

FPGA Implementation of a Message-Passing OFDM Receiver for Impulsive Noise Channels

Karl F. Nieman[†], Marcel Nassar[‡], Jing Lin[†], and Brian L. Evans[†]

[†]The University of Texas at Austin, Austin, TX

[‡]Mobile Solutions Lab, Samsung Information Systems America, San Diego, CA

Email: {karl.nieman, mnassar, jing.lin}@utexas.edu, bevans@ece.utexas.edu

Abstract—Conventional orthogonal frequency division multiplexing (OFDM) communication systems are typically designed assuming additive white Gaussian noise and interference statistics. However, in many applications, such as Wi-Fi and powerline communications (PLC), impulsive statistics are often observed. Impulsive noise can degrade the signal-to-noise ratio (SNR) of all subcarriers and impair communication performance. In this work, we design and implement a real-time OFDM receiver with approximate message passing (AMP) to estimate and mitigate impulsive noise. The goal is to meet throughput and latency requirements while guaranteeing improved communication performance in impulsive noise. Our contributions include (i) modeling functional parallelism in an AMP OFDM receiver in synchronous dataflow, (ii) converting an AMP OFDM PLC receiver to using only fixed-point data and arithmetic, and (iii) mapping the receiver in fixed-point onto a Field Programmable Gate Array (FPGA) target using a high-level graphical synthesis tool. Our FPGA OFDM transceiver testbed achieves full streaming throughput at G3-PLC rates and recovers up to 8 dB SNR of impulsive noise over a wide SNR range.

I. INTRODUCTION

One of the most widely accepted (and thus used) communication channel models consists of a linear time-invariant filter with additive noise. This noise is typically modeled as an additive, white Gaussian noise (AWGN) process with independent, identically-distributed (*iid*) samples. Though this model accurately reflects thermal noise present in communication system electronics, it fails to capture well-known empirical and physical model-derived noise and interference statistics.

Extensive measurement campaigns taking place over the last several decades have revealed that the statistics of this additive component are in fact *impulsive*, with spurious components sometimes reaching 40 dB above background noise levels [1]–[3]. Physical interference models, based on the reality of increasingly interference-dominated cellular and wireless access networks, have further reinforced these results [4]. Using a revised noise model, communication systems can be redesigned for robustness and achievable rate in the presence of impulsive noise [5], [6]. In this work, we take this approach to design and implement a real-time OFDM transceiver featuring an impulsive noise estimation and mitigation block (see Fig. 1). Our implementation is realized in hardware using field-programmable gate arrays (FPGAs).

This research was supported by equipment donations, technical support, and funding from National Instruments, Austin, TX USA through their Academic Lead User Program. Additional funding was provided by Semiconductor Research Corporation under SRC GRC Task 1836.063. M. Nassar had performed this research as a PhD student at The University of Texas at Austin.

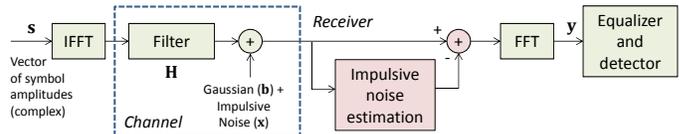


Fig. 1. An OFDM system with an impulsive noise channel. Impulsive noise estimation/mitigation is performed upstream of the conventional DFT receiver.

One application that suffers from strong impulsive noise is powerline communications (PLC). PLC systems operate by coupling modulated signals onto wires and transmission lines designed for electrical power delivery. These signals are subject to strong impulses and transient disturbances caused by the multitude of switching devices connected to the power grid. Methods for impulsive noise mitigation applicable to PLC include low-SNR techniques such as nulling and thresholding [5] and more robust, wide-SNR techniques based on sparse (many samples near zero) reconstruction [6], [7]. In this work, we focus on approximate message passing (AMP), a sparse reconstruction technique shown to have high reconstruction performance while being readily parallelizable and scalable using mostly scalar arithmetic. We target our application to G3-PLC, a widely-adopted baseband OFDM PLC standard (see Table I). FPGA processing is used in our design in order to deterministically achieve G3-PLC rates.

II. IMPULSIVE NOISE MITIGATION BY AMP

As demonstrated by Caire [6], OFDM systems can exploit the impulse spreading property of the discrete Fourier transform (DFT) to sample impulsive noise in the frequency domain. These samples can be applied in a compressed sensing framework to reconstruct time-domain noise. These estimates can be subtracted from incoming samples to reduce impulsive noise prior to equalization and decoding (see Fig. 1).

TABLE I
SPECIFICATIONS FOR G3-PLC OPERATION IN CENELEC-A BAND

Sampling frequency	$f_s = 0.4$ MHz
(Real) FFT Size (symbol length)	$N = 256$
Number of overlapped samples	$N_O = 8$
Number of cyclic prefix samples	$N_{CP} = 30$
Subcarriers used (indexing by zero)	23-58
First and last subcarrier frequencies	35.9 - 90.6 kHz
Null subcarriers (indexing by zero)	0-22, 59-128
“ ” used by AMP (indexing by zero)	1-22, 59-100

A. Approximate Message Passing (AMP)

AMP has been formulated as a low-complexity, scalable compressed sensing technique, which leads itself to efficient hardware implementation for signal reconstruction [8]–[10]. Recently, Nassar *et al.* proposed an AMP framework for joint impulsive noise mitigation and channel estimation in OFDM systems [11]. This framework exploits the spreading of impulsive noise information across subcarriers by virtue of the DFT. Frequency-domain noise is used to reconstruct the time-domain noise using AMP. We apply this method using noise samples available in G3-PLC null subcarriers. As referenced in Table I, G3-PLC’s baseband signaling has 92 such null subcarriers, of which we use 64 (subcarriers 1-22, 59-100).

B. OFDM Receiver with Message-Passing

An OFDM system with cyclic prefix diagonalizes the circulant channel matrix \mathbf{H} using the DFT/IDFT matrices \mathbf{F}/\mathbf{F}^* :

$$\mathbf{y} = \mathbf{F}\mathbf{H}\mathbf{F}^*\mathbf{s} + \mathbf{F}\mathbf{n}. \quad (1)$$

Let the noise vector \mathbf{n} consist of an impulsive and background component—i.e., $\mathbf{n} = \mathbf{x} + \mathbf{b}$. Samples are drawn from an *i.i.d.* two-mode Gaussian mixture distribution where γ_X and γ_B denote the impulsive and background noise power ($\gamma_X \gg \gamma_B$):

$$GM(\pi, \gamma_B, \gamma_X) = \pi\mathcal{N}(0, \gamma_B + \gamma_X) + (1 - \pi)\mathcal{N}(0, \gamma_B). \quad (2)$$

The AMP algorithm takes as inputs the samples of the null subcarriers $y_{\Omega i}$ from the set Ω where $|\Omega| = M < N$. N is the total number of subcarriers and thus the FFT size of the OFDM system. The process of sampling null subcarriers can be implemented efficiently by indexing the corresponding entries of the FFT output. We express this as \mathbf{I}_{Ω} , a subset of the Ω rows of the $N \times N$ identity matrix \mathbf{I} .

Table II outlines the explicit computational steps of the AMP formulation for this system model. Detailed derivations are provided in [11]. The inputs to this algorithm are the M -length vector of null subcarrier samples \mathbf{y}_{Ω} , background and impulsive noise variance (γ_B, γ_X), and impulse probability π . The output is the N -length time-domain estimate of the impulsive noise $\hat{x}_j(t+1)$, where t denotes iteration count.

The first step of AMP is an initialization step. The second, or output linear step, involves the forward FFT operation where a_{ij} denotes the ij^{th} entry of the transform matrix, the DFT in this case. Our target architecture, the Xilinx Virtex-5 FPGA, implements the forward FFT as $X_k = \sum_{n=0}^{N-1} x_n \omega_N^{nk}$ and the IFFT as $x_n = (1/N) \sum_{k=0}^{N-1} X_k \omega_N^{-nk}$, where $\omega_N = e^{-j2\pi/N}$. Our scaling by N and M factors reflect this. Step 3 is the output non-linear step which consists of simple scalar operations. Step 4, the input linear step, involves the IFFT. Step 5 requires the highest depth of sequential operations, the most prohibitive of which is the exponential since its output has a high dynamic range. This was averted using an approximation discussed in Section V. Steps 2-5 are repeated until convergence. In our implementation, we use 4 iterations to balance performance with throughput constraints.

TABLE II
AMP NOISE MITIGATION FOR OFDM RECEIVER

Step	Calculation
1	$\hat{x}_j = 0$ $\tau_j^x(t) = (1 - \pi)\gamma_X, \quad \forall j \in \{1, \dots, N\}$ $\hat{s}_i(t) = 0, \quad \forall i \in \{1, \dots, M\}$
2	$\tau_i^p(t) = \sum_j a_{ij} ^2 \tau_j^x(t) = \sum_j \tau_j^x(t)$ $\hat{p}_i(t) = \sum_j a_{ij} \hat{x}_j(t) - \tau_i^p(t) \hat{s}_i(t-1)$ $= \mathbf{I}_{\Omega} \text{FFT}(\hat{x}(t)) - \tau_i^p(t) \hat{s}_i(t-1)$
3	$\tau_i^s = \frac{1}{\gamma_B + \tau_i^p}, \quad \forall i \in \{1, \dots, M\}$ $\hat{s}_i(t) = \tau_i^s(t) (y_{\Omega i} - \hat{p}_i(t))$
4	$\tau_j^r(t) = N \left[\sum_i a_{ij} ^2 \tau_i^s(t) \right]^{-1}$ $\hat{r}_j(t) = \hat{x}_j(t) + \tau_j^r(t) \sum_i (a_{ij}^* \hat{s}_i(t))$ $= \hat{x}_j(t) + \tau_j^r(t) \text{IFFT}(\mathbf{I}_{\Omega}^T \hat{x}(t))$
5	$\eta_j = \frac{\pi}{1-\pi} \frac{\tau_j^r(t)}{\gamma_I + \tau_j^r(t)} \exp\left(\frac{\gamma_I \ \hat{r}_j(t)\ ^2}{\tau_j^r(t) (\gamma_I + \tau_j^r(t))}\right)$ $\rho_j = \frac{\eta_j}{1 + \eta_j}$ $\hat{x}_j(t+1) = \frac{\gamma_I}{\gamma_I + \tau_j^r(t)} \rho_j \hat{r}_j(t)$ $\tau_j^x(t+1) = \eta \left[\frac{\tau_j^r(t) \gamma_I}{\gamma_I + \tau_j^r(t)} + \left(\frac{\gamma_I}{\gamma_I + \tau_j^r(t)}\right)^2 \ \hat{r}_j(t)\ ^2 (1 - \eta_j) \right]$
*	repeat steps 2-5 until convergence, resulting time-domain noise estimate: $\hat{x}_j(t+1)$

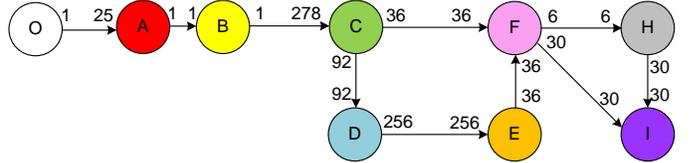


Fig. 2. Synchronous dataflow (SDF) model of the OFDM receiver with AMP. Nodes are tasks and edges represent data dependencies. When task C executes, it consumes 278 samples on its input, and produces 92 samples on its bottom output and 36 samples on its right output.

III. SYNCHRONOUS DATAFLOW MODEL

The streaming operation of the AMP-enhanced OFDM receiver can be modeled in synchronous data flow (SDF) as shown in Fig. 2. Nodes are application tasks and edges are first-in first-out (FIFO) queues that represent data dependencies. Each task produces and consumes a fixed number of samples in each execution. The tasks in Fig. 2 correspond to (A) sample rate conversion, (B) time and frequency offset correction, (C) FFT and CP removal, (D) AMP noise estimation, (E) FFT, (F) noise subtraction, (H) channel estimation, and (I) channel equalization. The deterministic properties of SDF enable fully-static analysis and synthesis of the system behavior. In particular, a periodic schedule can be determined by static analysis. For example, a periodic schedule for the SDF graph in Fig. 2 is (278A)(278B)CDEFHI. In Section V.B, we convert this SDF to a globally asynchronous, locally synchronous (GALS) model of computation using LabVIEW DSP Design Module. GALS affords more flexibility in hardware timing while still providing benefits of SDF.

IV. FIXED-POINT MAPPING

Before proceeding with the hardware implementation, the AMP impulsive noise mitigation algorithm in Table II needs to be mapped from floating-point to fixed-point. This typically involves determining the dynamic range of each variable and assigning an appropriate fixed-point representation of it that guarantees a specified level of performance. Although in-range data values, or the dynamic range captured by the fixed-point representation, correlates to performance, more complicated dependencies on variable sizings can be observed. To address this, we simulated the algorithm using the `fi` and `NumericScope` data types, both part of MATLAB’s fixed-point toolbox, as follows: (i) for each run of the algorithm we log the variables values; (ii) then, using the `NumericScope` we appropriate size these variables; (iii) finally, using the `fi` data type we simulate the algorithm with the variable sizing found in step (ii) and make sure we meet the target performance. The above procedure was performed for a fixed set of input parameters and input noise statistics—i.e., $\gamma_B = 0.0025$, $\gamma_X = 0.25$, and $\pi = 0.1$. The performance metric for the MATLAB simulation was the reduction in impulsive noise power over a large number of trials using simulated noise with matched parameters. Our fixed-point mapping was able to perform within 0.5 dB of the double precision floating-point MATLAB version of the algorithm.

The resulting variable sizings for the AMP implementation are shown in Table III. Our fixed-point data type representation convention is $SN_W.N_I$, where S is replaced by ‘U’ for unsigned and ‘I’ for signed, N_W is the wordlength in bits, and N_I is the integer width (or scaling by 2^{N_I}). The fractional width is $N_W - N_I$. By targeting 90% or higher in-range data values, most variables were able to be sized within 16-bit wordlengths, an efficient size to utilize the single-cycle digital signal processing (DSP) 48 blocks in the Virtex-5 FPGA.

V. FPGA IMPLEMENTATION

An OFDM transceiver using the fixed-point version of the AMP algorithm was implemented across several Xilinx Virtex-5 FPGAs. The intended hardware mapping was to one FPGA transmitter and one FPGA receiver. After initial sizing estimates, the AMP algorithm and the OFDM receiver were not able to fit within a single FPGA. The receiver was partitioned across two FPGAs that share data across the PXI-Express (PXIe) bus, an instrumentation-targeted version of the PCI-Express bus developed by National Instruments. Hardware mapping for the three FPGAs is detailed in the following section and presented graphically in Fig. 3.

The OFDM transmitter interleaves modulated $2 \times I8.1$ complex symbols with a conjugate-symmetric pair of data and reference symbols. Reference symbols are encoded as quadrature phase shift keying (QPSK) signals and are interleaved between every 5 data subcarriers. These symbols are used for channel estimation at the receiver. Conjugate-symmetry is enforced in order to make the output purely real-valued, and coherent QPSK is used as the data subcarrier modulation.

TABLE III
FIXED-POINT WORD SIZES FOR AMP PARAMETERS AND VARIABLES

Inputs	Complex?	Length	Representation
π		1	U16.0
γ_X, γ_B		1	U16.-2
y_i	✓	128	$2 \times I16.1$
Intermediates	Complex?	Length	Representation
$\hat{x}_j(t)$	✓	256	$2 \times I16.1$
$\tau_j^x(t)$		256	U16.-2
$\hat{s}_i(t)$	✓	128	$2 \times I16.6$
$\tau_i^p(t)$		1	U16.-5
$\hat{p}_i(t)$		128	$2 \times I16.0$
τ_i^s		1	U16.0
$\tau_j^r(t)$		1	U16.-2
$\hat{r}_j(t)$	✓	256	$2 \times I16.2$
η_j		256	U16.6
ρ_j		256	U16.6
Outputs	Complex?	Length	Representation
$\hat{x}_j(t+1)$	✓	256	$2 \times I16.1$

The receiver consists of two FPGAs. The first FPGA, ‘G3RX’, performs front-end processing—i.e. resampling and time/frequency synchronization. Resampling is performed using a multi-rate, two-stage finite impulse response (FIR) filter whose coefficients were designed using a LabVIEW multi-rate filter design tool. Synchronization is performed by correlating the signal with a delayed version of itself. Since the cyclic prefix (CP) is periodic, its autocorrelation exhibits a peak at a lag equal to the the symbol duration. The location of this peak and its resulting phase can be used for time and carrier frequency offset synchronization [12]. Frequency offset correction is performed using a discrete cosine generator synthesized to 0.1 Hz phase increments using Xilinx CoreGen. The second receiver FPGA, ‘AMPEQ’, performs the AMP algorithm, channel estimation, and equalization. The AMP algorithm uses frequency-domain null subcarrier samples and parameters as input from ‘G3RX’. Reference symbols are de-interleaved and used for channel estimation. Equalization is performed using a zero-forcing (ZF) equalizer.

In parameterizing the AMP algorithm for G3-PLC signaling, M and N are chosen to be 128 and 256, respectively. $M = 128$ instead of 64 (the number of null subcarriers) because we implement the full complex version of AMP based on 256-length complex-valued FFTs/IFFTs, whereas G3-PLC uses a 256-length real-valued FFT. In our implementation, we use null subcarriers 1-22 and 59-100 and their negative frequency pairs 234-255 and 156-197. The exponential function in Step 5 is approximated using a sixth-order Taylor series expansion. This was done to avert dynamic range issues when rescaling the output of the Xilinx CORDIC computation, which requires the inputs be normalized to $[0, 1]$. The ρ_j calculation in Step 5 in Table II involves a 16-bit division 256 times per iteration. This operation was parallelized into two streams, resulting in a 8656 cycle reduction in execution time at the expense of a small increase in resource utilization.

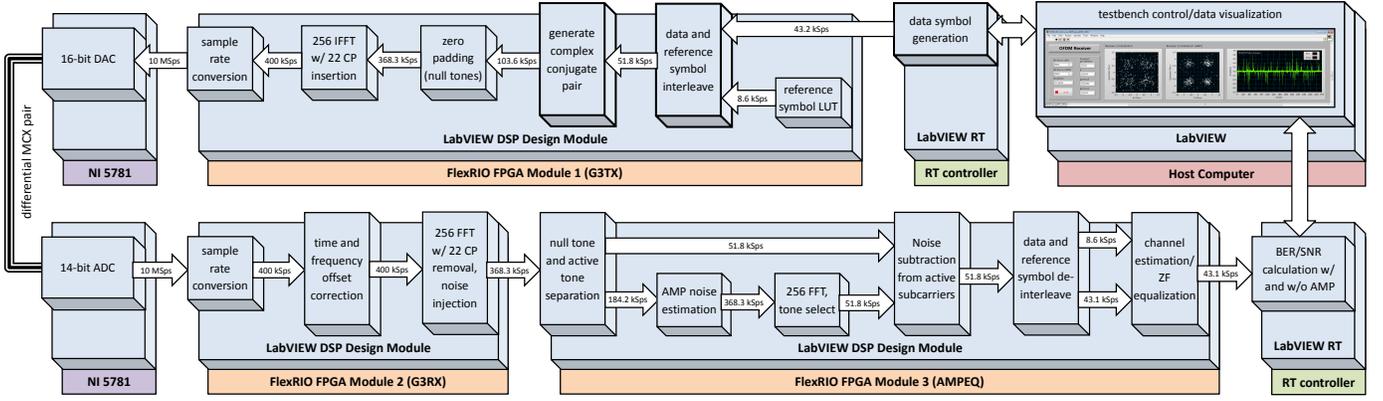


Fig. 3. A block diagram of the AMP-enhanced G3-PLC OFDM test system. Physical hardware blocks are separately colored, and software environments are shown next in the hierarchy. Functional blocks are the highest hierarchical level with throughput rates and dataflow direction shown between them.

A. Target Architecture and System Mapping

Fig. 3 shows the system block diagram of the AMP-Enhanced OFDM transceiver testbed. The testbed consists of two National Instruments PXIe-1082 chassis, one for the TX and one for the RX. Both chassis make use of an onboard PXIe-8133 1.73 GHz Quad-Core (Intel Core i7-820QM) PXI Express controller running a real-time operating system. These PXI controllers have been targeted to the deployment of LabVIEW Real-Time (RT) applications which can execute deterministic dataflow computations at granularity on the order of 1 ms. Each of the two systems communicates via gigabit Ethernet back to a host PC for high-level, non-deterministic performance analysis and visualization. Each of the two RT chassis are configured as follows:

- **TX chassis** is fitted with a single NI PXIe-7965R FlexRIO FPGA Module named ‘G3TX’ featuring a Virtex-5 SX95T FPGA that is used for interleaving data/reference symbols, OFDM modulation (IFFT), appending cyclic prefix (CP), and upsampling to the 10 MS/s NI-5781 digital-to-analog converter (DAC) sample rate. The host RT chassis generates random data symbols to feed the input symbol direct memory access (DMA) first-in, first-out (FIFO) buffer. Modulated and upconverted samples are clocked out of the FlexRIO and into the NI-5781 adapted module. These samples are then passed across two micro-coaxial (MCX) 50Ω differential pairs to the RX NI-5781 (although the output of the quadrature component is always zero by virtue of the conjugate-symmetry enforced in ‘G3TX’).
- **RX chassis** is fitted with two NI PXIe-7965Rs. The first FPGA module, ‘G3RX’, is configured for the front-end receiver processing—i.e. downsampling, synchronization, frequency-offset estimation, OFDM demodulation (FFT), CP removal, and noise injection. The second FPGA module, ‘AMPEQ’ is passed frequency-domain samples from ‘G3RX’ using a peer-to-peer (P2P) stream over the PXIe backplane. These samples, in addition to parameters γ_X , γ_B , and π are passed from the RX host controller and are used as inputs to the AMP algorithm.

B. Synthesis Using LabVIEW DSP Design Module (LVDDM)

LVDDM is a high-level FPGA synthesis tool developed by National Instruments. LVDDM takes as an input a high-level data flow diagram of DSP computations (see Fig. 4) and translates them to a LabVIEW FPGA register-transfer level representation. This representation is then mapped to a Verilog hardware description language product that is input to the Xilinx Integrated Software Environment FPGA compiler. LVDDM provides direct interfaces to Xilinx CoreGen features, allowing for the implementation of highly-optimized FPGA constructs including FFTs, discrete fourier transforms, direct digital synthesizers, and multi-rate FIR filter implementations.

The project files including the MATLAB fixed-point sizing simulation, LabVIEW host virtual instruments (VIs), DSP Diagrams, and generated FPGA bitfiles are available for download on the project webpage:

<http://users.ece.utexas.edu/~bevans/papers/2013/fpgaReceiver/>

C. Resource Utilization

The resource utilization and master clock (MCLK) of each of the three FPGAs used in the test system is shown in Table IV. Resource utilization is highest in ‘AMPEQ’, the FPGA with AMP processing and equalization. Overall resource usage could be reduced through further computation and buffer size optimization. Additionally, the use of real- vs. complex-valued FFTs could afford at least 2x reduction in resource utilization. Complex form was used so that the algorithm could be applied to other projects using complex signaling.

TABLE IV
RESOURCE UTILIZATION FOR EACH FPGA

Resource Name	Total Available	FPGA1 G3TX	FPGA2 G3RX	FPGA3 AMPEQ
Total slices	14720	32.6%	64.0%	94.2%
Slice registers	58880	15.8%	39.3%	59.0%
Slice LUTs	58880	17.6%	42.4%	71.4%
DSP48s	640	2.0%	7.3%	27.3%
Block RAMs	244	7.8%	18.4%	29.1%
MCLK (MHz)	-	125	40	40

VI. RESULTS

To evaluate the real-time performance of this implementation, a bit-error-rate (BER) testbench was constructed in LabVIEW RT running on each embedded PXI controller. Equalized symbols, input impulsive noise, error vector magnitude (EVM), and trailing bit-error-rates (BER) are plotted for both the AMP-enhanced and conventional OFDM receiver (see top of Fig. 4). Blocks of 6 OFDM symbols were processed at a time, though this number can be varied by setting a control in the transmit and receive VIs. Using this framework, a test array of input and output data bytes were transmitted across the OFDM link using synthesized Gaussian mixture noise input to the receiver. Received symbols were demodulated and compared for errors. Using the test environment, the receiver was configured with a fixed π , $\gamma_X = 0.25$, and γ_B was set to be a multiple of γ_X listed in Fig. 5. Transmit gain was swept over a 20 dB range in order to profile BER vs. SNR. Here, SNR is the transmit power over background noise power, γ_B .

The results of the BER analysis are shown in Fig. 5. As shown in the figure, by mitigating impulsive noise with AMP, SNR levels of up to 8 dB are recovered and gains are realized over a wide SNR range. The AMP and conventional receivers have nearly identical BER curves in non-impulsive, or $\pi = 0$, noise. Injected noise was synthesized using a match to AMP input parameters; however, estimation techniques discussed in [11] could be used to learn the parameters adaptively using amortization over OFDM symbols. These results indicate that this algorithm could be targeted for implementation in other OFDM receivers beyond G3-PLC systems.

VII. CONCLUSION

We design and implement a real-time OFDM receiver with AMP impulsive noise mitigation on FPGAs. Our design flow was to (i) model the receiver in synchronous dataflow (SDF), (ii) convert the receiver to fixed-point data and arithmetic, and (iii) synthesize the fixed-point SDF model onto two Xilinx Vertex-5 FPGAs. In the third step, we generalize the SDF receiver model to a globally asynchronous, locally synchronous computational model in LabVIEW DSP Design Module. Our OFDM transceiver testbed achieves real-time streaming throughput at G3-PLC rates and mitigates up to 8 dB SNR of impulsive noise.

REFERENCES

- [1] A. Spaulding and D. Middleton, "Optimum reception in an impulsive interference environment—part I: Coherent detection," *Communications, IEEE Transactions on*, vol. 25, no. 9, pp. 910–923, 1977.
- [2] K. Blackard, T. Rappaport, and C. Bostian, "Measurements and models of radio frequency impulsive noise for indoor wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 7, pp. 991–1001, 1993.
- [3] M. Sanchez, L. de Haro, M. Ramon, A. Mansilla, C. Ortega, and D. Oliver, "Impulsive noise measurements and characterization in a UHF digital TV channel," *IEEE Trans. Electromagn. Compat.*, vol. 41, no. 2, pp. 124–136, 1999.
- [4] K. Gulati, B. L. Evans, and S. Srikanteswara, "Joint temporal statistics of interference in decentralized wireless networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6713–6718, 2012.
- [5] D.-F. Tseng, Y. Han, W. H. Mow, L.-C. Chang, and A. Vinck, "Robust clipping for OFDM transmissions over memoryless impulsive noise channels," *IEEE Communications Letters*, vol. 16, no. 7, 2012.

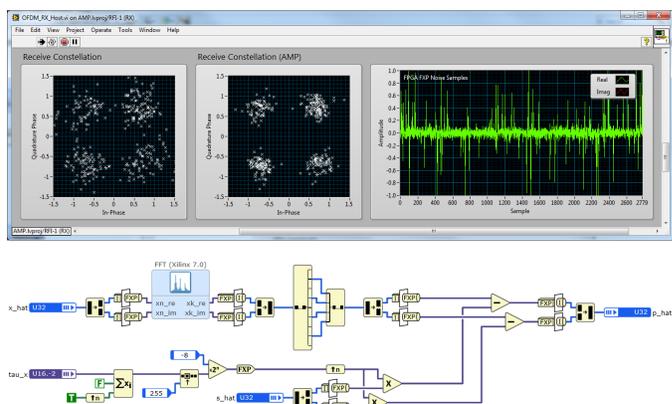


Fig. 4. (top) The receiver host VI, showing input impulsive noise and an equalized constellation for the conventional and AMP-enhanced OFDM receivers over one OFDM frame. (bottom) The LabVIEW DSP Design Module graphical representation of the computations in Step 2 of Table II.

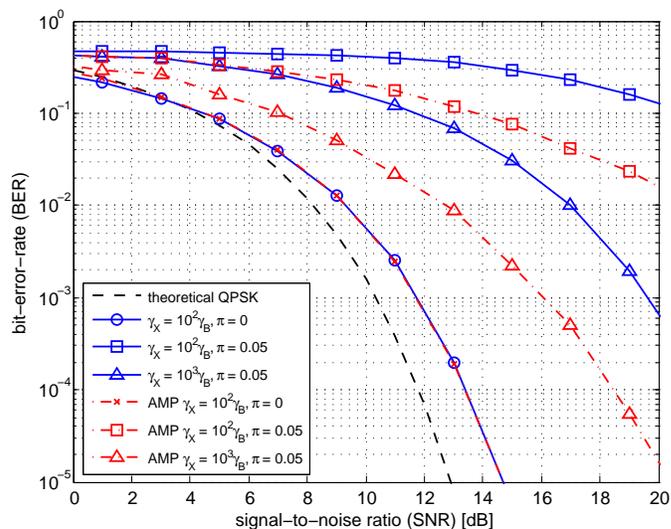


Fig. 5. BER plots of received symbols with and without AMP algorithm for impulse-free noise and for impulsive noise with $\pi = 0.05$ (5% probability of impulse) using impulses 20 and 30 dB above the background level.

- [6] G. Caire, T. Al-Naffouri, and A. Narayanan, "Impulse noise cancellation in OFDM: an application of compressed sensing," in *Proc. IEEE International Symposium on Info. Theory*, 2008.
- [7] A. Mehboob, L. Zhang, J. Khangosstar, and K. Suwunnapak, "Joint channel and impulsive noise estimation for ofdm based power line communication systems using compressed sensing," *Proc. IEEE Int. Symposium on Power Line Communications and Its Applications*, 2013.
- [8] D. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci.*, vol. 106, pp. 18914–18919, 2009.
- [9] L. Cavigelli and P. A. Hager, "Implementing the approximate message passing (AMP) algorithm on a GPU," *ETH Report*, 2012.
- [10] P. Maechler, C. Studer, D. Bellasi, A. Maleki, A. Burg, N. Felber, H. Kaeslin, and R. Baraniuk, "VLSI design of approximate message passing for signal restoration and compressive sensing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 579–590, 2012.
- [11] M. Nassar, P. Schniter, and B. L. Evans, "A factor graph approach to joint OFDM channel estimation and decoding in impulsive noise environments," *IEEE Trans. Signal Process.*, submitted June 7, 2013.
- [12] M. Sandell, J. van de Beck, and P. Borjesson, "Time and frequency synchronization in OFDM systems using the cyclic prefix," *IEEE International Symposium on Synchronization*, pp. 16–19, 1995.