ADC Clock Jitter Requirements for Software Radio Receivers

Vincent J. Arkesteijn, Eric A.M. Klumperink and Bram Nauta
IC-Design group, MESA$^+$ Research Institute, University of Twente, the Netherlands
http://icd.el.utwente.nl email: v.j.arkesteijn@utwente.nl

***Abstract*— The effective number of bits of an analogue-to-digital converter (ADC) is not only limited by the quantisation step inaccuracy but also by sampling time uncertainty. According to a commonly used model, timing jitter errors should not introduce a sampling error bigger than 1 quantisation level for full swing input signals at a frequency equal to half the sample rate. This results in unfeasible phase noise requirements for the sampling clock in software radio receivers with direct RF sampling as in figure 1. This paper explores the clock jitter requirements for a software radio application, using a more realistic model found in the literature and taking into account the power spectrum of both the input signal and the spectrum of the sampling clock jitter. Using this model, we show that the clock jitter is *not* the limiting factor in the feasibility of software radio receivers.**

I. Introduction

As opposed to a conventional single-standard receiver, a software radio is designed to receive signals from multiple standards. This flexibility is achieved by performing most of the signal processing in software. Because software runs on digital hardware and radio waves are analogue by nature, an analogue-to-digital converter (ADC) has to be included. An example of a software radio front-end is shown in figure 1. We are researching various software radio receiver architectures, in cooperation with colleagues from our university's signals and systems group [1].

Various radio standards use different portions of the radio spectrum. Enabling reception of signals from various radio standards thus necessitates a large RF input bandwidth, and because of this, a high sampling rate.

In this large bandwidth, many signals will exist, some weak, some potentially very strong. The resolution of an ADC is determined by the difference in power levels between the strongest signal present at the input and the quantisation noise level. Because weak signals have to be converted in the presence of very strong ones, the required ADC resolution is also very high.

ADCs combining the bandwidth and resolution required for software radios capable of receiving modern wireless standards are at present unfeasible and/or consume too much power. Nevertheless, it is interesting to analyze their clock jitter requirements, to show that clock jitter is not an obstacle for these ADCs.

As will be shown in secion II-A, according to a commonly used model for white ADC clock jitter [2], the resolution directly affects the clock jitter requirements, resulting in very stringent numbers. Better jitter models are available in the literature, e.g. [3], [4], [5]. Section II-B will show such a model. In this paper, we show that using this more realistic model, taking into account the spectra of both the input signals and the sampling clock jitter, sampling clock jitter requirements can be relaxed. Section III shows an example of a software radio receiver, where numeric results signify the relevance of the more realistic ADC model. Finally, section IV will present some conclusions.

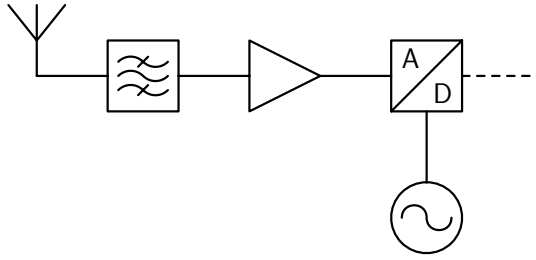

Fig. 1. A Software Radio Receiver Front-End

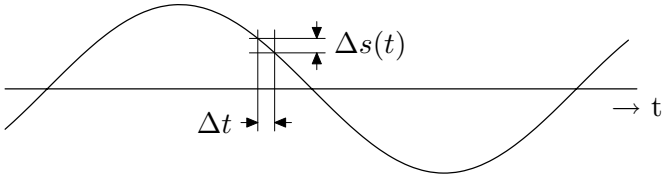

Fig. 2. An input signal as a function of time and the effect of sampling jitter

II. Jitter effects on sampling and mixing

In this section, effects of jitter in the sampling clock will be analysed. First, a commonly used model will be presented. This is followed by a more precise model.

A. ADC – White-noise model

Consider an incoming signal $s(t)$. Ideally, the sampled version of this signal with sample rate $1/\tau$ $s_\tau(k)$ is constructed as follows.

$$s_\tau(k) = s(k\tau) \tag{1}$$

Due to sampling jitter however, an error will be introduced, as can be seen in figure 2. The sampled signal can now be calculated as follows (for small Δt).

$$s_\tau(k) \quad \approx \quad s(k\tau) + \Delta t(k\tau) \cdot \frac{\partial}{\partial t} s(t)\Big|_{k\tau} \tag{2}$$

This signal consists of a sampled version of the input signal $s(t)$ plus an error signal. This error is generally assumed to have a white spectrum [2]. If this is the case, and assuming a full swing harmonic input signal at the maximum input frequency BW, the following relation between required RMS jitter and resolution can be derived.

$$\Delta t_{rms} = \frac{1}{2\pi BW 2^n}\sqrt{\frac{2}{3}M} \tag{3}$$

Here n is the resolution of the ADC and M is the oversampling ratio [6].

Using this equation, the required RMS jitter can be calculated for a given resolution. For software radio applications this yields clock

0-7803-8521-7/04/$20.00 © 2004 IEEE

jitter requirements that are not achievable with currently available (integrated) clock sources. Our calculations indicate that for instance a software radio capable of implementing a Hiperlan/2 receiver would require roughly 11 fs RMS jitter for the ADC sampling clock.

B. ADC – More realistic model

Equation 3 is based on the assumption that the error signal that results from sampling jitter is white. This would be a valid assumption, when at least one of $\Delta t(t)$ and $\frac{\partial}{\partial t}s(t)$ in equation 2 were white. However, in software radios both are usually non-white, as we show below.

The input spectrum of a receiver is not known in general, but wireless communication standards normally specify the level of interfering signals that have to be tolerated. Figure 3 for example shows the blocking levels for Hiperlan/2 [7]. Interfering signals that are close in frequency to the wanted signal have far lower power levels than those further away. Thus, $s(t)$ is not white, and neither is $\frac{\partial}{\partial t}s(t)$.

Furthermore, $\Delta t(t)$ usually is not white either. When the sampling clock is derived from a synthesizer containing an LC oscillator, $\Delta t(t)$ can be assumed to have a f^{-2} power spectrum outside the synthesizer loop bandwidth [8].

Because both the input signal and the jitter are non-white, the error signal is in general also non-white. As seen in equation 2, the error signal $\Delta s_\tau(k)$ is the time derivative of the input signal $\frac{\partial}{\partial t}s(t)$ multiplied with the sampling time error $\Delta t(k\tau)$:

$$\begin{aligned} \Delta s_\tau(k) &= s(k\tau + \Delta t(k\tau)) - s(k\tau) \\ &\approx \Delta t(k\tau) \cdot \frac{\partial}{\partial t}s(t)\Big|_{k\tau} \end{aligned} \tag{4}$$

Taking the discrete-time Fourier transform (DTFT) of both sides:

$$\mathcal{F}(\Delta s_\tau(k)) \approx \mathcal{F}\left(\Delta t(k\tau)\right) \star \mathcal{F}\left(\frac{\partial}{\partial t}s(t)\Big|_{k\tau}\right) \tag{5}$$

where $\mathcal{F}$ denotes the DTFT and $\star$ denotes convolution. This is result is also obtained in [3], [9]. For a harmonic input signal $s(t) = A\sin(\omega_{in}t)$ this becomes:

$$\mathcal{F}(\Delta s_\tau(k)) \approx \mathcal{F}\left(\Delta t(k\tau)\right) \star \omega_{in}A \cdot \mathcal{F}\left(\cos(\omega_{in}k\tau)\right) \tag{6}$$

Due to its f^{-2} nature, most energy in $\Delta t(k\tau)$ is at low frequencies. Knowing that in the frequency domain this is convoluted with the derivative of the input signal leads to the following.

1) The convolution operation in equation 5 shifts the jitter spectrum $\mathcal{F}(\Delta t(k\tau))$ to the frequencies of input signals. Therefore, the jitter-induced error in the output is concentrated around these frequencies.
2) Input signals with higher power are surrounded by more jitter-induced error in the output than input signals with lower power, due to the linearity of the convolution operation.
3) Input signals of higher frequencies are surrounded by more jitter-induced error in the output than signals at lower frequencies, because of the frequency dependent effect of $\frac{\delta}{\delta t}s(t)$. This is in accordance with the results in [5].

Because the jitter-induced output error is not white, but concentrated around the frequencies with the strongest input signals, it is less of a problem in the frequency band of interest.

III. Example

To illustrate the difference between the two ADC models, a numeric example will be shown. A software radio receiver capable of receiving different WLAN standards has been chosen for this. This receiver will be used for reception of Hiperlan/2 signals.

The RF filter at the input (see figure 1) will pass signals from around 2 GHz to around 6 GHz, since most WLAN standards are in either the 2.4 or 5.5 GHz bands. We will show the required sampling clock jitter first as estimated using the white-noise model, and then as estimated using the second model.

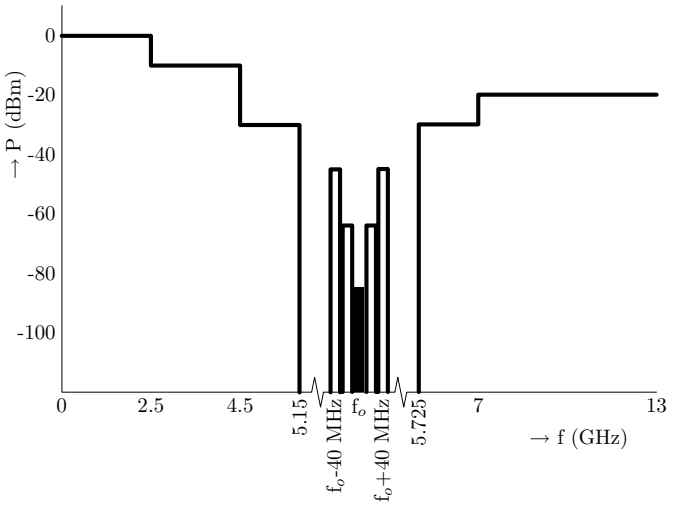

Fig. 3. In-band and out-of-band blocking levels for HiPerLAN/2 [7], together with the level of the wanted signal (solid bar at f_0) during blocking tests. Note: frequency axis is not entirely to scale.

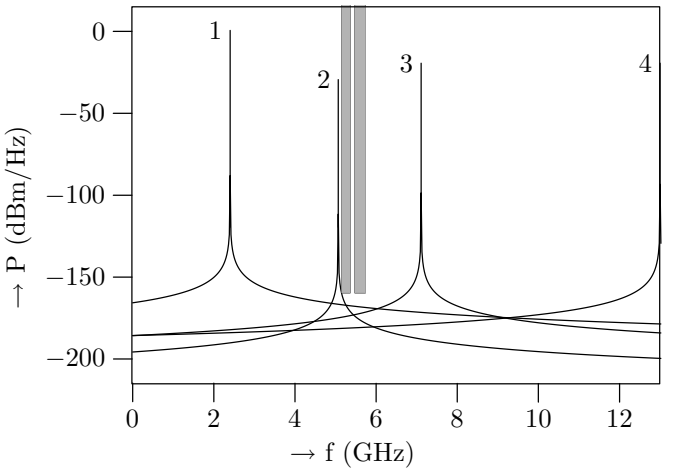

Fig. 4. ADC output for four demanding interfering input signals (according to the Hiperlan/2 specification). Acceptable reception requires the in-band jitter-induced output noise to be below the shaded area. This area is delimited by the band limits and the maximum acceptable in-band noise level for both Hiperlan/2 bands.

In order to meet Hiperlan/2 requirements, the resolution according to the first model can be calculated as follows. The largest signal level at the input at which the desired signal still has to be demodulated is 0 dBm according to [7] (see figure 3). The noise level that is still acceptable to the demodulator is 10 dB above thermal noise, so -164 dBm/Hz or -68 dBm in a 4 GHz bandwidth. If we allow all the noise in the receiver to be caused by quantisation in the ADC, this requires a resolution of 11 bits.

Using these numbers in equation 3 results in

$$\Delta t_{rms} = \frac{1}{2\pi \cdot 6G \cdot 2^{11}}\sqrt{\frac{2}{3}} \approx 11 \text{ [fs]}$$

This is one or two orders of magnitude smaller than what is achieved by currently published integrated synthesizers.

If we use the more realistic ADC model however, results are different. Figure 4 shows the output spectrum of an ADC, with four different input signals. The levels of these signals were taken to be the blocking levels as shown in figure 3, at the frequencies where their impact is most severe (2.4, 5.06, 7.1 and 12.98 GHz).

It is clear from figure 4 that the strongest signal (0 dBm at 2.4 GHz), which was a key factor in the first model, has no effect in the second model. Because of this, requirements are more relaxed.

The RMS jitter of the sampling clock used for figure 4 was 5 ps, with a flat power spectrum up to 100 kHz from the carrier and a f^{-2} roll-off above that. This shows that the more realistic model yields far more feasible requirements than the first model (5 ps RMS jitter instead of 11 fs).

IV. Conclusions

Deducing effects of clock jitter on sampling cannot be done with knowledge of only the RMS time jitter. Much more can be said, when the spectrum of the jitter is known as well.

Combining knowledge of the jitter spectrum with knowledge of the spectrum of the input signal, can lead to more accurate and far more relaxed estimates for clock jitter requirements in ADCs, in the example shown by more than two orders of magnitude.

Although at present many issues stand in the way of practical software radios, ADC clock jitter is not as severe a problem as often thought.

Acknowledgements

This research is supported by the PROGram for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW.

References

[1] Roel Schiphorst, Fokke W. Hoeksema, and Cornelis H. Slump, "A Bluetooth-enabled Hiperlan/2 receiver," in *Proceedings of the Vehicular Technology Conference 2003 - fall*, Oct. 2003.

[2] Rudy van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, 1994.

[3] David H. Shen, Chien-Meen Hwang, Bruce B. Lusignan, and Bruce A. Wooley, "A 900-MHz RF Front-End with Integrated Discrete-Time Filtering," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 1945–1954, Dec. 1996.

[4] Mitsuru Shinagawa, Yukio Akazawa, and Tsutomu Wakimoto, "Jitter analysis of high-speed sampling systems," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 1, pp. 220–224, Feb. 1990.

[5] Milan Patel, Izzat Darwazeh, and John J. O'Reilly, "Bandpass sampling for software radio receivers, and the effect of oversampling on aperture jitter," in *Proceedings of the 55th Vehicular Technology Conference*, May 2002.

[6] Bang-Sup Song, "High-speed pipelined adc," in *tutorial at European Solid-State Circuits Conference*, Sept. 2002.

[7] "Etsi ts 101 475 v1.2.2 (2001-02) technical specification broadband radio access networks (bran); hiperlan type 2; physical (phy) layer," Tech. Rep.

[8] Behzad Razavi, *RF Microelectronics*, Prentice Hall, 1998.

[9] Saska Lindfors, Aarno Pärssinen, and Kari A.I. Halonen, "A 3-V 230-MHz CMOS Decimation Subsampler," *IEEE Transactions on Circuits and Systems II*, vol. 50, no. 3, pp. 105–117, Mar. 2003.