

Real-time phase noise meter based on a digital signal processor

Leopoldo Angrisani⁽¹⁾, Mauro D'Arco⁽²⁾, Charles Greenhall⁽³⁾, Rosario Schiano Lo Moriello⁽²⁾

⁽¹⁾ Dip. di Informatica e Sistemistica, Tel: +39 081 7683170, e-mail: angrisan@unina.it

⁽²⁾ Dip. di Ingegneria Elettrica, Tel: +39 081 7683238, Fax: +39 081 2396897, e-mail: {darco; rschiano @unina.it}

^{(1) (2)} Università di Napoli Federico II, via Claudio 21, 80125 Napoli, ITALY

⁽³⁾ Jet Propulsion Laboratory (CIT). Tel: +1 818 393-6944 Fax: +1 818 393-6773, e-mail: cgreenhall@jpl.nasa.gov
4800 Oak Grove Dr., MS 298-100. Pasadena, CA 91109, USA*

Abstract – A digital signal-processing meter for phase noise measurement on sinusoidal signals is dealt with. It enlists a special hardware architecture, made up of a core digital signal processor connected to a data acquisition board, and takes advantage of a quadrature demodulation-based measurement scheme, already proposed by the authors. Thanks to an efficient measurement process and an optimized implementation of its fundamental stages, the proposed meter succeeds in exploiting all hardware resources in such an effective way as to gain high performance and real-time operation. For input frequencies up to some hundreds of kilohertz, the meter is capable both of updating phase noise power spectrum while seamlessly capturing the analyzed signal into its memory, and granting as good frequency resolution as few units of hertz.

After a brief outline of the aforementioned measurement scheme, key features of the adopted digital signal processor are highlighted. Hardware architecture, measurement process, and software strategies peculiar to the proposed meter are then described in detail. At the end, some results of experimental tests, carried out on sinusoidal signals provided by function generators and arbitrary waveform generators, give evidence of the meter's reliability and efficacy.

Keywords – Phase noise measurement, Digital signal processor, Data acquisition board, Power spectrum evaluation, Quadrature demodulation.

I. INTRODUCTION

Phase noise consists of random phase fluctuations that distort the linear trend of the phase of a periodic signal. It represents a severe limit for time precision of clocks and frequency stability of oscillators [1]. As a consequence, it affects the performance of a number of optical, microwave, and radiofrequency systems, such as satellite positioning apparatuses, where high precision clocks are required for accurate distance measurements [2], and analog and digital communication transmitters, where unstable carriers degrade modulation quality and signal integrity [3]-[5].

IEEE Std 1139-1999 gives some methods for phase noise description, and enlists power spectral density of phase fluctuations as standard measure for phase noise characterization [6].

Calibration services for dedicated phase noise measurement instrumentation are available from NIST, where traceable phase noise primary standards are realized and preserved [7].

Different analog techniques for phase noise measurement in time and frequency domain have gained acceptance worldwide. They rely on phase locking strategies, interferometric approaches, and direct spectrum analysis [8]-[10] but require some expertise and knowledge of the adopted measurement principles and phase noise topics for configuring instruments, executing measurement procedures, and correctly interpreting final results. Each of them exhibits particular shortcomings. Nonzero extent of phase-locked-loop bandwidth limits close-to-the-carrier measurement. Interferometric approaches only measure phase noise added by two-port devices; inherent phase noise of an active source cannot be analyzed. Direct spectrum analysis offers poor results whenever either AM noise or PM and/or FM residuals affect the analyzed signal.

Digital signal-processing approaches for phase noise measurement have also been proposed [11]-[14]. They are recently gaining great interest due to the availability of high performance data acquisition systems. Actually, they could be utilized for developing either advanced measurement apparatuses, capable of overcoming the limits of analog techniques, as well as cheap and automatic solutions for mass-production tests, where money saving and short time-to-market are of major concern. The earliest proposals, however, exhibit some drawbacks. In particular, the method presented in [11], being specialized for far-from-the-carrier measurement, suffers from poor frequency resolution; information about phase-noise at low frequency offsets is thus difficult to be achieved. The methods presented in [12] and [13] offer satisfying frequency resolution for close-to-the-carrier analysis, but the values of the obtained phase-noise power spectrum become more and more unreliable upon the frequency offset's getting far from the carrier. These drawbacks have recently been overcome by a more mature method that attains phase noise power spectrum through a series of stages, each of which provides for the evaluation of a different portion of the spectrum, with a different frequency resolution [14]. A suitable measurement

* The work of this author was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

apparatus, made up of a data acquisition system and an external processing unit, has also been proposed in [14]. The apparatus, however, does not succeed in highlighting all method potentialities. The use of an external processor for a batch execution of the measurement algorithm is, in fact, a bottleneck that has to be removed.

The paper presents a real-time phase noise meter, which works on sinusoidal signals and exploits the measurement scheme peculiar to the method in [14]. Its special hardware architecture, providing for a core digital signal processor directly connected to a data acquisition board, and efficient measurement process, arranging measured phase noise data in a suitable hierarchy, allow the meter to operate in real-time. Phase noise power spectrum is, in fact, continuously updated while input signal is seamlessly acquired and captured into memory. Moreover, variable frequency resolution is achieved, optimized according to the specific range of analysis of the measured power spectrum.

II. PRELIMINARY NOTES

A. Measurement scheme

The measurement scheme, which the proposed meter relies on, aims at estimating the power spectrum of input signal phase noise through six digital signal processing stages, namely

1. multiplication of input signal by two orthogonal sinusoidal signals (mixing stage),
2. low-pass filtering,
3. arctangent evaluation,
4. phase unwrapping and decimation,
5. compensation for frequency deviation,
6. phase noise power spectrum evaluation through fast Fourier transform (FFT).

All stages are shown in the block diagram given in Fig.1. The first four stages are repeatedly performed in order to acquire different phase noise segments, i.e. arrays collecting the discrete time evolution of phase noise.

Phase noise power spectrum is evaluated with variable frequency resolution. FFT is performed, in fact, on phase noise segments having the same length but characterized by different sample rates. From each segment a different portion of the whole power spectrum is achieved. Far-from-the-carrier portions are related to segments characterized by high sample rates, while close-to-the-carrier ones are derived from segments acquired at low sample rates, which offer finer frequency resolution. Linking all portions gives the whole power spectrum. More details can be found in [14].

B. Adopted digital signal processor

The adopted digital signal processor, the ADSP-21364™ processor, is specifically addressed to real-time applications. Fig.2 shows a simplified block diagram of its architecture, in which three fundamental parts can be distinguished: the core processor, memory block and I/O processor. The core

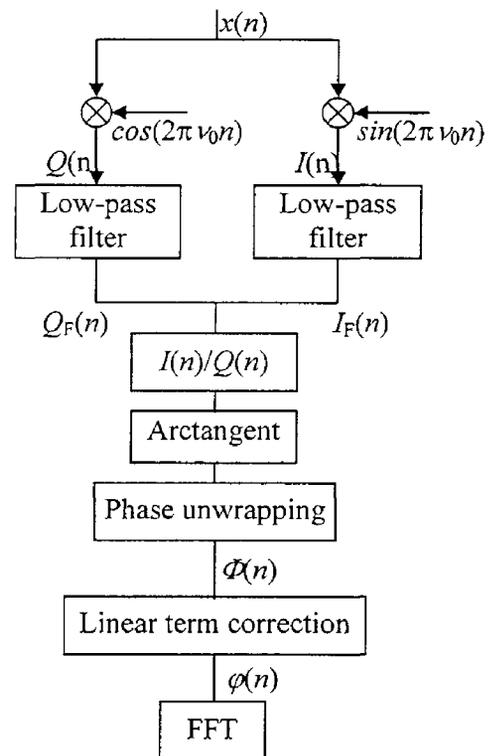


Fig.1. Block diagram of the measurement scheme

processor has two computational units, namely PEx and PEy, and a cache memory to store recurrent instructions. Each computational unit is equipped with a multiplier, an ALU, a shifter and a data register file, and can execute a multiplication, a sum and a dual data fetch in a single cycle, which is useful when algorithms based on massive use of multiplications, additions, and accumulations are involved. During normal operation, called single-instruction-single-data (SISD) mode, a single computational unit is active. Concurrent operation of both computational units, called single instruction multiple data (SIMD) mode, can be also enabled. In particular, PEx and PEy can execute the same instruction on different input data, provided that input data are placed at consecutive addresses in memory. Moreover, when data are involved in looped operations, the computational units are disengaged from data routing in order to achieve full speed execution. Addressing operation is performed, in this case, by two data address generators, namely DAG1 and DAG2.

The memory block is partitioned into data memory for collecting input data, program memory for saving intermediate and final results, and instructions memory for executable binary code. Memory and core processor are connected by two 32-bit buses for addressing operations, and 64-bit buses, named DM and PM, each of them capable of fetching or downloading couples of 32-bit data placed at consecutive addresses. The PM bus can be used to transport either data or instructions. Thanks to the DM and PM buses, the core processor can simultaneously fetch four operands,

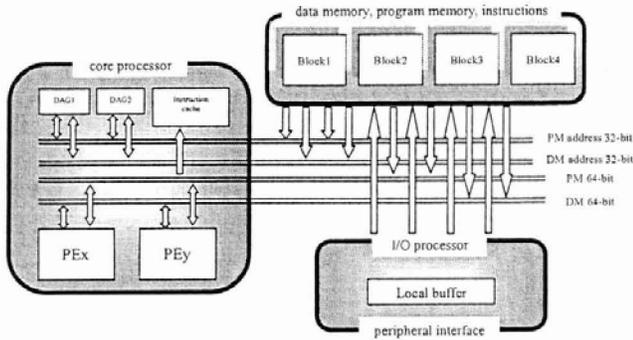


Fig.2. Simplified block diagram of ADSP Sharc processor architecture.

two over each bus, and one instruction from the cache, all in a single cycle.

A dedicated I/O processor, also shown in Fig.2, performs direct memory access to transfer data from the peripheral interface to memory, independently and invisibly to the core processor. The core processor can be interfaced to external devices such as A/D converters or external memories.

III. PHASE NOISE METER

A. Hardware architecture

The proposed meter is made up of ADSP-21364TM Sharc processor, mounted on a compatible development board by Analog Devices (EZKIT). The board is equipped with an extender that allows interconnection to an external analog-to-digital converter, namely AD9244TM, for the digitization of the signal under test (Fig.3). The converter is characterized by 14-bit resolution and 65 MS/s maximum sample rate. The sample clock of the converter can be provided either by the processor itself through its internal precision clock generator or an external source; in both cases it is four times greater than input carrier frequency.

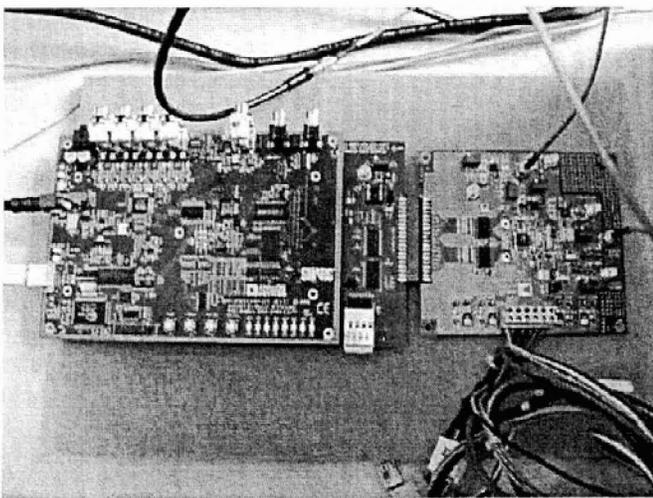


Fig.3. Main hardware components of the prototype. Sharc ADSP-21364TM is characterized by 333 MHz clock speed, 3 Mbit SRAM, and I/O processor for direct memory access.

B. Measurement process

The meter manages four segments of measured phase noise data, organized in a four-level hierarchy into memory. More specifically, input signal phase noise is extracted in real-time and stored in a memory segment, which is managed according to the common first-in-first-out (FIFO) approach. This segment is the root of a hierarchy made up of other identical segments. Each segment in the hierarchy collects a low-pass decimated version of the time series flowing in the segment immediately above. Going down through the hierarchy, captures of phase noise covering longer and longer time intervals are available, by means of which the inspection of lower and lower frequency ranges of phase noise power spectrum is made possible. The process runs until the lowest segment in the hierarchy, which results from multiple decimation steps, is collected and analyzed through the FFT.

Due to some technical constraints, the highest processing rate is reached using segments the length of which is a power of two. Consequently, the use of a decimation factor of 8 to pass data from an upper segment to the lower one has been advisable. In particular, the second, third and fourth segment are decimated by 8, 64, and 512, respectively. Moreover, each sample of a decimated segment is obtained as the average of a number P ($P = 8$) of samples flowing in the next upper segment. Averaging permits enhanced sensitivity in the analysis of power spectrum at very low frequencies.

C. Software architecture

The code for the ADSP-21364 is written in C and ADSP assembly languages. C language has been used when high-level abstraction is pursued, (initialization and execution-flow-control routines), while assembly language has been preferred when speed is crucial, (fundamental stages of the measurement algorithm). In detail, it consists of a boot thread aimed at system initialization, and a measurement thread, which runs until all segments in the hierarchy have been filled and analyzed.

➤ Boot thread

The boot thread is launched at startup. It puts the thread mandated to phase noise measurement in a sleeping state, and establishes an event object to wake it up. Then, it ends by unmasking an interrupt request that activates data transfer from the A/D converter to processor buffer.

Data transfer is managed by the on-board I/O processor, that moves data, transferred from A/D converter and collected into a local buffer of the peripheral interface, to the data memory independently and invisibly to the core processor.

➤ Measurement thread

The measurement thread wakes on the event established by the boot thread. The event occurs when 1024 samples have been collected in the data memory, which is organized according to a double-buffer technique. The measurement algorithm can operate on data collected in the first half of the

buffer, while successive samples are transferred from the peripheral interface to the other half.

Mixing stage - Thanks to the particular choice of the sample rate, multiplication of the digitized signal with orthogonal sinusoidal signals can be attained using very simple sequences, namely $\{0,1,0,-1\}$ and $\{1,0,-1,0\}$, i.e. the values assumed by the sine and cosine functions at $0, \pi/2, \pi, 3\pi/2$ arguments, thus avoiding long sequence storage. Fig.4 shows a portion of the file registers of both processors PEx and PEy, which are used during mixing; PEx uses registers RN, while PEy uses registers SN ($N = 0, \dots, 7$). Both input data and those resulting after calculation are depicted; they are separated by slashes. In detail, the sequence $\{0,1,0,-1\}$, used to attain the $I(n)$ signal of Fig.1, is preset respectively into the registers R0, S0, R1, and S1, while $\{1,0,-1,0\}$, used to attain the $Q(n)$ signal, is preset into R2, S2, R3, and S3. At a generic step, in a single clock cycle and by dual fetch operations, the registers R6 and S6 are updated with samples $x(k)$ and $x(k+1)$, values in R4 and S4 are downloaded into program memory, where the results of mixing stage are collected, and values in R5 and S5 are multiplied with R1 and S1, respectively, overwriting R5 and S5 to save the results. At the next step, the registers R7 and S7 are updated with $x(k+2)$ and $x(k+3)$, values in R5 and S5 are downloaded into program memory, and values in R6 and S6 are multiplied with R2 and S2, respectively, overwriting R6 and S6 to save the results. When the latter step is completed, the results related to the mixing of the incoming signal with the sequence preset into the registers R0, S0, R1, and S1, up to the sample $x(k+3)$, have been downloaded into program memory, and those related to the other sequence are available to be downloaded into the registers R6, S6, R7, and S7. The mixing stage goes on updating the registers R4 and S4 for mixing the samples $x(k+4)$ and $x(k+5)$, and concurrently downloading the results in R6 and S6. The mixing stage produces the unfiltered $I(n)$ and $Q(n)$ components in 1024 steps.

Low-pass filtering - To remove high frequency components and retain only base-band components, low-pass filtering has been executed on $I(n)$ and $Q(n)$ signals through a M -tap finite

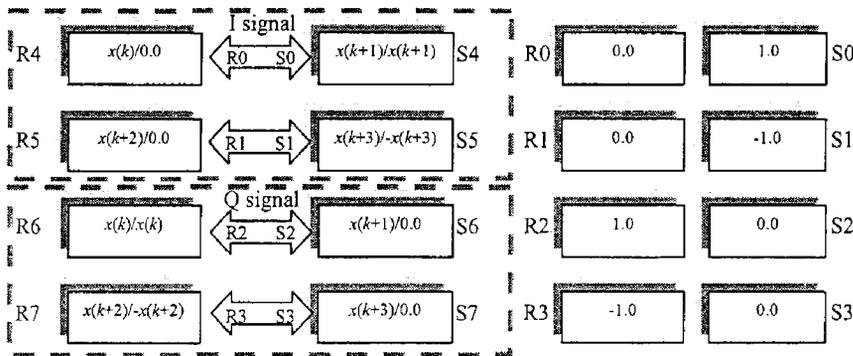


Fig.4. Register files of computational units peculiar to ADSP Sharc processor. Registers RN's are used by PEx, registers SN's by PEy. At each step a pair of input data at consecutive addresses is loaded into appropriate registers RN and SN. Slashes separate input data from data resulting after multiplication.

impulse response (FIR) filter, ($M = 64$), which has been chosen as the right trade-off between the time needed to filtering and stop-band attenuation.

FIR filtering is performed in the frequency domain. The signals are zero-padded up to 2048 points, and then transformed from the time into frequency domain using the FFT. The transformed sequence is multiplied by the filter's frequency response, which is available in the program memory, and the result is transformed back into the time domain using the inverse FFT. As well known, when filtering is performed through FFT-based approaches, successive blocks must be added while overlapped by $M-1$ samples, in order to produce the correct filtered results. A suitable buffer in the program memory collects the samples that must be overlapped and added to the results of the next filter operation. Though this approach sounds more complicated than filter implementation in the time domain, it is actually much faster due to the use of FFT radix 4 algorithm, which is optimized for SIMD mode featured by the adopted hardware [15].

Arctangent phase unwrapping and decimation - The ratio between the filtered $I_f(n)$ and $Q_f(n)$ components is passed to a four-quadrant arctangent function, which returns the arctangent of the argument that can range from $-\pi$ to π rad. Phase unwrapping is then needed to eliminate discontinuities around $\pm\pi$ introduced by the arctangent function in the presence of a frequency offset. To correct discontinuities a 5-state finite state machine (FSM) has been implemented, which is capable of operating even though the inherent random nature of measurement noise produces multiple accumulated discontinuities. The FSM uses two state registers, which for the sake of convenience are referred to as φ and θ , φ accounts for each state value, while θ records 2π -cycles accumulated. The updated phase sample is obtained by adding the values of both registers.

FSM operation is illustrated through the state diagram shown in Fig.5, where the round rectangles are states, and the arrows, called transitions, describe how the FSM evolves from one state to another; bold and dotted arches are used to distinguish critical transitions. The label on a transition can have two parts separated by a slash. The first is the event that triggers the transition. The second is the action to be performed once the transition has been triggered. As it can be observed, FSM moves the discontinuity points from $\pm\pi$ to $\pm\pi/3$. When a transition associated to a dotted arch occurs, 2π is added (or subtracted), but it isn't stored into FSM state register θ . On the contrary, in the presence of transitions associated to bold arches, 2π is added and stored in θ .

After the execution of phase unwrapping, the attained 1024-points segment is decimated three times, by 8,

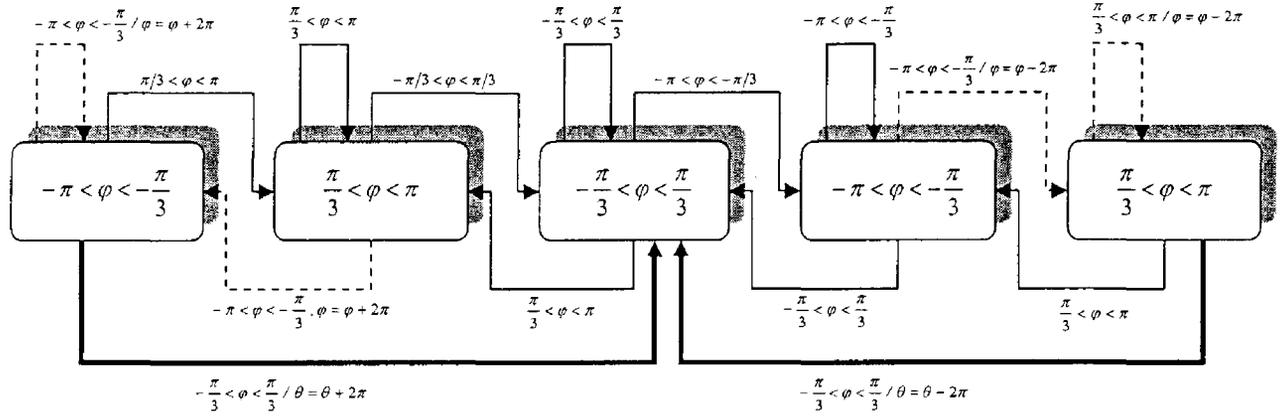


Fig.5. FSM state diagram. Round rectangles represent different states. Arrows represent transitions, which describe how FSM evolves from one state to another. Any transition has a label, which can have two parts separated by a slash; the first is the event that triggers the transition, the second is the action to be performed once the transition has been triggered; if the second part is omitted no action has to be undertaken.

64, and 512, attaining three shorter sequences, which are appended to the other segments of the hierarchy. The procedure stops when the lowest segment in the adopted hierarchy has been filled.

Compensation for frequency deviation - In each segment of the hierarchy, the available discrete-time signal can be represented according to:

$$\Phi(n) = \varphi(n) + 2\pi(\nu_x - \nu_0)n \quad (1)$$

where ν_x and ν_0 are, respectively, the real frequency of the analyzed signal, and the nominal one, both normalized to the actual decimated sample rate. To remove the linear term $2\pi(\nu_x - \nu_0)n$, which is a deterministic effect due to the lack of accuracy on ν_x , and achieve the only term related to phase noise, $2\pi(\nu_x - \nu_0)n$ has to be estimated and, accordingly, a straight line has to be subtracted from $\Phi(n)$. The straight line is estimated through a common end-points line technique.

Phase noise power spectrum evaluation through FFT After frequency deviation correction, each record is first multiplied by a Hanning window and then processed in order to evaluate its periodogram according to:

$$S_x^i(\nu) = \frac{1}{NW_n} |X(\nu)|^2 \quad (2)$$

where i is the current iteration, ν is the normalized frequency, W_n stands for the energy in the adopted window, and $X(\nu)$ is evaluated through the FFT of the windowed record. The phase noise power spectrum is then estimated by averaging K periodograms related to consecutive records

$$\bar{S}_x(\nu) = \frac{1}{K} \sum_{i=1}^K S_x^i(\nu) \quad (3)$$

IV. PERFORMANCE ASSESSMENT

The performance of the proposed phase noise meter has been assessed through a number of experimental tests,

during which concurrence of its output results with those offered by a spectrum analyzer, namely *Agilent Technologies HP 8594ETM*, has also been checked. Typical phase noise affecting sinusoidal carriers has been generated through an arbitrary waveform generator, namely *Tektronix AWG2020TM*. The same setup shown in [14] has been arranged to attain signals characterized by a power spectrum with $1/f$ evolution versus frequency, generally referred to as flicker noise signals. *Agilent Technologies 33220ATM* signal generator has provided the timing clock, both uniform and jittered. Flicker noise signals have modulated, as external sources, the phase of sinusoidal carriers used in the tests and provided by *HP 8648BTM* signal generator.

A. Sensitivity assessment

To assess the noise floor of the proposed phase noise meter, several signals characterized by high spectral purity have been analyzed. In particular, *Agilent Technologies 33220ATM* signal generator has provided these carriers.

Noise floor has, in particular, been obtained by averaging 15 consecutive periodograms. It has resulted slightly dependent on the adopted sample rate. As an example, Fig.6 shows the noise floor exhibited by the meter after analyzing a sinusoidal signal with a frequency equal to 250 kHz, which is the current upper limit of input frequency range.

B. Comparative analysis

Several tests have been conducted in order to assess the performance of the proposed phase noise meter in terms of maximum sample rate and concurrence with the outcomes of the spectrum analyzer, assumed as reference. Due to well-known difficulty of spectrum analyzers of carrying out reliable close-to-the-carrier measurements, only results accounting for far-from-the-carrier phase noise have been considered in order to draw a proper comparison. Moreover, in each test the resolution bandwidth of the spectrum analyzer has been regulated in such a way as to be as close as

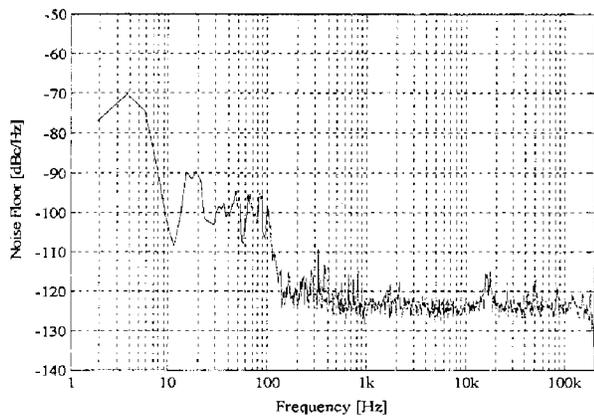


Fig.6. Noise floor of the proposed phase noise meter after analyzing a sinusoidal signal with a frequency equal to 250 kHz. The adopted sample rate has been equal to 1 MS/s.

possible to worst frequency resolution assured by the meter in the considered portion of phase noise power spectrum. As an example, a portion both of power spectrum provided by the proposed meter (15 periodograms averaged) and that measured by the spectrum analyzer is given in Fig.7. These results refer to a sinusoidal signal, characterized by a frequency equal to 204.8 kHz and whose phase is modulated by flicker noise; the adopted sample rate has been equal to 819.2 kS/s. The resolution bandwidth of the spectrum analyzer has been fixed equal to 1 kHz, while the worst frequency resolution offered by the meter has been 800 Hz. Very good concurrence has been noticed in all tests.

V. CONCLUSIONS

A new meter mandated to phase noise measurement on sinusoidal signals has been presented. Main tasks of its core digital signal processor, fundamental stages of measurement process, and key features of software architecture have been illustrated in detail. Particular emphasis has been paid to all design choices that greatly contribute to grant real-time operation; the meter is, in fact, capable of updating its

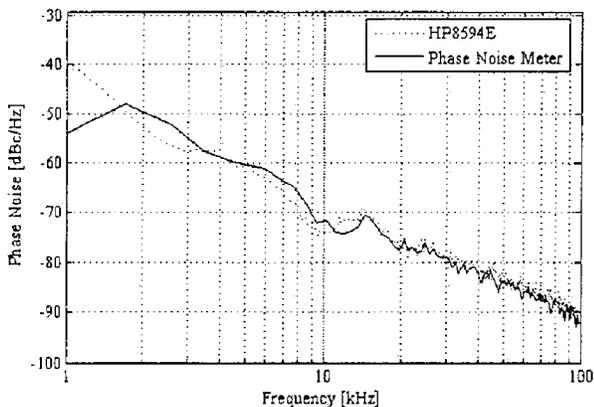


Fig.7. Portion both of phase noise power spectrum provided by the proposed meter and that measured by the spectrum analyzer HP8994E™. Very good concurrence can be noticed.

measurement result without losing information on the input signal. Thanks to suitable strategies for measured data management and processing, variable frequency resolution is assured in the final phase noise power spectrum, ranging from few units of hertz for close-to-the-carrier analysis up to fractions of kilohertz for inspection at frequency offsets very far from the carrier. Current input frequency range extends up to 250 kHz.

To assess the performance of the meter, several experiments have been carried out on real sinusoidal signals exhibiting either high spectral purity or flicker phase noise. Noise floor has resulted always lower than -120 dBc/Hz at 1 kHz frequency offset. Moreover, very good concurrence between phase noise power spectra measured by the proposed meter and those provided by a spectrum analyzer has been experienced.

On-going research activity is mainly oriented both to enlarge the input frequency range, by making operative an optimized band-pass sampling technique, and better assess the performance of the meter through the use of phase noise dedicated measurement systems.

REFERENCES

- [1] D.W. Allan, "Statistics of Atomic Frequency Standards," *Proc. IEEE*, vol. 54, February 1966, pp.221-230.
- [2] B.Hofmann-Wellenhof, H.Lichtenegger, and J.Collins, "Global Positioning System: Theory and Practice," New York: Springer-Verlag, 1994.
- [3] K.Feher, *Telecommunication, Measurement, Analysis and Instrumentation*, Prentice-Hall Inc., New Jersey, 1987.
- [4] Coombs, *Electronic Instrument Handbook*, 2nd Edition, McGraw-Hill.
- [5] S.Bregni, "Clock stability characterization and measurement in telecommunication," *IEEE Trans. Instrum. and Meas.*, vol. 46, Dec. 1997, pp.1284-1294.
- [6] *IEEE standard definitions of physical quantities for fundamental time and frequency metrology – random instabilities*, IEEE Std 1139-1999, July 1999.
- [7] NIST Time and Frequency Metrology Group, *The role of phase noise research and metrology*, <http://tf.nist.gov/timefreq/phase/index.html>
- [8] *Phase Noise Measurement Seminar*, Hewlett & Packard Literature No.5968-3734EE, 1998.
- [9] E.Rubiola, V.Giordano, "Correlation based phase noise measurement," *Review of Scientific Instruments*, vol.71, N°8, August 2000, pp.3085-3091.
- [10] L.D.Cosart, L.Peregrino, and A.Tambe, "Time domain analysis and its practical application to the measurement of phase noise and jitter," *IEEE Trans. Instrum. and Meas.*, vol. 46, Aug. 1997, pp.1016-1019.
- [11] L.Angrisani, M.D'Apuzzo, M.D'Arco, "A digital signal processing approach for phase noise measurement," *IEEE Trans. Instrum. and Meas.*, vol.50, No.4, Aug. 2001, pp.930-935.
- [12] L.Angrisani, A.Baccigalupi, M.D'Arco, "Method for close-to-the-carrier phase noise measurement", *IEE Proceedings Science Measurement and Technology*, Vol.150, n.1, January 2003, pp.35-39.
- [13] C.A.Greenhall, "Digital signal processing in the radio science stability analyzer," in TDA Progress Report 42-121, May 15, 1995.
- [14] L.Angrisani, A.Baccigalupi, M.D'Arco, "Evaluating phase noise power spectrum with variable frequency resolution," *IEEE Trans. Instrum. and Meas.*, vol.53, No.3, June 2004, pp.685-691.
- [15] Analog Devices ADSP Sharc® processor, *Preliminary technical data ADSP-21364*, Rev. PrA, January 2004, pp.1-44. www.analog.com © Analog Devices, Inc.