

Software Defined Radio Approach to Distance Measuring Equipment

Omar A. Yeste-Ojeda and René Landry, Jr.

Laboratoire Spécialisé en Systèmes Embarqués, Navigation et Avionique (LASSENA)
École de technologie supérieure
Montreal, Canada

Omar.Yeste@lassena.etsmtl.ca, ReneJr.Landry@etsmtl.ca

Abstract—This work describes the methodology used for the development of an avionics system, specifically the distance measuring equipment (DME), from the simulation phase to its implementation in a low cost embedded real time system. The approach considered is based on software defined radio (SDR), that is, all the main functions and signal processing are carried out by an embedded central processing unit (CPU). The purpose of this paper is to show the challenges, such as compatible designing, and advantages of the SDR approach to avionics systems. The laboratory test results show the feasibility of the low cost prototype attaining distance measuring accuracies better than 20 m. Basic minimum operational performance standards are met, while full compliance must be still assessed in order to obtain authorization for the next phase of the project, that is flight testing.

Keywords—Distance Measuring Equipment; Software Defined Radio; SWaP-C; Avionics; APNT

I. INTRODUCTION

In the last years, the modernization of avionic communications and radio navigation technologies has skyrocketed the interest of aviation and aerospace companies in developing flexible, compact, light, low power consumption and low cost devices, establishing commercial standards of reconfigurability whilst minimizing SWaP-C requirements: Size, Weight, Power and Cost. Software Defined Radios (SDRs) represent a step beyond in the minimization of all these requirements. They make possible the integration of almost all signal processing in software, leaving aside the problems related to hardware, significantly decrease the amount, hence the weight, of required equipment and interconnection hardware, perform at least as well as the old equipment and reduce the development, deployment and maintenance costs [1]. Moreover, a single SDR piece of equipment can be programmed to perform different functions, e.g. the Secondary Surveillance Radar (SSR) transponder, the Instrument Landing System (ILS) receiver or the Distance Measurement Equipment (DME), which this work is focused on.

Along with Global Positioning System (GPS) and the Flight Management System (FMS), DME is one of the most useful elements in air navigation. The main purpose of the DME is to display the slant range between an airplane and a ground station, by measuring the delay between the transmission of a pair of pulses by the aircraft (interrogation) and the reception of

the corresponding reply transmitted by a land-based transponder (see Fig. 1) [2]. Combined with the VHF Omnidirectional Range (VOR) system, DME provides the position of an aircraft from a single ground station [3]. According to the US Federal Navigation Plan, the operation of DME systems is planned to be continued and even expanded as an Alternate Positioning, Navigation, and Timing (APNT) system (also referred to as “DME/DME navigation”) as part of NextGen modernization [4]. This mandate has reactivated the research community interest in enhanced DME (eDME) systems, not only focused on the new APNT functionality [5]-[8], but also on the system upgrade by application of modern technologies [9]-[10].

In this work, an SDR approach and low-cost implementation of a DME system is presented. The main contributions of this paper are twofold: First, a low cost SDR implementation of the airborne DME equipment is presented. The SDR hardware chosen in this work is Nutaq’s ZeptoSDR [11], costing about 2,000 \$ per SDR unit. These units include a high quality radio module (Radio420X), a Xilinx Zynq FPGA with embedded Dual ARM Cortex-A9 processor. They allow fast prototyping in remote operation mode using a host computer, as well as embedded operation mode for final prototype. Thus, no additional equipment is required for the development of the SDA system. The performance characterization of the system and an assessment on its

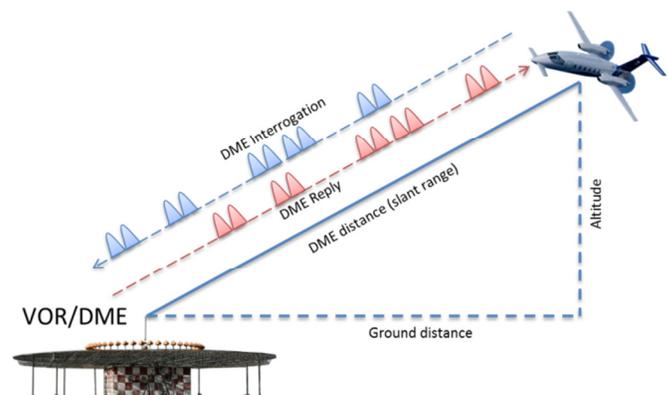


Fig. 1. DME principle of operation: The slant range is computed at the aircraft from the delay between the transmitted interrogation and the received reply.

This research project entitled “AVIO-505” was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ), and with the collaboration of Bombardier Aerospace, MDA, Marivent Corporation and Nutaq.

compliance with the currently in-effect regulation is also provided. Second, the methodology described in this paper allows almost seamless migration of the DME system from simulation environment (using MATLAB/Simulink platform) to a proof-of-concept low-cost embedded prototype. The proposed methodology can be split into three sequential phases: 1) MATLAB/Simulink design and characterization, 2) real time SDR implementation, and finally, 3) integration into an embedded system.

The purpose of this paper is to show the challenges and advantages of the SDR approach to avionic systems. For instance, the main challenge of the design is to make it compatible for all development phases. Seamless migration requires the system design during the simulation phase to be compatible with the subsequent SDR platforms. Section II is devoted to describe this first phase of the development consisting in software modulation. The second phase of the methodology is to move the MATLAB/Simulink model to the GNU Radio software platform, yielding a real time SDR implementation of the DME system in the ZeptoSDR platform. Section III describes the main challenges in this phase and the solutions adopted in this work. The final part of the paper describes the embedded SDR implementation, the in lab testing environment and the performance evaluation of the system. One of the advantages of the ZeptoSDR over other systems is that it can be operated in remote and embedded versions. Therefore, phase 2 can be rapidly developed operating the ZeptoSDR in remote mode, while the final prototype is integrated into the embedded version. Additional optimization of the computational burden is required for the system to achieve full operational capability. Section V provides and discusses some results obtained by the embedded DME system. Initial performance results from laboratory tests reveal distance measurement accuracy better than 20 m at near distances and small accuracy degradation with distance. Those results demonstrate not only the feasibility of low cost software defined avionics, but also the success of the methodology used for rapid development of SDR systems. Finally, Section VI draws the main conclusions of the paper and anticipates some of the further steps to be followed with the aim of extensive validation of the prototype including flight testing.

II. PHASE 1: DESIGN AND SIMULATION

In this work, MATLAB/Simulink software platform has been chosen for the simulation environment, while GNU Radio open source Software Development Kit (SDK) [12] is used for the real time and embedded system implementations. The flowchart concept used when developing signal processing blocks in GNU Radio is based on the generation of output samples as input samples are received. Moreover, the interface with the ZeptoSDR receiver and transmitter blocks is implemented through sample buffers. Using the same approach in the simulation model design facilitates the migration to the SDR implementation, while also minimizes the risk of software bugs producing incorrect or unexpected results. The block diagram used in the design and simulation testing of the DME system is shown in Fig. 2. All features and functionalities described in [13] were taken into account in the design and implementation of the DME system. In order to carry out more

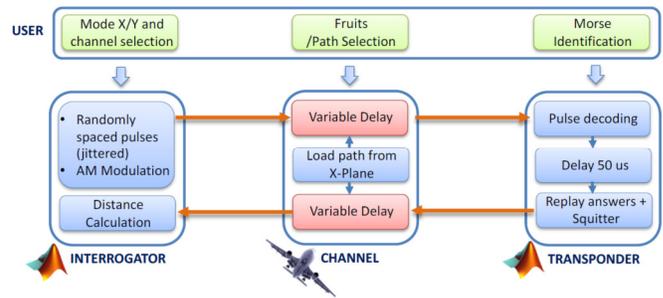


Fig. 2. Block diagram of the MATLAB/Simulink DME model.

realistic tests, the simulation software takes its input data from the commercial flight simulator X-Plane.

The main goal of Phase 1 of the methodology is to design the Software Defined Avionics (SDA) system and assess its performance in an idealized simulation environment. One of the benefits of SDR is its reconfigurability, so that there is no need to incorporate non-idealities related to the hardware in the simulation environment, as the SDA system can be easily adjusted during the subsequent phases of the process. Thus, the idea at this point is to obtain an initial design satisfying the requirements of the SDA system.

In this work, the designed SDA system is the DME. Fig. 2 represents the simulation architecture of the DME model used for design and simulation. Among other reasons, the selection of MATLAB/Simulink as the software simulation platform has been also motivated because of its capability of exporting the model to C++ code. This feature is highly interesting for our purpose since the SDR units used in the next phases are programmed in this language using the open source software development kit GNU Radio [12].

In order to facilitate the migration to GNU Radio, the SDA model should be designed following several recommendations:

1) The input and output signals of the SDA system should be the complex baseband equivalent of the RF signals. The reason is the input and output data exchanged with the SDR platform in the following phases is performed through the complex baseband equivalent of signals. The SDR platform carries out the conversion between digital and analog domains, as well as the up/down-conversion of baseband signal to the programmable carrier frequency. Fig. 3 shows the block diagram of the Radio420X module used by ZeptoSDR platform. One additional advantage of using baseband signals is that the bandwidth is drastically decreased. Thus the simulation times are reduced accordingly, accelerating the development of the SDA.

2) The model should be able to work at any specified sampling frequency. The reason for this is twofold: First, it enables the possibility of adjusting the sampling frequency according to the computational power of the embedded SDR platform. Second, though it is out of the scope of this work, by making the simulation model independent of the sampling frequency, the same model can be migrated easily to different

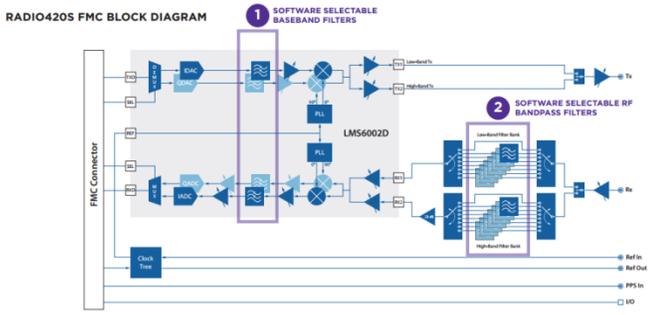


Fig. 3. Block diagram of the Radio420X module used by ZeptoSDR platform. RF signals are up/down-converted to baseband prior to digital conversion.

SDR platforms which most likely will not use the same master clock for their ADC/DACs.

3) A special care should be taken with the selection of the variable types of the different interfaces. For instance, the input and output of the ZeptoSDR platform uses interleaved (I and Q channels) short integer type.

4) It is also convenient to incorporate to the model the sample buffers used by the SDK GNU Radio and the SDR platform for the interconnection of different blocks. Thus, the functionality of the designed SDA can be validated in the presence of input/output delays.

III. PHASE 2: SDR IMPLEMENTATION

Once the simulation model has been tested and exhibits operational capability, the Simulink model can be transmuted into C++ code in order to implement the SDA into the SDR platform ZeptoSDR. Simulink software includes an option to export the simulation model into C++. Although the code generated is not ready to be used with GNU Radio SDK it saves a great deal of the programming task and provides a starting point for the SDA real time implementation.

The approach used in the simulation phase enables a rapid migration and focusing on solving the new challenges specific to real time operation. For example, the SDR implementation revealed the need to compensate the delays originated in the internal round-trip paths between the running code and the transmitting and receiving antennas. Since these delays vary any time the DME system is initialized, the solution adopted in this work is based on providing the DME prototype with auto calibration capabilities. Moreover, this solution allows for compensation of additional delays produced when connecting the DME system to different airborne equipment for flight testing.

One of the characteristic of GNU Radio is that signal processing functions and hardware definition and separated, therefore ideally any compatible SDR platform can be utilized for the development. In this work, the choice of Nutaq’s platform ZeptoSDR has been mainly motivated by the excellent characteristics of its radio module, Radio420X, such as its high dynamic range (at least 70 dB) and the short PLL setting time¹ (20 μ s). Table I summarizes the main features of this radio. The Radio420X FPGA mezzanine card (FMC) is a

multimode SDR RF transceiver module designed around the multi-standard, multiband Lime Microsystems LMS6002D RF transceiver IC, which operates in two selectable bands [14]. For DME system implementation, the low band must be chosen (see Table I) as it operates in the Aeronautical Radio Navigation Services (ARNS) band between 960 MHz and 1215 MHz.

As stated above, the code generated by Simulink must be adapted to GNU Radio SDK. A new block (out-of-tree block [15]) called “DME interrogator” is generated as a result of this process. Basically, the adaption consists of implementing the common application programming interface of GNU Radio, and separating signal processing tasks (implemented in the “general work” method) from the initialization steps (implemented in the “make” method) [15]. Once this block has been generated, the input and output ports must be connected to the real-time data exchange (RTDEx) blocks of Nutaq’s GNU Radio plugin enabling real time communication between Nutaq’s hardware and the DME system implemented in the CPU. In this Phase 2, the RTDEx blocks perform the full duplex transmission of data between the ZeptoSDR and the host computer using an Ethernet connection. The GNU Radio Companion flowchart used for setting up the DME module and interconnecting the different blocks is shown in Fig. 4.

GNU Radio Companion is used for creating the python

TABLE I NUTAQ’S RADIO420X TRANSCEIVER SPECIFICATIONS

Parameter		Value
RF range:	Low band	300-1600 MHz
	High band	1500-3800 MHz
PLL setting time		20 μ s
Wideband noise floor:	Receiver	-100 dBc/Hz
	Transmitter	-124 dBc/Hz
Sensitivity (SNR = 5 dB, BW = 200 KHz)	Low band	-103 dBm
	High band	-90 dBm
Gain control	Receiver (low band)	79 dB
	Receiver (high band)	73 dB
	Transmitter	70 dB
Maximum input power		-13 dBm
IMD3	Receiver (low band)	-61 to -56 dBc
	Receiver (high band)	-50 to -45 dBc
	Transmitter	-60 dBc
SFDR		50 dBc
P1dB output	Low band	20 dBm
	High band	15 dBm
Carrier suppression		-50 dBc
Side band suppression		-45 dBc

¹ The requirement of short PLL setting time is motivated by future development of new related applications, such as DME/DME navigation using a single radio.

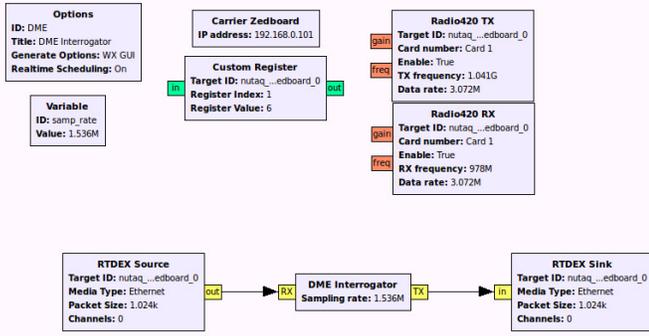


Fig. 4. GNU Radio Companion flowchart used for generating the executable python code interconnecting the out-of-tree DME interrogator module with RTDEX Nutaq’s blocks.

executable file running the DME system. The python interfaces for C++ classes (blocks) are generated using SWIG [16]. After generation of this python executable, there is no need to use GNU Radio Companion any more, as the python code can be directly edited as required. For instance, several low rate functions can be implemented in the python executable, such as block reconfiguration, channel switch, range display or as seen next, DME calibration step.

After the DME interrogator block is finalized and the flowchart is fully functional, the real time DME system is validated using the test equipment IFR 6000 developed by Aeroflex [17], which is operated manually. The IFR 6000 is a unit designed for testing MODE A/C/S transponder, DME, TCAS, ADS-B and TIS avionics systems. The first challenge observed is the existence of a long delay between the transmission and reception channel of the SDR implementation of DME. Specifically, the delay measured between the output of a pair of pulses and the reception of its correspondent reply from the test equipment is in the order of several milliseconds. This delay is produced by several buffers in the transmission and reception of data such as those implemented in the Ethernet physical layer, the FPGA of the ZeptoSDR, etc. As a consequence, not only the measured distance is biased, but the replies cannot be processed as they fall outside the temporal window of 2.53 ms opened right after the transmission of a pair of pulses [13].

In addition to that, the delay changes every time the DME system starts running in the host computer. Therefore, an auto-calibration procedure is carried out during the setup of the system. The functional flowchart of the DME system becomes as shown in Fig. 5. During the initialization step, several configurable parameters are defined, such as the sampling frequency, the gains of the transmitter and receiver chains, the DME channel, etc. After that, the Radio420X module requires a calibration step for adjusting the gains and compute PLL parameters. During this step, it is recommended that the transmission and reception frequencies are separate enough, which is satisfied by the DME system (63 MHz apart). After the Radio420X calibration step, the DME auto-calibration procedure starts. The purpose of the calibration step is twofold: On the one hand, it allows delaying the opening of the temporal window in order to enable the reception of the replies from a

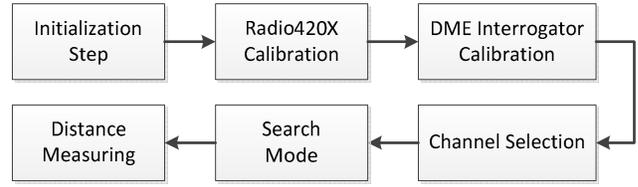


Fig. 5. Functional flowchart of the DME SDR implementation. The auto-calibration of the DME interrogator allows compensating for the delays introduced by the transmitter and receiver chain as well as for the biases in the distance measurements.

DME ground station. On the other hand, the delay measured during the calibration step is subtracted from the delay measurements leading to accurate distance measuring. Once DME calibration is completed, the transmission and reception frequencies are reconfigured to the values corresponding to the selected DME channel and the DME system operates normally, first in search mode, and secondly, measuring distances when it enters into track mode.

Basically, the auto-calibration procedure consists of measuring the delay between the transmission of a pair of pulses and the reception of that pair of pulses by the DME system. In other words, the DME system interrogates itself at a given frequency in order to determine the round-trip delay time of the transmission and reception chains. The auto-calibration is performed using one of the carrier frequencies allocated for DME (specifically, 978 MHz). Therefore, the DME interrogator uses random delays between pair of pulses so that its own interrogations can be distinguished from any other reply that may be received from any near DME ground station. During the auto-calibration, the DME system operates in Mode X, since this mode preserves the pulse spacing (12 μ s) in the interrogation and replies. Otherwise, using a different mode, such as Mode Y, will cause the receiver not to recognize as valid replies the interrogation sent by the transmitter, since the pulse spacing is 30 and 36 μ s respectively. The last point to be considered during the calibration step is related to the amplification gains of the transmitter and receiver chains. In normal operation, the receiver is set to maximum sensitivity and the transmitter to maximum output power. Using this configuration during the calibration step will saturate the receiver and the calibrated delay measurement will be perturbed, which will affect the accuracy of the system. This is due to the definition of time-of-arrival (TOA) as the point where the pulse amplitude is half the maximum amplitude. Therefore, both amplification gains must be adjusted during the auto-calibration step to prevent the saturation of the receiver. Once the calibration is finished, these gains are set back to their normal operation values. Note that the values used during the auto-calibration must be adjusted according to the equipment being used, e.g. they vary from lab testing to flight testing. The main factor to be taken into account is related to the separation between the transmitter and receiver antennas.

IV. PHASE 3: EMBEDDED SDR

The final phase of the development starts after validating basic operational capabilities of the real time SDR implementation using a host computer. The objective is to

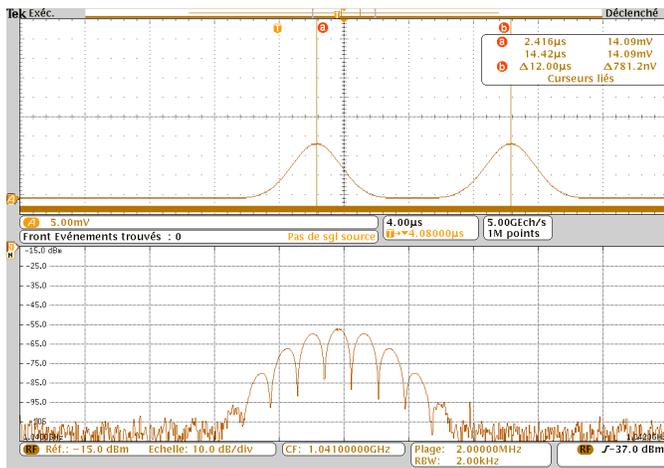


Fig. 6. Waveform and spectrum measured at the output of the embedded SDR implementation of the DME system (operating at channel 17X). Both plots show full compliance with the minimum operational performance standards.

embed the DME system into the internal processor of the SDR platform ZeptoSDR. This platform integrates a Dual Arm Cortex-A9 under OS Linux Linaro compatible with GNU Radio SDK. Therefore, the migration of the DME system from the host computer towards the embedded processor is completely seamless, and it only needs to reconfigure the interfaces of the RTDEx blocks (this reconfiguration affects only the initialization steps).

Nonetheless, the main issue faced during this phase concerns the computational capabilities of the embedded processor. Compared to Phase 2, the computational power of the embedded CPU is quite limited. For that reason, some minor modifications have to be done in order to minimize the computational burden. The goal is to obtain a functional DME system capable of operate at a sampling frequency as high as possible, since simulation results showed that the higher the sampling rate, the better performance in terms of accuracy. With the optimization, the embedded DME system is functional when operating at sampling frequencies up to 2 MHz. This limitation is not such that the accuracy of the system is severely compromised, as it is described in the next section. Moreover, the accuracy attained is much better than the required by the performance standards [13].

As a final remark, it is noteworthy that Phase 2 of the development could have been omitted and to jump directly from Phase 1 to Phase 3. Nonetheless, the motivation for completing all the phases of the methodology is that first, the development is faster (e.g. in terms of compilation time) when using the remote version of the system; and second, using a host computer eliminates the issues caused by limited computational power and thus facilitates the identification of other problems.

V. RESULTS

All the results described in this section correspond to the embedded version of the DME system. The spectral characterization of the transmitted waveform is carried using standard testing equipment in order to assess the compliance

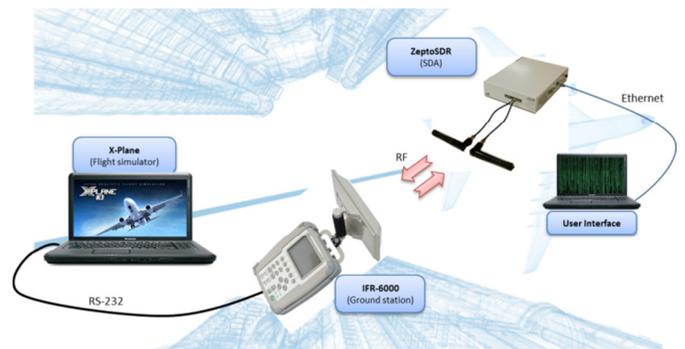


Fig. 7. Experimental setup used for testing the DME prototype. The flight simulation software X-Plane is used to remotely program the parameters of the IFR 6000 and generate more realistic testing conditions for the lab testing.

with current regulation. Fig.6 shows the envelope and spectrum of a pair of pulses transmitted using a carrier frequency of 1041 MHz (DME system operating at channel 17X) captured through a scope. In this plot, the DME system was operating at a sampling frequency of 1.536 MHz. It can be seen clearly the Gaussian shape of the two pulses and a pulse spacing of 12 μ s corresponding to operation in Mode X as specified in [13]. The duration of the pulses, defined by the time between the points at which each pulse reaches half its maximum amplitude, is 3.59 μ s. This satisfies the specification in [13], which requires a pulse duration of 3.5 μ s \pm 0.5 μ s. In addition, the transmitted spectrum has been also validated with a spectrum analyzer. No spurious signal or non-ideality has been appreciated. In addition, more thorough tests are currently being carried out in order to assess electromagnetic compatibility before applying for flight test authorization.

Fig. 7 describes the experimental setup used for assessing the performance of the embedded version of the DME system. The experimental setup used for lab testing consists of:

- 1) A laptop running flight simulator X-Plane to generate realistic data, which is used to program,
- 2) An Aeroflex IFR-6000 DME test equipment, to simulate ground station and propagation channel,
- 3) A laptop connected to the ZeptoSDR embedded SDA through Ethernet, only acting as the user interface of the DME system.

Fig. 8 shows the ranging accuracy of the DME as a function of the SNR (measured after the Analog-to-Digital Converter) and several values of the sampling frequency. For lower SNR values, the DME system is not able to track the DME ground station. For higher sampling frequency values, the embedded CPU is not able to generate the output samples at the required rate. This produces a loss of synchronism between the reception and transmission channels. As a consequence, the delay measured during the auto-calibration procedure performed at the initialization becomes invalid.

The performance results reveal distance measurement accuracy better than 20 m for moderate and high SNR. Actually, the specifications of the testing equipment IFR 6000 guarantee that the replies to the SDR interrogation are generated with a range delay that is accurate within ± 18.52 m

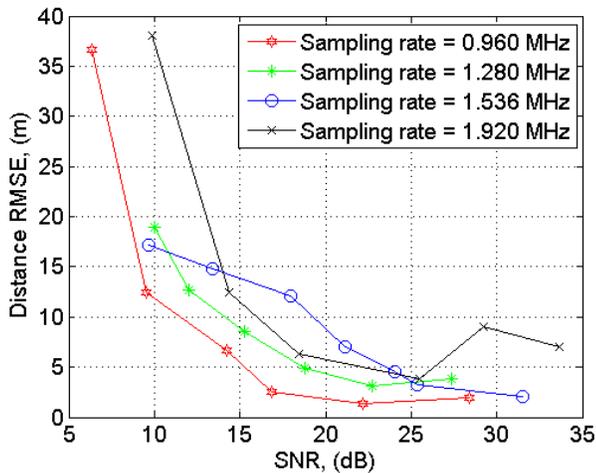


Fig. 8. Experimental ranging accuracy of the DME system embedded into the ZeptoSDR platform, as a function of the SNR and several sampling frequency values.

(± 0.01 nmi) [17]. This accuracy is small enough to assess DME compliance with minimum operational performance standards (MOPS), which require an airborne DME system to achieve a ranging accuracy better than ± 0.17 nmi (on a 95% probability basis) in the most restrictive case [13]. The accuracy of the developed prototype is at least one order of magnitude better than required by MOPS. However, trying to infer any conclusion from the curves shown in Fig. 8, such as the effect of SNR or sampling rate on system's accuracy, is not possible: It cannot be determined how much of the error is due to the SDR DME system and how much to the IFR 6000.

VI. CONCLUSIONS

This work demonstrates the feasibility of implementing SDA systems in low cost SDR platforms. Besides all the benefits obtained from the use of SDR, such as SWAP-C optimization, reconfigurability, etc., it is also remarkable from the point of view of prototyping that no hardware component has had to be designed for the realization of this work. The methodology used consists of three phases: simulation, real-time and embedded implementations. Following some recommendations, this methodology has allowed developing the DME system in short time, and implementing it in an embedded SDR platform with almost no difficulties. The most relevant challenge that has had to be faced is related to timing issues, as the significant delays of the transmission and reception channels due to communications buffers exhibit some randomness that was not taken into account during the simulation phase.

Lab testing results has shown that the accuracy of the system is much better than that required by MOPS, beyond some common testing equipment specifications, even at the lowest sampling rates. Other performance requirements, such the waveform and spectrum compliance, maximum time to start tracking, etc., are also satisfied by the low cost prototype developed.

In the next phases of the development, it is planned to test the prototype in field, using real a DME ground station. After obtaining the required authorizations, it is programmed to install the equipment in a real airplane. Additional testing and validation will be carried out, with the ultimate goal of demonstrate the operability of the DME system through flight test missions.

REFERENCES

- [1] J. Mitola, "Software Radio," in Encyclopedia of Telecommunications, John Wiley & Sons, Inc., 2003.
- [2] A. D. Helfrick, Principles of Avionics, Avionics Communications, Leesburg, VA, 2004.
- [3] Aeronautical Information Manual. Official Guide to Basic Flight Information and ATC Procedures, Federal Aviation Administration, February 9, 2012.
- [4] 2010 Federal Radionavigation Plan, Department of Defense, Department of Homeland Security and Department of Transportation, 2010.
- [5] Lo, Sherman C., Enge, Per, "Assessing the Capability of Distance Measuring Equipment (DME) to Support Future Air Traffic Capacity," NAVIGATION, Journal of The Institute of Navigation, Vol. 59, No. 4, Winter 2012, pp. 249-261.
- [6] Kim, E., "Investigation of APNT Optimized DME/DME Network Using Current State-of-Art DMEs: Ground Station Network, Accuracy, and Capacity," Proceedings of IEEE/ION PLANS 2012, Myrtle Beach, South Carolina, April 2012, pp. 146-157.
- [7] Kim, O-J., Kim, C., Song, J., Yun, H., Kim, D., Kee, C., "A New Concept of APNT: MOSAIC/DME 3-D Positioning with a Single DME Station," Proceedings of the 2012 International Technical Meeting of The Institute of Navigation, Newport Beach, CA, January 2012, pp. 142-150.
- [8] Pelgrum, Wouter, Li, Kuangmin, Smearcheck, Matt, van Graas, Frank, "eDME Architecture Development and Flight-Test Evaluation," Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012), Nashville, TN, September 2012, pp. 812-825.
- [9] Li, Kuangmin, Pelgrum, Wouter, "Enhanced DME Carrier Phase: Concepts, Implementation, and Flight-test Results," NAVIGATION, Journal of The Institute of Navigation, Vol. 60, No. 3, Fall 2013, pp. 209-220.
- [10] Naab-Levy, A., Li, K., Pelgrum, W., "DME/N Error Budget Allocation and DME-Next Proof-of-Concept Flight Test and Performance Evaluation," Proceedings of the ION 2013 Pacific PNT Meeting, Honolulu, Hawaii, April 2013, pp. 434-450.
- [11] Nutaq, "ZeptoSDR," Available: <http://nutaq.com/en/products/zeptosdr>.
- [12] GNU Radio - WikiStart [Online]. Available: <http://www.gnuradio.org>
- [13] Minimum Operational Performance Standards for Airborne Distance Measuring Equipment (DME) Operating Within the Radio Frequency Range of 960-1215 Megahertz, RTCA Standard DO-189, September 20, 1985.
- [14] Nutaq, "Radio420X," Available: <http://nutaq.com/en/products/radio420x>.
- [15] GNURadio, "Out-of-tree modules," Available: <http://gnuradio.org/redmine/projects/gnuradio/wiki/OutOfTreeModules>.
- [16] "Welcome to SWIG," Available: <http://www.swig.org/>.
- [17] "IFR 6000", Available: http://www.aeroflex.com/ats/products/product/Avionics/XPDR_INTER_DME_TACAN_TCAS/IFR_6000~190.html.