range and thus the effective dynamic range is diminished to the afore mentioned 10 dB.

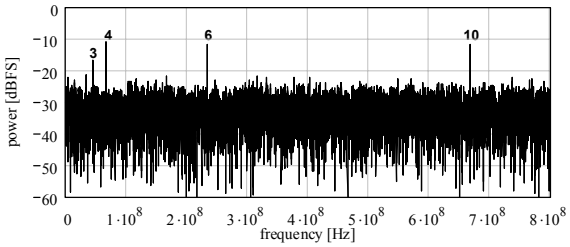

Figure 3: First iteration spectrum (simulation)

A value clearly not sufficient to make this technique attractive to RF environments where a dynamic range of 50 or even 60 dB is more desirable.

Luckily there is a remedy to this problem. Assuming that the sampled signal is band limited in nature and constitutes from a finite number of frequencies one can extract the most powerful component and subtract its time representative $c_i(t)$ from the original signal at the nonuniform sampling instants. The resulting difference at iteration step $i+1$ is a signal carrying less power than the signal at iteration step i. Thus, the noise floor of the zero stuffed FFT will be lowered with respect to the previous iteration step when $x_{d,i+1}$ is transformed. The algorithm is sketched in (9).

This is a recursive algorithm with no defined termination. A termination can be constructed using a set of conditions leading to termination if either condition is met:

- A fixed number of most powerful components (at fixed frequencies or in fixed frequency bands) has been found
- The noise floor has dropped to a desired level (determined by statistical means)
- A fixed limit of iterations has been reached.

Applying 40 iterations of sequential component extraction to our test signal described in Table 1 we obtain the spectrum depicted in Figure 4.

$$1.)\quad X_i(f) = \sum_{m=0}^{N_{FFT}-1} x_{FFT,i}\left(mT_Q\right)\exp^{-j2\pi fmT_Q}$$

$$2.)\quad n_{\max,i} = \max_n\left\{\left|X_i\left(2\pi\frac{nf_Q}{N_{FFT}}\right)\right|^2\right\}$$

$$n \in \mathbb{Z} \quad 0 \le n < N_{FFT} \qquad . (9)$$

$$3.)\quad c_i(t) = F^{-1}\left\{X_i\left(2\pi\frac{n_{\max,i}f_Q}{N_{FFT}}\right)\right\}$$

$$4.)\quad x_{d,i+1}(t) = \sum_{k=0}^{K-1}(x_i(t_k) - c_i(t_k))\delta(t-t_k)$$

The result is quite impressive. However it has been obtained under idealised simulation conditions:

- No quantisation
- No modulation effects in any of the observed frequencies
- Signal was sampled using typical delay steps of the DCDL chip (data sheet). Thus the deviations from the ideal sampling scheme were known and considered in step 4.) of algorithm (9).

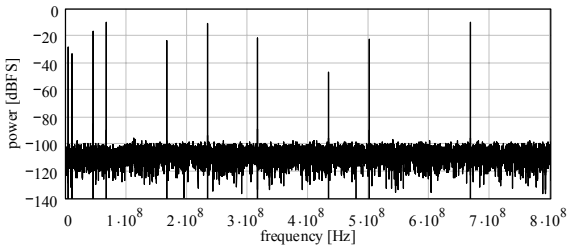

Figure 4: Spectrum after 40 iterations (simulation)

To demonstrate the effect of not correctly determining the DCDL delay steps Figure 5 is presented here. This simulation result was obtained by sampling a mono frequency (component six of Table 1) signal using typical DCDL delays (obtained from [1]). Then signal analysis took place assuming that sampling points were taken with no errors (without deviations from the ideal scheme (5)).

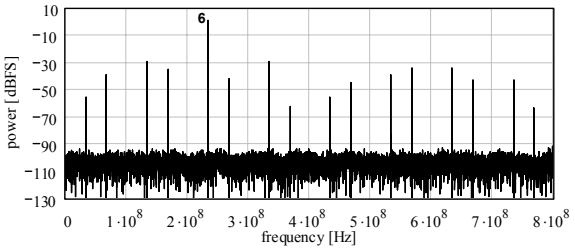

Figure 5: Analysed 233 MHz signal (simulation)

Note that these are not alias frequencies. These spurious spectral components stem from the fact that during the component subtraction step in (9) the component $c_i(t)$ was in fact sampled at time instants t_k. carrying a different error then the signal $x_{d,i}(t)$. Errors statistics are accessible because the whole process is simulated. They are depicted

in Figure 6. It is easily observed that deviations are limited to fixed regions creating the effect observed in Figure 5. In a real measurement the error is hidden and can only be approximated through a calibration procedure. It will be more random, which will weaken spurious frequency content but add to noise floor.

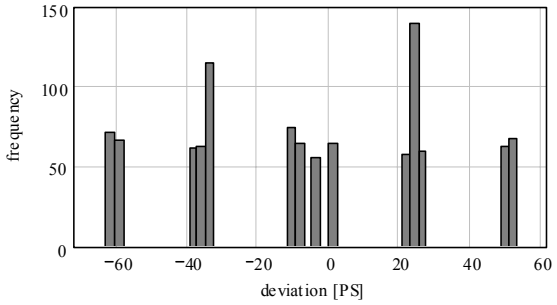

Figure 6: Error statistic for typical delay steps

Simulations show that this is the most error sensitive part of the component extraction algorithm. Errors in time instant estimation will lead to spurious frequency content in subsequent spectra. Therefore, it is easily concluded that the true delay steps of the DCDL must be determined as precise as possible. The next Section will propose a procedure to achieved such measures.

3. Calibration Algorithm

The sampling scheme used for this calibration algorithm is similar to uniform sampling. It determines every delay by taking advantage of a sinusoid calibration signal with system clock frequency f_{clk}, which is easily described analytically:

$$x_c(t) = A_c \cos(2\pi f_{clk} t + \varphi_c) . \qquad (10)$$

Our basic idea is to match sampling and calibration signal frequency by deriving both from the system frequency f_{clk}. This results in the same sample value for every sample. A change in the SD delay line just changes the sampling instant phase, thus, delivering a different sampling value specific to a particular delay step under investigation. Base delay is defined by pulling low all delay line tap bits. Then the base sample x_{base} is taken. During the next system clock cycle the delay step tap bits are activated and the delay step specific calibration sample x_{cal} is taken as shown in Figure 7. Due to the constant phase φ_c of the calibration signal base sample and calibration sample can be mapped into the same period of the calibration signal. To illustrate this Figure 8 shows two sample pairs mapped into the same period.

Using the inverse mathematical function of the calibration signal, the delay between base sample x_{base} and calibration sample x_{cal} is easily estimated. The estimation formula is given by:

$$\Delta\hat{t} = \frac{\arccos\left(\frac{x_{cal}}{\hat{A}_c}\right) - \arccos\left(\frac{x_{base}}{\hat{A}_c}\right)}{2\pi f_{clk}} \qquad (11)$$

To obtain reliable estimates statistical analysis has to be applied to a set of tuples $\{x_{base}; x_{cal}\}$ using the above scheme. The mean value will be an estimate of the true delay of the inspected delay. The standard deviation will quantify the random measurement error.

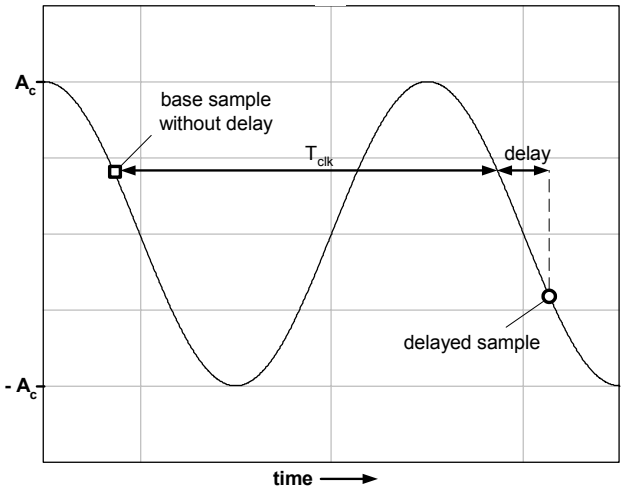

Figure 7: Alternating delay steps between two consecutive sampling pulses

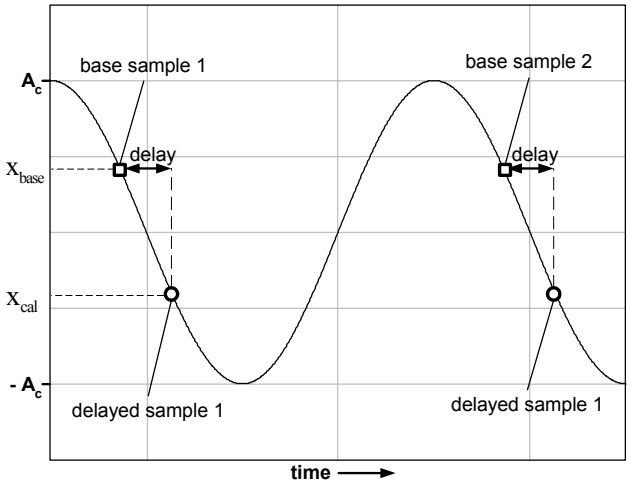

Figure 8: Mapping both, base and calibration sample into same period of calibration signal

Accuracy is improved when sampling both, base sample as well as calibration sample, at time instances where the sine wave has a steep slope. This can be achieved by shifting the phase of the calibration signal before performing the calibration measurement for a particular delay step. It is easily observed from (11) that knowledge of the calibration signal amplitude A_c is essential for calculation. A_c is obtained by shifting the phase of the calibration

signal while sampling in uniform mode. The maximum is kept for further calculations as an estimate of A_c.

In order to get a sine wave with low jitter, the main clock source of the system comes in quite handy. In our case it is a stable clock of approximately 1 ps RMS jitter. A common clock source also guaranties that frequency drift, always present in real systems, between signal and sampling frequency is synchronous! The sine wave calibration signal is extracted by passing the delayed clock signal through a low-pass filter (see Figure 9). The used delay chip inside the calibration module realises phase adjustment of the calibration signal. The complete structure of the SD with calibration capability is depicted in Figure 9.

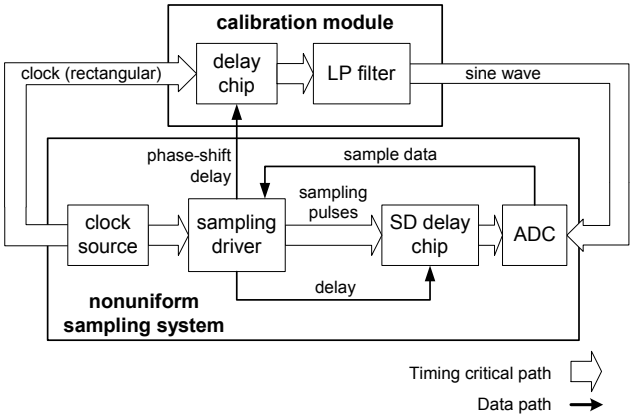

Figure 9: Basic structure of sampling system with calibration module

4. Calibration and Measurement Results

To obtain meaningful statistics we gather around 8000 tuples per delay step of the SD delay chip after warm-up of the board. As representative results measurements for delay step one and two are given in Figure 10 and Figure 11, respectively.

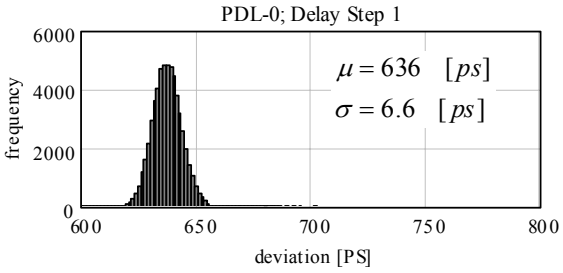

Figure 10: SD delay chip, delay step one statistics

The actual delay step values are fed into the simulation engine that is thus prepared to process real nonuniform sampled data from our nonuniform sampling digitiser hardware.

Figure 12 to Figure 14 show representative nonuniform sampling measurement results after 40 iterations of sequential component extraction for input frequencies of 25, 100 and 350 MHz, respectively.

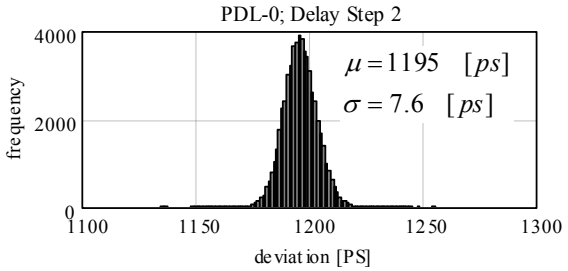

Figure 11: SD delay chip, delay step two statistics

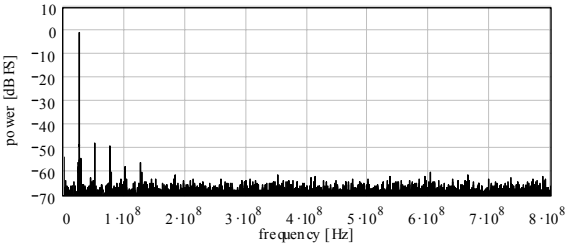

Figure 12: Measurement 25 MHz (R&S SWP)

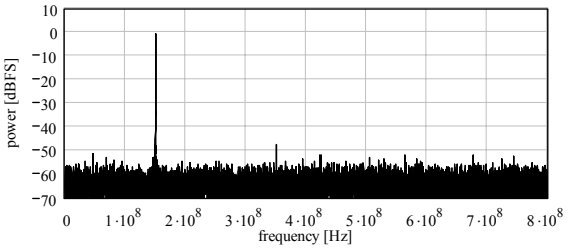

Figure 13: Measurement 150 MHz

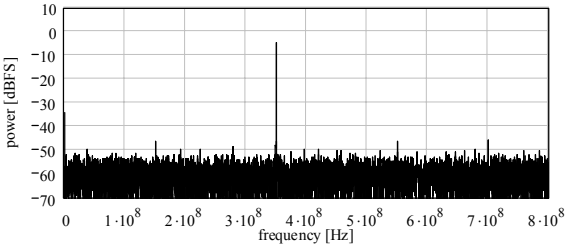

Figure 14: Measurement 350 MHz

Our system has a mean sampling rate T_{mean} of 66 MHz. The 25 MHz measurement falls well below the Nyquist frequency, considering the fact that the used ADC (AD9432) can be operated uniformly at a sampling rate of 100 MSPS. Clearly observed are the harmonics of the used R&S generator at around 50 dBc. We place Figure 12 as a reference for the other spectra shown here. As the input signal frequency increases spurious frequencies are observed. The source of these spurious frequencies has been discusses thoroughly in Section 2. Here, however, the effect is less disastrous mainly because the quality of

actual delay estimation is high. A spectral dynamic range of 45 dB is maintained throughout the analogue bandwidth of our digitiser board.

5. Conclusions

If nonuniform sampling techniques are to make their way into mainstream designs then one utterly important component is the calibration module. This module will determine the true delay steps by statistical means. The acquired delay step information is used to calculate derivations from ideal sampling points. It is necessary for successful application of sequential component extraction to process nonuniform sampling data. The zero stuffed FFT is a robust transform with regard to the delay step errors of the delay line. However, when calculating the signal difference in (9) step 4.) delay step errors have to be known with a precision at least an order of magnitude less then T_Q. Otherwise spurious artefacts will appear in the spectrum. It is this restriction that that limits the realisable bandwidth of nonuniform sampling systems along, of course with the analogue bandwidth of the ADC.

We propose a calibration scheme that we successfully use to determine the true delay steps in our prototype system. The results obtained with the real system in terms of spurious spectral artefacts are quite promising comparing them to other published designs (e. g. [2]). However, the application of our hardware to more complex signals containing more then one carrier remains to be researched. Simulation and measurement results presented in Section 2 and 4, respectively, indicate that a spectral dynamic range of at least 45 dB is realistic. This is a usable figure for RF processing systems.

The main benefit of nonuniform sampling designs for spectral analysis is mainly where a direct digital processing of a wide band (500 to 750 MHz) is desirable. This is the case for instance in software defined radio (SDR) systems. SDR systems require a central RF environment characterisation unit coupled to a control unit that steers a couple of narrow band units operating at selected bands <10 MHz (see [9]).

Obviously a nonuniform sampling system must be capable of real-time processing. What is real-time has to be defined in the context of the concrete application. Our system is not a RT system solution since the data is currently processed offline in a PC environment. However, RT capability can be achieved using state of the art DSP or FPGA solutions.

6. References

[1] MC10EP195, MC100EP195 - 3.3V ECL Programmable Delay Chip. OnSemiconductor . 2003.

[2] Artyukh, Y., Rybakov, A., and Vedin, V., "Evaluation of Pseudo-random Sampling Processes," *Sampling Theory and Applications, 1997. SAMPTA '97., Proceedings of the International Workshop on,* Aveiro, Portugal, pp. 361-363.

[3] I. Bilinskis and I. Mednieks. *Introduction to digital alias-free signal processing*, Riga: Institute of Electronics and Computer Science, 2001.

[4] I. Bilinskis and A. Mikelsons. *Randomized Signal Processing*, Prentice Hall International, 1992.

[5] Bland, D. M. and Tarczynski, A., " Optimum nonuniform sampling sequence for alias frequency suppression," *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on,* Hong Kong, pp. 2693-2696 vol.4, June 1997.

[6] Marvasti, F. *Nonuniform sampling, theory and practice*, New York: Kluwer Academic, 2001.

[7] Mednieks, I. , "Methods for spectral analysis of nonuniformly sampled signals," *Sampling Theory and Applications, 1999. SAMPTA '99., Proceedings of the International Workshop on,* Loen, Norway, pp. 190-193.

[8] Mikelsons, A., "Alias-free spectral estimation of signals with components of arbitrary frequencies," *COST 229 Workshop,* Ljubljana, Slovenia, pp. 105-108, Apr. 1994.

[9] Mitola, J., The software radio architecture *Communications Magazine, IEEE*, vol. 33, no. 5, pp. 26-38, 1995. 0163-6804.

[10] Papenfuss, F., Artyukh, Y., Boole, E., and Timmermann, D., "Nonuniform sampling driver design for optimal ADC utilization," *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on,* Bangkok, Thailand, pp. IV-516, IV-519 vol.4, May 2003.

[11] Papenfuss, F., Hecht, R., and Timmermann, D., "Scalable VHDL architectures for non-uniform sampling driver design," *Proceedings of the 8th Biennial Baltic Electronic Conference, BEC2002,* Tallinn, Estonia, pp. 249-252, 2002.

[12] Tarczynski, A., Bland, D. M., and Laakso, T. I., "Spectrum estimation of non-uniformly sampled signals," *Industrial Electronics, 1996. ISIE '96., Proceedings of the IEEE International Symposium on,* Warsaw, Poland, pp. 196-200 vol.1, June 1996.

[13] Wojtiuk, J. J., Randomised sampling for radio design 2000. Univ. of South Australia. http://www.itr.unisa.edu.au/~steven/thesis/jjw2.ps.gz.