Rapid Prototyping and FPGA-in-the-Loop Verification of a DFrFT-based OFDM System

Atul Kumar and Maurizio Magarini
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, 20133 Milano, Italy
{atul.kumar, maurizio.magarini}@polimi.it

Stefano Olivieri
MathWorks Italia
Torino, Italy
stefano.olivieri@mathworks.it

Abstract—**Orthogonal frequency division multiplexing (OFDM) based on the use of discrete fractional Fourier transform (DFrFT) has recently gained interest due to its lower sensitivity to synchronization errors in comparison with conventional OFDM based on the use of the discrete Fourier transform (DFT). Although this higher robustness to synchronization errors is a well-recognized fact, only few works are available in the literature that concern with DFrFT hardware implementation. In this work, we consider its implementation in a Field Programmable Gate Array (FPGA). To verify the design of the DFrFT-based OFDM system, we use FPGA-in-the-Loop (FIL) co-simulation method to evaluate bit error rate (BER) in presence of carrier frequency offset (CFO) when transmission takes place over a frequency selective Rayleigh fading channel.**

***Keywords*- OFDM; FPGA-in-the-Loop co-simulation; Rapid prototyping; carrier frequency offset (CFO); discrete fractional Fourier transform (DFrFT).**

I. Introduction

Orthogonal frequency-division multiplexing (OFDM) based on discrete Fourier transform (DFT) has drawn major attention in wireless communication due to its various advantages like efficient bandwidth utilization, high data rate, less complex equalization, robustness against multi-path fading channel, etc. [1]. Hence, is has been adopted in many wireless communication standards, such as IEEE 802.11a, IEEE802.16a, LTE, LTE-Advanced, and terrestrial digital video broadcasting system [2].

In spite of several advantages, OFDM also presents some disadvantages, among which a prominent one is high sensitivity to synchronization errors, *i.e.* symbol timing offset and carrier frequency offset (CFO) [3, 4]. The presence of CFO depends on several causes: Doppler spread, phase noise, and mismatching of transmitter and receiver oscillators' frequencies. The main negative effect of CFO is the introduction of inter-carrier interference (ICI) which, basically, consists in a loss of orthogonality between sub-carriers.

Nowadays, discrete fractional Fourier transform (DFrFT) is emerging as an efficient tool for performing time-frequency analysis in many fields of digital signal processing [1]. DFrFT is a generalization of the DFT. There are many papers available in the literature that deal with the study of DFrFT-based OFDM systems [5]-[10]. The main motivation behind use of DFrFT in OFDM system is its higher robustness to synchronization errors compared to DFT, especially CFO.

A careful survey of the literature reveals that many discrete versions of DFrFT have been so far proposed [10]. All the proposed versions fall into four categories: linear combination-based method, sampling-based method, weighted summation-based method, and eigenvector decomposition-based method. It is shown in [10] that when the length N of the block of samples used in DFrFT computation is a power of 2, the complexity of sampling-method and linear-based method is in the order of $O(N \log_2 N)$, which is the same as that of fast Fourier transform (FFT), the efficient implementation of DFT. In contrast, for weighted summation-based method and for eigenvector decomposition-based method the complexity is in the order of $O(N^2)$. As shown in Table I of [10], complexity of different DFrFT algorithms depends on the constraints that are set on the discrete implementations. As far as the use of DFrFT in an OFDM system is concerned, the most important constraint is reversibility for which the inverse DFrFT (IDFrFT) satisfies Hermitian property and, therefore, allows us to invert the direct transform operation done by DFrFT. For this reason, we focus here on the closed-form type form of sampling-based method since it ensures reversibility with the lowest possible complexity.

The main contribution of this work is a software defined radio (SDR) implementation of the closed-form type of sampling-based DFrFT and its hardware co-simulation in an OFDM system. We present an FPGA prototyping of the OFDM receiver based on DFrFT by using the model based design flow for the Xilinx ZedBoard, equipped with a Zynq-7000 FPGA family SoC device. The realized hardware design is tested by using FPGA-in-the-Loop (FIL) co-simulation methodology. The bit error rate (BER) of the DFrFT-based OFDM system is evaluated in case of transmission over a frequency selective Rayleigh fading channel in presence of CFO. Zero forcing equalization of the received signal is performed in the receiver. The SDR model is built by using the Toolboxes provided by MATLAB and Simulink.

The paper is organized as follows. Section II describes the DFrFT-based OFDM system. The FIL co-simulation set-up is described in Sec. III. The details about implementation results are given in Sec. IV. Section V concludes the paper.

II. Description of the DFrFT-based OFDM System

The block diagram of the DFrFT-based OFDM system is shown in Fig. 1. Starting from the encoder block, a high data rate stream is split into a number of low data rate streams

ISBN 978-0-9928626-7-1 © EURASIP 2017

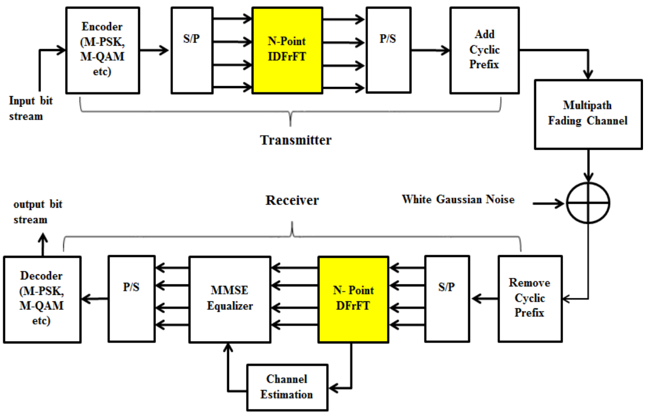

Fig. 1: System model of the DFrFT-based OFDM system.

that are first applied to a serial to parallel (S/P) converter and then transmitted in parallel using *N* sub-carriers. After S/P conversion, the diagram shows an OFDM modulator based on the sampling-based method of inverse IDFrFT kernel, where the information block contains *N* symbols. To mitigate the effect of inter-symbol interference, caused by channel time spread, a cyclic prefix (CP) is inserted between two successive OFDM symbols. After that, we consider transmission over a frequency selective Rayleigh multi-path fading channel where additive white Gaussian noise (AWGN) is added to the received signal. By assuming the perfect knowledge about the OFDM symbol start at the receiver, CP is removed and the resulting samples are S/P converted. After this, the DFrFT is used to process the samples and the resulting symbols are first equalized and then again serialized. The equalized samples are finally decoded to obtain the originally transmitted data.

A. IDFrFT kernel at the transmitter

The block of *N* transmitted symbols is applied to the input of the IDFrFT kernel. After *N*-point IDFrFT computation, the expression of the *m*-th transmitted sample is written as

$$x(m) = \sum_{k=0}^{N-1} X(k)\, F_{-\alpha}(m,k), \quad m = 0,1,\dots,N-1, \qquad (1)$$

where $X(k)$ is the symbol transmitted on the *k*-th subcarrier and $F_{-\alpha}(m,k)$ is the IDFrFT kernel given by

$$F_{-\alpha}(m,k)$$
$$= \underbrace{\sqrt{\frac{\sin(\alpha)+j\cos(\alpha)}{N}}}_{\text{Constant\_A}}\ \underbrace{e^{\frac{-jm^2T_s^2\cot(\alpha)}{2}}}_{\substack{\text{1st\_Term}\\ \text{Kernel}}}\ \underbrace{e^{\frac{-jk^2u^2\cot(\alpha)}{2}}}_{\substack{\text{2nd\_Term}\\ \text{Kernel}}}\ \underbrace{e^{\frac{j2\pi mk}{N}}}_{\substack{\text{3rd\_Term}\\ \text{Kernel}}}, (2)$$

where T_s and u are the sampling intervals in the time and in the fractional Fourier domain, respectively, which are related as $u \times T_s = 2\pi|\sin(\alpha)|/N$. The fractional Fourier domain makes an angle $\alpha = a \times \pi/2$ with the time-domain, with $0 \leq a \leq 1$. For $\alpha = \pi/2$, *i.e.*, $a=1$, DFrFT converts into its DFT counterpart.

In order to get an efficient receiver implementation, a full fixed-point data type FPGA prototyping is addressed in this paper. The IDFrFT kernel given in (2) has many floating-point operations: square root, exponential and trigonometric functions. There are many hardware efficient algorithms for representing a floating-point function into its fixed-point form. Among these, the set of shift-add algorithms, known as CORDIC, can be employed for computing a wide range of trigonometric, hyperbolic, linear, and logarithmic functions [11]. The IDFrFT kernel contains the following floating-point functions: square root, exponential, sin(·), cos(·), and cot(·). First, we need to convert into fixed-point functions using the CORDIC algorithms. While the CORDIC implementations of sin(·), cos(·), and exp(·) functions are already available, there is no direct implementation available for cot(·), which is present in the exponent of (2). By using the trigonometric property of cot(·)=cos(·)/sin(·), implementation becomes possible. After the implementation of CORDIC algorithms, the Simulink model of the IDFrFT kernel is shown in Fig. 2. Each factor in the mathematical expression of the IDFrFT kernel in eq. (2) is reported as a sub-block in the Simulink model. It is worth observing that HDL Coder will be used in the following to automatically generate the VHDL code for the FPGA. As given in eq. (2), the IDFrFT kernel has two variables, m, k, both taking values from 0 to $N-1$. So, after computation of "1st_Term" and "2nd_Term" of the kernel, their product is obtained as matrix multiplication. This is because HDL Coder does not support matrix multiplication operation. Hence, to perform the required "Matrix Multiplication", we had to develop our own block implementing elementwise multiplication.

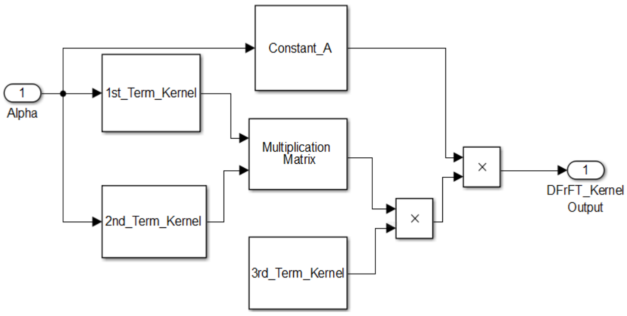

Fig. 2. Simulink model of the implemented IDFrFT and DFrFT kernels.

B. DFrFT kernel at the receiver

At the receiving side, after DFrFT computation, the received signal on the q-th subcarrier is

$$y(q) = \sum_{n=0}^{N-1} r(n)\ F_{\alpha}(q,n), \qquad q = 0,1,\dots,N-1, \qquad (3)$$

where $r(n)$ is the received signal before the DFrFT block and $F_{\alpha}(q,n)$ is the kernel given by

$$F_{\alpha}(q,n)$$
$$= \underbrace{\sqrt{\frac{\sin(\alpha)-j\cos(\alpha)}{N}}}_{\text{Constant\_A}}\ \underbrace{e^{\frac{jn^2T_s^2\cot(\alpha)}{2}}}_{\substack{\text{1st\_Term}\\ \text{Kernel}}}\ \underbrace{e^{\frac{jq^2u^2\cot(\alpha)}{2}}}_{\substack{\text{2nd\_Term}\\ \text{Kernel}}}\ \underbrace{e^{\frac{-j2\pi nq}{N}}}_{\substack{\text{3rd\_Term}\\ \text{Kernel}}}. (4)$$

The Hermitian property of the DFrFT kernel makes almost identical to that of IDFrFT and, therefore, the same considerations done in Sec. III.A for the IDFrFT kernel hold for the DFrFT kernel as well.

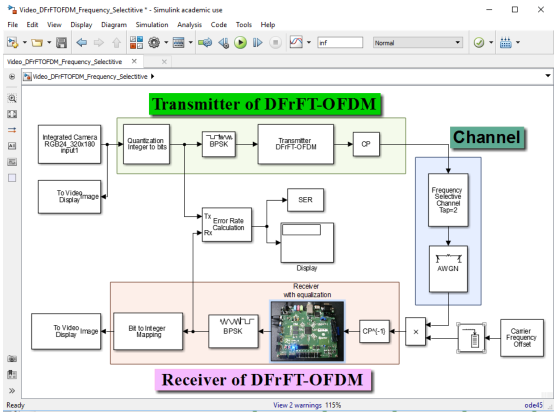
Video_DFrFTOFDM_Frequency_Selectitive * - Simulink academic use
File Edit View Display Diagram Simulation Analysis Code Tools Help
inf
Normal
Video_DFrFTOFDM_Frequency_Selectitive
Transmitter of DFrFT-OFDM
Integrated Camera RGB24_320x180 input1
Quantization Integer to bits
BPSK
Transmitter DFrFT-OFDM
CP
Channel
Frequency Selective Channel Tap=2
AWGN
To Video Display Image
SER
Tx Error Rate Calculation Rx
Display
Receiver with equalization
To Video Display Image
Bit to Integer Mapping
BPSK
CP^{-1}
×
Carrier Frequency Offset
Receiver of DFrFT-OFDM
Ready
View 2 warnings 115%
ode45

Fig. 3. Simulink model of the FIL receiver co-simulation of the DFrFT-based OFDM system.

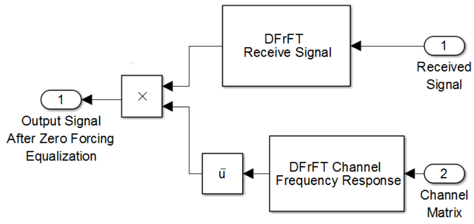

Fig. 4. Simulink model of receiver with zero-forcing equalization.

III. FPGA-in-the-Loop co-simulation of Receiver with equalization of DFrFT-based OFDM

As an example of Simulink model that implements a DFrFT-based OFDM system, we consider the case of transmission of a video signal as shown in Fig. 3. The video signal is captured from a generic webcam and FIL co-simulation is run both to verify the correctness of the FPGA implementation and to accelerate the simulation at the receiver [12-14]. Since the receiver of the DFrFT-based system includes also zero forcing equalization, its overall complexity turns out to be higher than that at the transmitter and, therefore, the possibility of performing its rapid is more challenging. The Simulink setup of the receiver with zero forcing equalization of the DFrFT-based OFDM system is given in Fig. 4. As a figure of merit we analyse the quality of the reproduced video by comparing the output of DFrFT-based OFDM system with that of the conventional one based on the DFT for different CFO [15].

A. Receiver with zero-forcing equalization of the DFrFT-based OFDM system

Zero forcing equalization is implemented by assuming ideal knowledge of the channel. We compute the frequency response of the channel and use it to divide sub-carrier by sub-carrier the received signal after the DFrFT block.

B. Floating-point to fixed-point conversion

The FPGA implementation requires conversion from floating-point to fixed-point data type. However, conversion from floating-point to fixed-point is very challenging and time consuming, typically demanding from 25 to 50% of the total design and implementation time. The conversion process introduces quantization errors that depend on the word and the fractional lengths that, in turn, impacts on the FPGA hardware resources. The optimization of the word length is an iterative process that is carried out using the Fixed Point Toolbox, available in Simulink, with the goal of achieving the best possible performance.

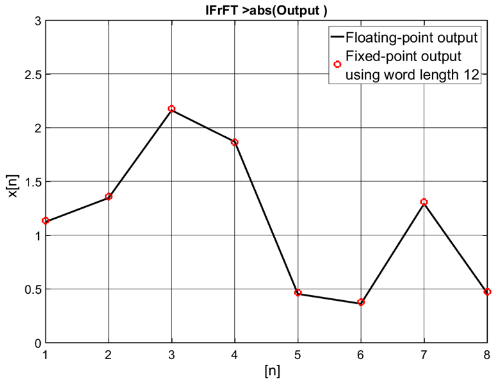

Fig. 5. Comparing the floating-point and fixed-point output of fixed-point converted IDFrFT kernel by applying the random input.

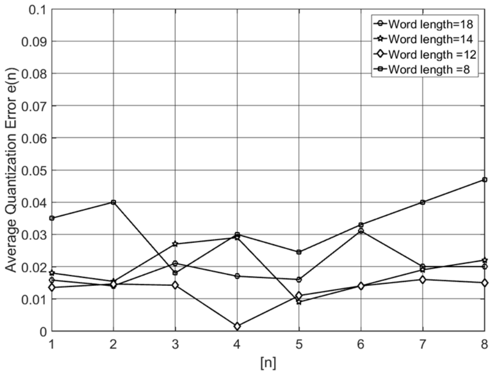

Fig. 6. Average absolute quantization error versus index of the sample for different value of the word length for IDFrFT fixed-point block.

Here, we compare floating-point and fixed-point results for the implementation of the IDFrFT/DFrFT kernels. Figure 5 shows an example of the output of the IDFrFT kernel for floating-point and for fixed-point with word length of 12 bits. Steps for the design are repeated serval times using different random inputs to average over different fixed-point outputs. In Fig. 5 the average quantization error due to the conversion from floating-point to fixed-point for different values of word length is shown. The average quantization errors obtained for different values of the word length are shown in Fig. 6. Similarly, the results for floating- and fixed-point outputs of the fixed-point DFrFT kernel are shown in Fig. 7. Those for the average quantization error obtained for different values of the word length are shown in Fig. 8.

The average quantization error $e_c(n)$ for fixed-point representation with codeword length of c bits is defined as

$$e_c(n) = \frac{1}{B}\sum_{i=1}^{B}\left|e_{c,i}(n)\right|, \qquad n = 1, \ldots, N, \qquad (5)$$

where B is the number of input blocks containing the N samples to be transformed and $e_{c,i}(n)$ is the quantization error the i-th input block given by

ISBN 978-0-9928626-7-1 © EURASIP 2017

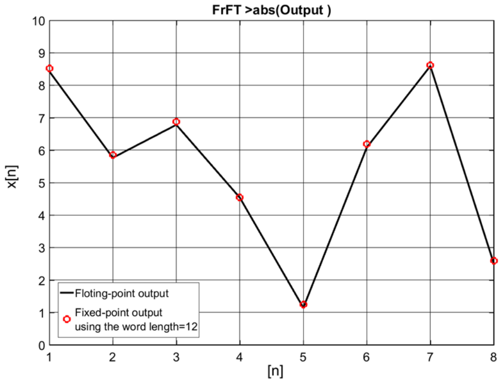

Fig. 7. Comparing the floating and fixed point output of fixed-point converted DFrFT kernel by applying the random input.

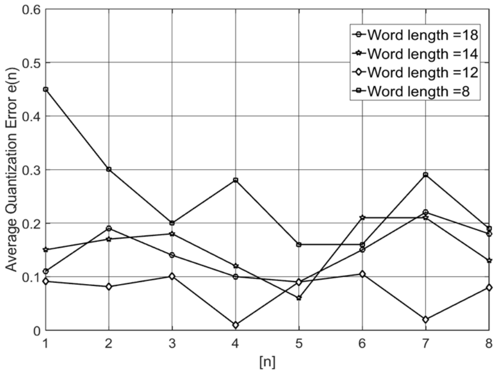

Fig.8. Average absolute quantization error versus index of the sample for different value of the word length for DFrFT fixed-point block.

$$e_{c,i}(n) = \left| |floting\ point_i(n)| - |fixed\ point_{c,i}(n)| \right| \quad (6)$$

with i=1,…,B.

Based on the definition of $e_c(n)$, we consider the fixed-point implementation of the model by selecting the proper word length values from Table 1 and then we perform the floating-point and fixed-point simulation of the model and compare the results of IDFrFT/DFrFT kernels, as reported in Figs. 5 and 7, respectively. The computation of the average quantization error using eq. (5) is shown in Figs. 6 and 8. After computation of the average quantization error, we have also computed the maximum percentage quantization error (MPQE) as

$$\text{MPQE} = 100 \times \frac{|TE|}{MAE}, \quad (7)$$

where top error (TE) = max (maximum positive error, maximum negative error) and maximum absolute value MAE = max (abs (floating-point value)).

Based on the definition of MPQE, we simulated the model by applying four different inputs as a test bench into our IDFrFT kernel. The computation results of MPQE for each input is given in the corresponding column of Table 1. We also performed the average of MPQE over the different inputs for fixed word lengths. Results for the average MPQE are reported in Table 1 from which it is clear that our system performs better if we select word length =12.

Table 1: Computation of max percentage quantization error on the different values of the word length.

update the fractional length for specified word length		Fixed-point implementation of the IDFrFT kernel			
Word length	Test Input	TE	MAE	MPQE	*Average MPQE over input*
18	1st	0.0392-0.0129i	1.451	2.8518	2.31
	2nd	-0.0019+0.007i	1.0236	0.711	
	3rd	$0.0352 - 0.0285i$	1.579	2.8514	
	4th	$0.0144 - 0.0268i$	1.0645	2.85	
14	1st	$0.0395 - 0.0126i$	1.451	2.851	2.28
	2nd	$-0.0014 + 0.006i$	1.0236	0.693	
	3rd	$0.0355 - 0.028i$	1.579	2.866	
	4th	$0.0143 - 0.0269i$	1.0645	2.867	
12	1st	$0.0364 - 0.0126i$	1.451	2.6	1.62
	2nd	$-0.004 + 0.005i$	1.0236	0.65	
	3rd	$0.038 - 0.027i$	1.579	2.95	
	4th	$0.0039 + .003i$	1.0645	2.914	
8	1st	$0.0550 + 0.0412i$	1.451	4.73	3.28
	2nd	$0.0018 + 0.038i$	1.0236	4.15	
	3rd	$0.0318 + 0.05i$	1.579	3.9	
	4th	$-0.019 - 0.025i$	8.5	0.37	

C. Code Conversion (from. mdl and .m file to VHDL)

After fixed-point implementation of the kernels, we have used HDL Coder to generate the VHDL code. The HDL workflow advisor available in HDL Coder guides through the conversion of the Simulink model to VHDL code.

D. FPGA-in-the-Loop co-simulation

For verification purposes, HDL Verifier was used to run FIL co-simulation of the DFrFT-based OFDM receiver along with equalization. Our goal was to run both the receiver and the equalizer on the FPGA board, in order to increase the simulation speed as well as get rapid development of these algorithms in FPGA. Validation of the FIL co-simulation was done through a comparison with the theoretical analysis given in [6].

IV. Implementation Results

First, the correctness of DFrFT-based OFDM system is verified by substituting $\alpha=\pi/2$ in order to get conventional OFDM-based DFT. An 8-point implementation of the DFrFT-based OFDM is considered. Transmission takes place over a multi-path Rayleigh fading channel with 2-tap equal power delay profile in presence of a normalized carrier frequency offset $\varepsilon = 0.1$. As shown in Fig. 9, BER performance of the DFrFT-based OFDM system performs better than the one based on DFT in presence of CFO. In the Monte Carlo simulation of DFrFT-based OFDM system, implementation of the IDFrFT/DFrFT block is given by the floating-point model.

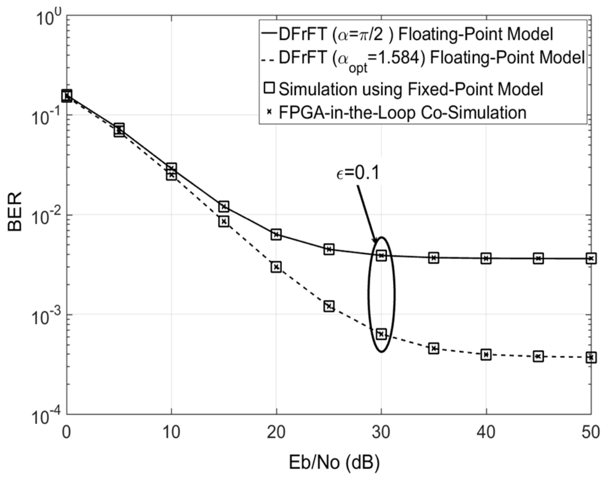

Fig.9. BER comparison for DFT- and DFrFT-based OFDM BPSK transmission over a 2-tap equal power delay profile frequency selective Rayleigh fading channel at CFO = 0.1.

Considering the conversion of the floating-point Simulink model of IDFrFT/DFrFT kernel into fixed point, we have optimized the fixed-point model by using the proper value of the word length based on what reported in Table 1. We also computed the quantization error from floating to fixed-point for given word lengths. Figure 5 shows the results for floating and for fixed-point output of IDFrFT kernel at word length of 12. The computation of the quantization error was done for different values of the word length, so that we were able to analyze the impact of word length on the quantization conversion error as given in Fig. 6. Similarly, results for the DFrFT kernel output and for the quantization error are given in Fig. 7 and Fig. 8, respectively.

A summary of the Average max percentage quantization error for different inputs at different word lengths is reported in Table 1. After fixed-point conversion, we run simulations based on the fixed-point model. In Fig. 9, a perfect match is observed with Monte Carlo simulations (marked as □). The figure reports BER versus versus signal-to-noise ratio per bit E_b/N_0, where E_b is the received signal energy per bit and N_0 is the power spectral density of AWGN. After that, the VHDL code of the kernels was generated and optimized. In order to validate our design on the hardware, we generated the bit stream for programming the FPGA on the ZedBoard using Xilinx ISE design suite, that is integrated in HDL Coder. ZedBoard is a complete prototyping development kit for Xilinx Zynq®-7000 all programmable SoC family. Xilinx ISE compiles and generates the bit stream file which is then loaded into the FPGA using JTAG via the USB connection. After generation of the bit stream, we were able to run the FIL co-simulation and verify the correctness of our model. A Perfect match is observed between FIL co-simulation (marked as ×), compared with the fixed-point implementation (marked as □) and also with the Monte Carlo simulation shown in the Fig. 9.

V. Conclusion

In this paper rapid prototyping of the receiver for a DFrFT-based OFDM system has been considered by adopting a Model-Based Design approach with the help of MATLAB and Simulink. Iterative verification of each step was done starting from floating-point to fixed-point representation, HDL code generation and, finally, hardware co-simulation. We have considered FIL co-simulation of the receiver with the implementation of the equalization of DFrFT-based OFDM system for BPSK transmission over a frequency selective Rayleigh fading channel in presence of CFO. Simulation results clearly demonstrate that the FPGA implementation of a DFrFT-based OFDM system in presence of CFO has the same performance as that obtained from Monte Carlo simulation. Also, the performance is validated with the fixed-point model of the DFrFT-based OFDM. The approach described in the paper constitutes an efficient way to convert the floating-point model into a fixed-point one to be run in an FPGA and then verify its correctness through FIL co-simulation.

References

[1] E. Sejdic, I. Djurovic, and L. J. Stankovic, "Fractional Fourier transform as a signal processing tool: an overview of recent developments," Signal Processing, vol. 91, no. 6, pp. 13512–294, June. 2011.

[2] B. Bellalta, *et al.* "Next generation IEEE 802.11 wireless local area networks: current status, future directions and open challenges," Computer Communications, vol. 75, pp. 1–25, Feb. 2016.

[3] U. Mahesh and A. K. Chatuvedi, "Closed form BER expressions for BPSK OFDM systems with fractional timing offset and carrier frequency offset," in Proc. of NCC, pp. 1–4, Kharagpur, India, 2012.

[4] Y. Wang, Z. Zhang, and Y. Chen, "BER analysis of BPSK OFDM systems with residual frequency and timing offsets over frequency selective Rayleigh fading channels," J. Comp. Inform. Syst., vol. 8, no. 24, pp. 10349-10357, Dec. 2012.

[5] Jing Zhang, Zulian Wang, "ICI analysis for FRFT-OFDM systems to frequency offset in time frequency selective fading channels," IEEE Comm. Lett., vol. 14, pp. 888-890, October 2010.

[6] S. Kumari, *et al.* "Exact BER analysis of FRFT-OFDM system over frequency selective Rayleigh fading channel with CFO," Elect. Lett., vol. 49, no.13, pp. 1299-1301, Sept. 2013.

[7] M. R Mousavi and A. Shahzadi, ''Hyperbolic FRFT-OFDM system BER analysis over high Doppler Rician fading channels,'' Elect. Lett., vol. 51, no. 24, pp. 2003-2005, Nov. 2015 .

[8] A. Kumar, M. Magarini, H. D. Joshi, and R. Saxena, "Exact SER analysis of DFrFT-based QPSK OFDM system over frequency selective Rayleigh fading channel with CFO," Journal of Computer Networks and Communications, vol. 2016, pp. 7, 2016.

[9] M. H. Azmy, *et al.* ''Performance improvement of the OFDM system based on fractional Fourier transform over doubly dispersive channels,'' in Proc. of ICCES, Cairo, Egypt, pp. 81-85, Nov. 2012.

[10] A. K. Singh and R. Saxena, "DFRFT: a classified review of recent methods with its application," Journal of Engineering, 2013.

[11] Y. H. Hu and S. Naganathan, "A novel implementation of chirp z transformation using a CORDIC processor," IEEE Transactions on ASSP, vol. 38, pp. 352-354, 199O.

[12] M, Alexander, *et al.* "Revolutionizing software defined radio: case studies in hardware, software, and education," IEEE Communications Magazine, vol. 54, pp. 68-75, 2016.

[13] S. Beek, and S. Sharma, ''Best practices for FPGA prototyping of MATLAB and Simulink Algorithms,'' Electronic Engineering Journal, August 2013.

[14] P. Ządek, A. Koczor, M. Gołek, L. Matoga, and P. Penkala, ''Improving efficiency of FPGA-in-the-loop verification environment,'' IFAC, vol. 4, no. 48, pp.180-185, 2015.

[15] A. Kumar, M. Maurizio, and S. Olivieri, ''FPGA-in-the-loop simulation,'' Conference: MathWorks EXPO, DOI: 10.13140/RG.2.2.24267.18727, June 2016.