GNURadio as a digital signal processing environment: application to acoustic wireless sensor measurement and time & frequency analysis of periodic signals

J.-M Friedt
SENSeOR, c/o FEMTO-ST Time & Frequency, Besançon, France
Email: jmfriedt@femto-st.fr

***Abstract*—The flexibility, reconfigurability and stability of software defined radio yield an attractive alternative to the analog strategy of probing acoustic transducers acting as passive sensors probed through a wireless link or to phase noise characterization of oscillators. However, developing processing blocks is a time consuming activity, yet metrology applications require a dedicated understanding of each processing step. We consider GNURadio as a means of exploiting opensource software as an optimum tradeoff between software re-usability yet compatible with an audit for assessing performance. This signal processing environment is demonstrated on two practical examples, FMCW probing of acoustic delay lines acting as sensors, and quartz tuning fork characterization. Both examples are considered as introductory setups for training and teaching yet a suitable environment for research activities.**

I. Introduction

Digital signal processing has been identified as a significant improvement over analog signal processing for a number of reasons [1], [2], including stability, flexibility and accuracy. Respectively, digital signal processing is not prone to component aging or drift over time; is reconfigurable in order to update algorithms or parameters in order to perform new functionalities on a given hardware; and provides a pre-defined, quantitative computation accuracy given the size of the handled datasets. Digital signal processing is nowadays ubiquitous and the mostly infinite computational power available yields to the trend of Software Defined Radio [3], in which general purpose radiofrequency source and sampling hardware is used for software processing of the recorded data for multiple application on a given experimental setup. Although processing power has been increasing continuously, the extensive bandwidth needed for radiofrequency (RF) sampling often remains beyond the general purpose hardware. On the other hand, most RF signal are narrowband and do not require full bandwidth sampling, so a common trend is to use an analog frontend with an amplifier and a mixer with a local oscillator to provide a zero-intermediate frequency (IF) configuration. We consider here the use of such a configuration in the context of time and frequency metrology signal processing, and most significantly consider the software environment needed to implement the associated processing algorithm, both from application and a teaching considerations since the skills for applying these concepts are mostly lacking.

II. FMCW RADAR for passive wireless sensor monitoring

Acoustic transducers [4]have been demonstrated to be relevant alternatives to silicon-based radiofrequency identification tags (RFID) when harsh environments (temperatures above CMOS operating range) or long interrogation ranges (no threshold on the received power in the case of piezoelectric substrate) are needed. By patterning electrodes on single crystal substrates, a delay line configuration aims at providing a simple means to convert the incoming electromagnetic signal to a mechanical wave whose propagation velocity on the surface of the substrate is dependent on the physical property under investigation [5]. Through direct piezoelectric effect, the pulse reflected on a mirror patterned on the substrate generates a returned electromagnetic signal: the purpose of the readout unit is to allow for a measurement of this time of flight in order to recover the physical quantity value [6].

A classical pulsed-mode RADAR approach requires large measurement bandwidths B in order to achieve time resolution $1/B$: typical pulse widths and maximum time of flight durations are respectively 100 ns and 5 μs, so that $B > 10$ MS/s is needed. Although the pulsed mode, wideband RADAR approach provides some significant range and measurement speed advantages, a simpler approach has been favored in the literature through the use of the frequency-modulated continuous wave (FMCW) RADAR approach [7]. In the latter method, a local oscillator frequency f is linearly swept along a bandwidth B within the bandpass function of the acoustic delay line. The time delayed returned τ signal is mixed and low pass filtered with the local oscillator, so that a beat signal $\Delta f(\tau) = f(t) - f(t+\tau)$ is recorded after removal of the carrier. The resulting architecture is simple and only requires low frequency sampling rates, but also introduces multiple free parameters which are best tested in a software defined radio approach. Among the parameters are the central operating frequency f and the sweep rate T: $\Delta f(\tau)/\tau = B/T$ since the frequency band B is swept linearly over a duration T.

Recently, a tutorial has been presented on the MIT OpenCourseWare web site [8] demonstrating a basic FMCW approach: this hardware platform is used as the starting point of this experiment, although the linear RF oscillator sweep and recording a no longer performed by dedicated hardware but by a personal computer sound card running GNURadio (http:

//gnuradio.org/). The recorded signals are Fourier transformed in real time in order to display the multiple echoes returned by the acoustic delay line $\Delta f(\tau_i)$ with i an index of the echo number. In the case of the transducer we have experimented with – available from the Carithian Tech Research (CTR, Villach, Austria), $i \in [1..8]$.

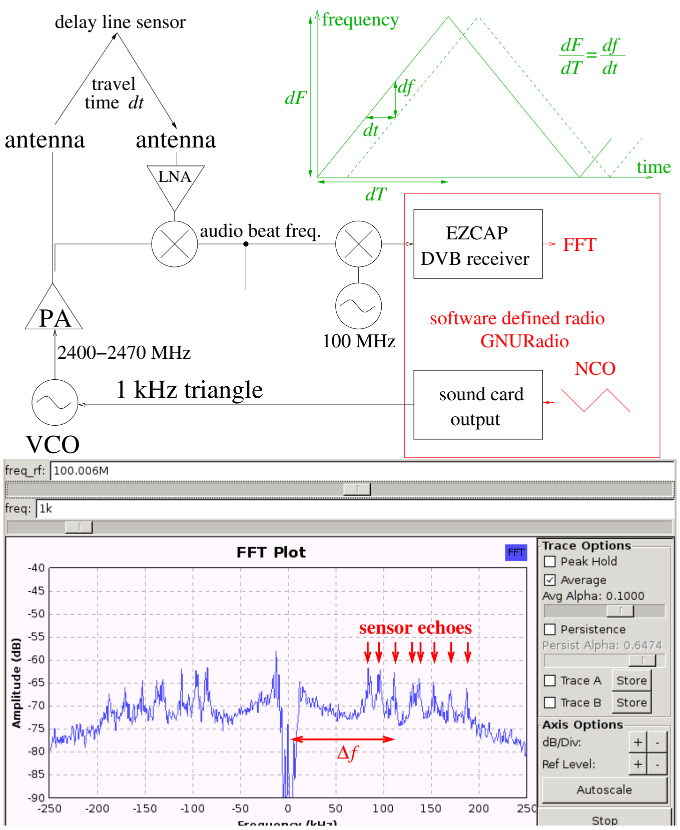

Fig. 1. Top: schematic diagram of the FMCW RADAR used to demonstrate probing a 2.45 GHz acoustic delay line acting as passive, wireless sensor. The RF hardware is based on the MIT design [8], while the signal source (triangle-shaped Voltage Controlled Oscillator – VCO – control signal) and sink (beat frequency recording and Fourier transform) are performed using the GNURadio software. Since the recording by an DVB receiver requires an input signal higher than 100 MHz, the audio beat signal is mixed with a fixed frequency oscillator to reach the requested frequency band. Bottom: real time display of the 8 echoes observed as 8 discrete frequency components of the beat signal.

Fig. 1 exhibits a screenshot of the monitored signal, in which the 8 delay line echoes are well resolved as 8-discrete beat frequencies. The software defined signal source – here a linear sweep on the output sound card – provides the flexibility to compensate for the RF oscillator non-linear frequency output with respect to the control voltage: this linearity on the whole range B is mandatory for all contributions to the beat signal to add coherently. Lack of linearity of the RF oscillator yields a given delay τ to induce variable $\Delta f(\tau)$ as the oscillator is swept and hence a broadening of the returned signal beat frequency peaks. This aspect is well demonstrated with the software defined radio as well.

For teaching purposes, the two options available to us when using widely available, commercial off the shelf analog to digital converters, are a sound card and a Digital Video Broadcast (DVB) receiver, both acting as dual I and Q channel data streams. The sound card is a high resolution analog to digital converter (typically 16 bit at least) but only provides a bandwidth up to 48 kHz (96 kHz sampling rate), while internal anti-aliasing filters prevent the monitoring of higher frequency signals. Since the CTR-delay line requires a bandwidth of at least 30 MHz and the time delay is in the 5 μs range, the sweep rate T must be $T < \frac{B\times\tau}{\Delta f(min)}$ or in this case 3.125 ms. An output frequency of 320 Hz is well within the bandpass of a soundcard output. Alternatively, higher refresh rates are achieved by sweeping the frequency source at a faster rate, hence also improving the frequency detection resolution since Δf becomes larger for a given delay τ. Since the sound card is no longer an option at higher signal frequencies and the DVB receiver only operates above 100 MHz, a frequency transposition strategy is used to bring the monitored signal in the 100 MHz range (mixing with a local quartz oscillator) before the DVB receiver demodulates the signal and provides the I/Q datastream.

III. Time & Frequency Metrology Algorithm Prototyping and Assessing Hardware Limitations

Software implementation of classical analog signal processing methods are best suited for rapid prototyping [2]. While frequency counting is intrinsically a digital method, its software implementation easily demonstrates the gains and influence on the statistics of various strategies (direct counter, reciprocal counter, sliding average functions [9], [10]). More challenging is the phase noise measurement [11]–[14], which is classically performed by mixing the test oscillator with a reference local oscillator and low-pass filtering the beat signal, possibly providing a feedback control signal for the local oscillator to track the measured oscillator frequency if long term drift is compensated for (statistics on the control signal rather than on the beat signal output itself). All these blocks are implemented as software, but the influence on the phase noise spectra as a function of the digital hardware performance often remains challenging to assess [15] .

Efficient processing of the narrowband signals considered remains a challenging aspect of the data acquisition chain: under the assumption that the oscillator under investigation is stable, sampling is not needed to meet the Shannon criteria under the hypothesis that only a narrow band around the carrier frequency holds the needed information. One such consideration yields to the fact that aliasing on purpose the measured signal to only meet the analyzed bandwidth – without the ability to reconstruct the whole signal – is enough, yet the impact on the noise level is still under consideration.

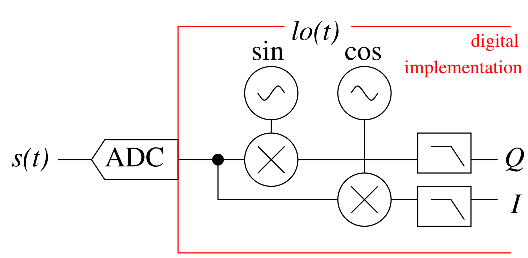

Fig. 2. Classical phase detection scheme considered here in its digital implementation.

When single sample operations are needed, such as

I/Q demodulation in which a local numerically controlled oscillator (NCO, described in the GNURadio source tree in `gnuradio-core/src/lib/general/gr_nco.h`) is mixed with the incoming signal, no stream handling functionality is needed. However, the data block size is not known and data are transmitted from one block to another asynchronously. If a minimum number of data is needed to perform a given processing step, e.g a Fourier transform, data are accumulated in a temporary array which is processed only once filled. The buffer is then freed of the processed data and the remaining values are moved to the beginning of the buffer to be processed with the next batch of data.

Phase noise analysis is the most basic processing tool for characterizing an oscillator stability. Many schemes have been devised using analog electronics, most classically mixing the oscillator output under investigation with a local oscillator assumed to exhibit better stability, low pass filtering the mixer output to get rid of the signal at the sum of the frequencies, and controlling the reference oscillator in order to keep the quadrature condition at the mixer so that $\sin(\varphi) \simeq \varphi$. In order to implement such a scheme digitally and use at best digital signal processing – flexibility in setting the local oscillator frequency, low pass filter characteristics and phase noise extraction – we assess the number of bits on which the computation has to be performed to comply with the targeted resolution. Let us assume, to justify this calculation, that we aim for phase noise resolution calculations reaching -180 dBrad^2/Hz (Fig. 3).

The phase noise S_φ defines the phase fluctuations σ_φ in a given bandwidth BW:

$$S_\varphi = \sigma_\varphi^2/BW \text{ rad}^2\text{/Hz}$$

Considering the phase noise at carrier offset 1 to 10 MHz and 1000 points/decade, $BW = 9$ kHz in such conditions. S_φ is usually expressed in dB with $S_{dB} = 10\log_{10}(S_\varphi)$. The classical scheme of mixing the sine wave translates, in terms of digital computation [16], [17], into a product of the local numerically controlled oscillator LO and sampled signals from the oscillator under investigation s, followed by a low pass filter which we here consider to be a finite impulse response (FIR) filter with coefficients a_n:

$$I, Q = \frac{1}{N}\sum_N s_k \times LO_k \times a_{N-k}$$

where I, Q are the in-phase and quadrature coefficients obtained from two expressions of the local oscillator phase shifted by 90^o (*i.e.* expressed as cos and sin). From these two sets of coefficient, the phase is classically expressed as the argument of the complex $I + jQ$ or $\varphi = \arctan(Q/I)$.

In the low pass filter expression, the sum yields a standard deviation rising as $\sqrt{N}$ for samples affected by an additive gaussian noise, and the $1/N$ factor yields the classical decrease of the standard deviation during averaging by $1/\sqrt{N}$. This number of samples N is given by the ratio of the sampling frequency f_s to twice the maximum offset from the carrier at which phase noise is computed. Assuming a 500 Msamples/s sampling rate and a phase noise calculation at 10 MHz from the carrier, then $N = 25$.

The uncertainty of the I, Q calculation is given by the sum of the uncertainties of each term in the sum: $dI, Q = \frac{1}{N}\sum_N ds \times LO \times a_{N-k} + dLO \times s \times a_{N-k} + da \times LO \times s$. If only quantization noise is considered when representing information on M bits, then all infinitesimal terms are equal to 2^{-M} and each quantity is assumed to have been scaled to be around unity, so that this uncertainty on the computation of I and Q are dominated by $3 \times 2^{-M}/\sqrt{N}$ since the sum of the independent noise source rises as $\sqrt{N}$. We then target a given phase noise floor S_dB expressed in dB, then

$$\sqrt{BW \times 10^{S_dB/10}} = 3/\sqrt{N} \times 2^{-M}$$

$$\Leftrightarrow M = -\ln_2\left(\sqrt{N \cdot BW \times 10^{S_dB/10}}\right)/3$$

Keeping the previous assumption of $B = 500$ MHz and a targeted phase noise plot up to a 10 MHz offset from the carrier hence requiring $N = 25$, computation on 26 bits is needed to reach the targeted resolution of -180 dBrad^2/Hz. Even at -160 dBrad^2/Hz, $M = 23$ bits are needed to reach the requested computation resolution. The sum of three sources of noises rises the resolution need by 1.6 bits: if the filter can be tuned in order to prevent resolution uncertainty on its coefficients ($da = 0$) and if a square wave is used instead of the sine wave defining the NCO ($dLO = 0$), then the needed resolution becomes 25 and 21 bits respectively. However, the use of a square wave NCO brings new challenges including the rejection of unwanted harmonics and its aliases after the mixing step, while making sure than none of these aliases are brought back within the band of interest [18], [19].

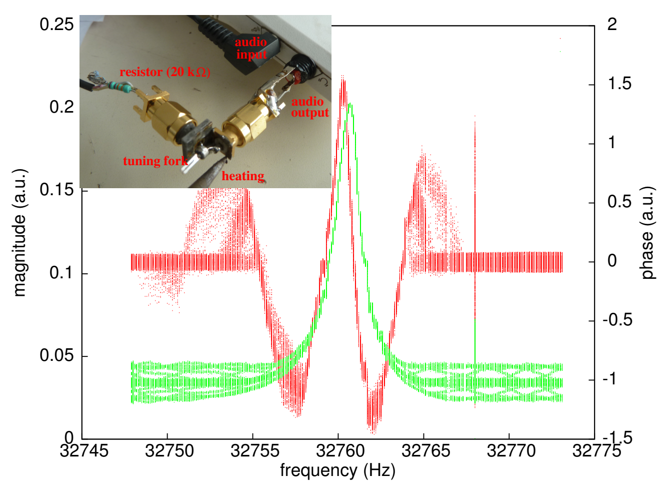

Fig. 3. Illustration of the I/Q demodulation scheme on a low frequency quartz tuning fork yielding phase (red) and magnitude (green) of the audiofrequency signal transmitted through the dipole. Notice that this experiment only requires a full-duplex sound card programmed through GNURadio as a network analyzer since the emission and recording are performed simultaneously. In this particular application, phase monitoring is used when the unpackaged tuning fork acts as a temperature sensor.

IV. LIMITATIONS OF THE GNURADIO APPROACH

While most digital signal processing blocks are readily available and the missing functions dedicated to time & frequency metrology are being implemented thanks to the opensource software aspect of the tool, one major limitation in terms of signal processing bandwidth is due to the software running on general purpose central processing units (CPU).

Although dedicated hardware provides FPGA-based data acquisition, the signal processing is performed on the data-flow recovered from the FPGA and not implemented on the gate matrix itself, thus reducing the processing bandwidth to the communication bus bandwidth. Hence, while this environment provides a flexible testbench for real time prototyping of signal processing algorithms, it might not be suited when large (> 100 MS/s) bandwidths are considered. Alternate solutions include on-purpose aliasing or stroboscopy [19], [20].

Nevertheless, we consider this environment best suited for getting familiar with the basics of digital signal processing of RF signals, as well as for teaching purposes since this aspect is needed to train the current and future generation of researchers interested in such a topic. The easily available hardware (sound card, DVB receiver) and the ability to add signal source blocks thanks to the opensource code makes this software most versatile.

V. Conclusion

While analog implementations of most radiofrequency signal processing schemes provide suitable results for RADAR detection of signals returned from acoustic sensors acting as passive sensors and for characterizing oscillator stability, the need to design new hardware for each new application or new frequency band is time consuming and prone to the introduction of design errors in each new hardware setup. The flexibility of software reconfigurable processing chains is well suited for such applications. However, the need for a detailed understanding of the methods implemented in each processing block is hardly compatible with the use of closed source software, and an opensource solution as offered by the GNURadio software suite is considered. Reaching the targeted resolution of low phase noise oscillator using a full software implementation in which the RF front end is solely made of a fast analog to digital converter and all further processing steps are performed by software appears challenging with current available technologies.

Acknowledgements

These activities were motivated by the Oscillator IMP LABEX grant.

References

[1] D. Mindell, *Digital Apollo: Human and Machine in Spaceflight*. The MIT Press, 2011.

[2] J. Hamkins, M. Simon, and J. Yuen, *Autonomous software defined radio receivers for deep space applications*. Wiley, 2006.

[3] P. Kenington, *RF and baseband techniques for software defined radio*. Artech House, 2005.

[4] L. Reindl and V. Plessky, "Review on SAW RFID tags," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, vol. 57, no. 3, pp. 654–668, 2010.

[5] R. White, "Surface acoustic wave sensors," in *Proc. IEEE Ultrasonics Symposium*, 1985, pp. 490–494.

[6] W.-E. Bulst, G. Fischerauer, and L. Reindl, "State of the art in wireless sensing with surface acoustic waves," *IEEE Trans. Industrial Electronics*, vol. 48, no. 2, pp. 265–271, 2001.

[7] S. Schuster, S. Scheiblhofer, L. Reindl, and A. Stelzer, "Performance evaluation of algorithms for SAW-based temperature measurement," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, 2006.

[8] G. Charvat, A. Fenn, and B. Perry, "The MIT IAP radar course: Build a small radar system capable of sensing range, doppler, and synthetic aperture (SAR) imaging," in *IEEE Radar Conference*, 2012, pp. 138–144.

[9] S. Dawkins, J. McFerran, and A. Luiten, "Considerations on the measurement of the stability of oscillators with frequency counters," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 54, no. 5, pp. 918–925, May 2007.

[10] E. Rubiola, "On the measurement of frequency and of its sample variance with high-resolution counters," *Rev. Sci. Instrum.*, vol. 76, no. 5, p. 054703, May 2005.

[11] L. Angrisani, M. D'Apuzzo, and M. D'Arco, "A digital signal-processing approach for phase noise measurement," *IEEE Transactions on instrumentation and measurement*, vol. 50, no. 4, pp. 930–935, August 2001.

[12] J. Grove, J. Hein, J. Retta, P. Schweiger, W. Solbrig, and S. Stein, "Direct-digital phase-noise measurement," in *IEEE International Ultrasonics, Ferroelectrics, and Frequency Control Joint 50th Anniversary Conference*, 2004, pp. 287–291.

[13] L. Angrisani, A. Baccigalupi, M.D'Arco, and L. Ferrigno, "A new digital signal processing method for accurate phase noise measurement," in *11th IMEKO TC-4 Symp. – Trends in Electrical Measurement and Instrumentation*, Lisbon, Portugal, September 2001, pp. 43–.

[14] V. Ivo and H. Josef, "Phase noise measurement," in *Proceedings of the Radioelektronika*, 2001, pp. 131–134.

[15] L. Angrisani, M. D'Arco, C. Greenhal, and R. L. Moriello, "Real-time phase noise meter based on a digital signal processor," in *IMTC - Instrumentation and Measurement Technology Conference*, Sonento, Italy, April 2006.

[16] A. Hati, D. A. Howe, F. L. Walls, and D. Walker, "Noise figure vs. PM noise measurements: A study at microwave frequencies," in *Proc. of the 2003 IEEE International Frequency Control Symposium and PDA Exhibition, Jointly with the 17th European Frequency and Time Forum*, 2003, pp. 516–520.

[17] A. Hati, D. Howe, F. Walls, and D. Walker, "Merits of PM noise measurement over noise figure: A study at microwave frequencies," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 53, no. 10, pp. 1889–1894, October 2006.

[18] M. Mishali and Y. Eldar, "Sub-Nyquist sampling," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 98–124, 2011.

[19] ——, "Sub-Nyquist sampling: Bridging theory and practice," *arXiv: 1106.4514v1*, 2011.

[20] C. Nelson and D. Howe, "A sub-sampling digital PM/AM noise measurement system," in *Proc. IEEE International Frequency Control Symposium (FCS)*, 2012, pp. 1–4.