Hardware Modelling of a PLC Multipath Channel Transfer Function

Ann E. Dulay, Roderick Yap, Lawrence Materum
De La Salle University
ann.dulay@dlsu.edu.ph

***Abstract*— PLC channel transfer functions are critical in developing channel emulators. Emulators are used to test PLC transceivers that would have been otherwise tested on a live network which is not only hazardous but also is not repetitive since the PLC network's characteristics vary with time and frequency. The PLC channel transfer function that is considered here is the multipath channel model of Zimmermann and Dostert. Zimmermann's multipath channel transfer function includes not only the attenuation experienced by the signal, but also the phase change due to the multipath. The inclusion of the phase in the transfer function requires the processing of imaginary numbers in the FPGA. Euler's theorem is used to convert from the polar form to the rectangular form of the transfer function. This paper focuses on the hardware modelling of the rectangular form using Hardware Description Language. The real and imaginary components of the rectangular form are modelled as two circuit models. The designed channel transfer function module uses fixed point (FP) format. The HDL model was successfully synthesized to a hardware equivalence using the XILINX ISE tool. XILINX core generator was used for the CORDIC module that computes the sine, cosine, hyperbolic sine and hyperbolic cosine functions. Simulation result of the hardware model revealed an accuracy of up to at least 3 decimal places when compared to the theoretical results using Excel or Matlab.**

***Index Terms*— PLC; Multipath Channel; Transfer Function; Complex Number.**

I. INTRODUCTION

Power line communications (PLC) is a special type of communications system that uses the power cable as the communications media [2]. It can be classified both as a wired and a wireless system. It is a wired system for the obvious reason that there is a physical wire that carries the signal, and this is where the transmission line model is based. It can also be thought of as a wireless system since the power cables may experience multipath propagation due to branching caused by distribution taps to houses in the outdoor PLC network [3], or branching caused by outlets in the indoor network [4]. The latter is modeled using the multipath propagation approach and the well-known channel model for this is the one presented in [1]. These models are used in developing PLC channel emulators.

Communications channel emulators are devices that mimic the characteristics of the channel. The channel emulator contains several blocks that would typically include data conversion blocks (ADC/DAC), transfer function block, FFT/IFFT blocks, noise blocks and other functional blocks. Except for the data conversion, all other blocks reside in the FPGA. At the time of this writing, all PLC channel emulators implement the transfer function block using impulse response [5], [6], [7], [8], [9]. The impulse response is realized using a FIR filter whose coefficients depend on the number of points. The implementation of the FIR filter is simpler compared to the frequency domain counterpart for a smaller number of parameters, however, the complexity increases as the number of parameters grow, hence, for most of the PLC channel emulator presented, an external PC or DSP is used to generate the filter coefficients. Frequency domain offers a constant complexity and is practically useful for a very large set of coefficients [10].

The critical block of the emulator that is implemented in the FPGA is the channel transfer function. The channel transfer function can be implemented in the hardware using a look-up table or as a module. The utilization of the look up table is easier, however, one needs to generate in Matlab all the values for all the indices needed, then copy and save to LUT in FPGA. This is not a very flexible approach since some of the parameters may change, and it would be tedious to generate the values in Matlab when a change is needed. Aside from this, the transfer function considered in this paper has both a real and imaginary component. This paper presents the hardware model and implementation of the Zimmermann multipath PLC channel model.

II. LITERATURE SURVEY

A. Power Line Communications

Power Line Communications uses the power cables as the medium in transmitting data or control signal. It is now being considered as one of the key technologies in smart grid systems [11]. There are two types, the narrowband and broadband PLC. Narrowband PLC operates in the frequency range 30 kHz to 500 kHz while broadband PLC operates up to 30 MHz. Narrowband are thought to be practicable for automatic meter reading while broadband is expected to pervade the internet applications [12], [13]. Figure 1 shows an illustration of PLC concept.

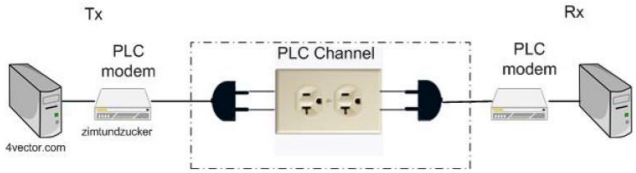

Figure 1: The PLC Concept

B. PLC Channel Emulator

PLC channel emulator is a piece of hardware that mimics the characteristics of the power line networks. Since it replaces the real network, it allows full control on the variations of different parameters including electrical loads, number of paths, and noise present in the power line. This

makes testing of PLC transceivers and other devices more flexible and can provide a more useful information for improving the design of the PLC transceiver. The PLC channel represented in Figure 1 can be modeled as in Figure 2. Two approaches have been used to model the channel, the top-down approach and the bottom-up approach [14]. The latter models the PLC channel using the transmission line theory. The former approach models it using multipath propagation. Regardless of the model used, the emulator is typically implemented on a hardware like FPGA (Figure 3). The blocks inside the FPGA comprise the FFT, channel transfer function (CTF), multiplier, controller, and noise blocks. The emulator also includes an analog front end (AFE) that converts the signal from and to the format the FPGA understands. In this paper, the development of the channel transfer function block for implementation in the FPGA is presented.

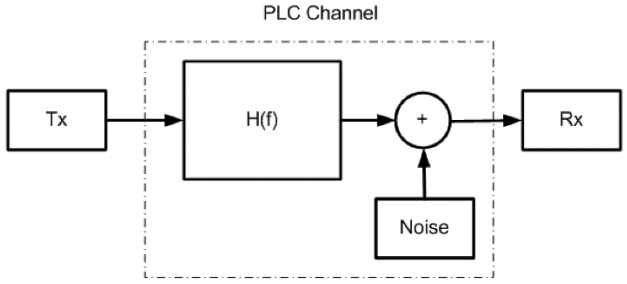

Figure 2: Basic Power Line Channel Model

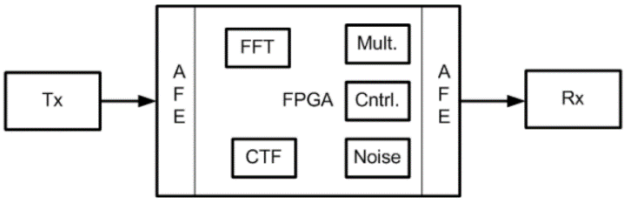

Figure 3: FPGA-Based PLC Channel Emulator

C. The Zimmermann Multipath Channel Model

The transfer function of the Zimmermann model is shown in Equation (1).

$$H(f)=\sum_{i=0}^{N} g_i.e^{-(a_0+a_1 f^k)d_i}.e^{-j2\pi f\frac{d_i}{v_p}} \quad (1)$$

where the weighting factor g_i is a product of transmission and reflection factors (note that $g_i \leq 1$), i is the number of paths, a_0, a_1 are the attenuation parameters, k is the exponent of the attenuation factor, d_i is the path length, and v_p is the propagation velocity. a_0 and k for the fifteen-path used here are 0 and 1 respectively, hence, they can simply be eliminated from the above equation such that the above can be simplified to Equation (2):

$$H(f)=\sum_{i=0}^{N} g_i.e^{-c_1 f}.e^{-jc_2 f} \quad (2)$$

where

$$c_1 = -a_1 d_i \quad (3)$$

$$c_2 = 2\pi\frac{d_i}{v_p} \quad (4)$$

Division in FPGA is difficult hence the v_p in Equation 4 is implemented by taking its reciprocal and do multiplication operation such that c_2 becomes Equation (5):

$$c_2 = 2\pi d_i\frac{1}{v_p} \quad (5)$$

The expression in Equation (2) has two exponential expressions, both of which contain the variable f. The first exponential is determined by using the following relationship:

$$e^{-c_1 f} = \cosh(c_1 f) - \sinh(c_1 f) \quad (6)$$

The second exponential expression with the j component is translated by the Euler's theorem to the rectangular form presented in Equation (7) as follows:

$$e^{-j\varphi} = \cos\varphi - j\sin\varphi \quad (7)$$

where the *cos* represents the real component and *sin* represents the imaginary component, hence the transfer function given in Equation (2) can be written as:

$$H_f = \sum H_{re(i)} + \sum H_{im(i)} \quad (8)$$

Where

$$H_{re(i)} = g(i).e^{c_1 j}.\cos(c_2 f) \quad (9)$$

$$H_{im(i)} = -g(i).e^{c_1 f}.\sin(c_2 f) \quad (10)$$

III. HARDWARE DESIGN

As has already been presented in Sec. II-C, the Zimmerman mathematical model represents a summation of a set of terms. For this study, the fifteen-path model is chosen, hence each set of summation is comprised of 15 terms to be added based on one frequency value. Figure 4 shows the general block diagram of the hardware model which is based on Equation (9) and (10). The operation starts with the entrance of the frequency value to the Data entry module. This module computes for the angle c_2f and the exponent value c_1f. The implementation is straightforward multiplication. The output of the data entry module is fed to the XILINX core generated modules. One is Coordinate Rotation DIgital Computer (CORDIC) module for computing sine and cosine and the other is another CORDIC module for computing hyperbolic sine and hyperbolic cosine. The hyperbolic sine and cosine functions are employed for this design to compute for e^{-c1f}. It is a known relation that $\cosh x - \sinh x = e^{-x}$. This explains the presence of the subtractor module after the hyperbolic sine/cosine module. The CORDIC modules are ready made library models provided by XILINX. Based on the CORDIC datasheet, the CORDIC functions can only validly operate within a limited range. In the case of the sine and cosine CORDIC module, the acceptable input angle is only between π and $-\pi$. The sign corrector module, in conjunction with the

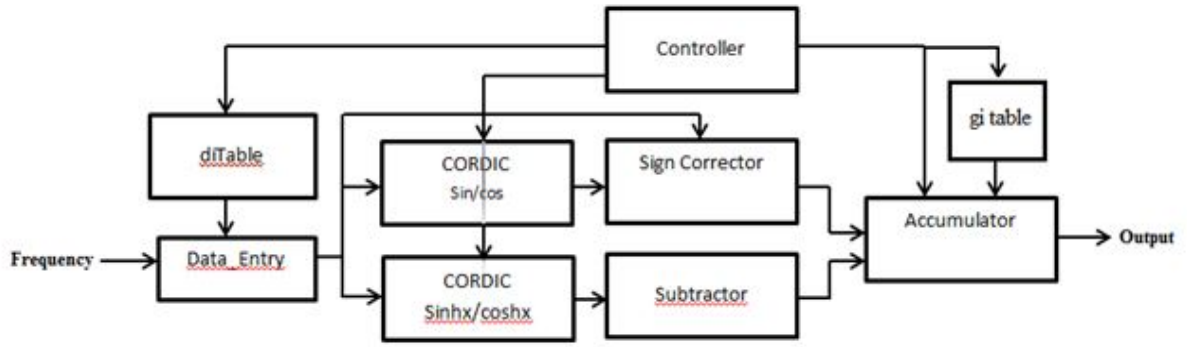

Figure 4: Block Diagram of the Zimmerman Hardware Model

input angle, works to correct the core generator output in the event that the input angle falls outside of the -π to π range. The di table and the gi table are implemented using multiplexer approach. The index values ranging from 1 to 15, released from the controller module serve as select lines of the multiplexer while the di or gi values are hardcoded as inputs to the multiplexers.

The timing of the entire system is done by the controller. The Controller is implemented using Finite State Machines. Figure 5 summarizes the operation. The controller outputs an index value every clock cycle for the release of the appropriate di and gi values. It sends the appropriate start signal to the CORDIC modules to start the computation of the hyperbolic sine/hyperbolic cosine and sine/cosine named as CORDIC1 and CORDIC2 respectively. The controller waits for the "finish" signal from both modules. When both CORDIC modules have finished computing, the controller sends a fetch signal to the accumulator. This module then multiplies the appropriate value of gi to the 2 core generator outputs. The accumulator does this repeatedly and accumulates the values obtained every fetch cycle and then adds them all up. The Controller on the other hand repeats the process of sending the fetch signal to the accumulator until all index values from 1 to 15 are consumed after which a "complete" signal is sent from this module to the accumulator to output the sum. When a pulse of "complete" signal is executed, it means the next frequency value can be evaluated and the entire process repeats again. Figure 6 shows the timing relation between the core generated modules signals and the controller signals. The "nd" is the start signal sent to the core generators while "cossinfinish" and "hyperfinish" are the "finish" signals coming from the respective core generator modules. The fetch pulse always comes after the "cossinfinish" signal while the "complete" pulse only comes when the index value has reached 15.

IV. Data and Results

The entire HDL code was synthesized using a Virtex FPGA library. Simulation of the HDL model was focused on every term to be generated by the Zimmerman mathematical model. We constructed a separate excel file to calculate every term out of the given formula. The excel spreadsheet result serves as our reference for determining whether the HDL simulation results are valid or not. Simulation results reveal very little discrepancy between the values generated by the hardware model and the values obtained from the excel file. Table 1 shows the comparison between the theoretically computed values using the math functions of Microsoft Excel and the HDL simulation result for the real part of the Zimmerman model. Table 2 shows the comparison between the theoretically computed values using the math functions of Microsoft Excel and the HDL simulation result for the imaginary part of the Zimmerman model. The HDL values obtained from the simulation results are initially hexadecimal values. These values are converted one by one to decimal using online hexadecimal to decimal converter. Other frequencies values e.g. 100KHz were also tried and the results obtained show very similar closeness between theoretical computation and the simulation result.

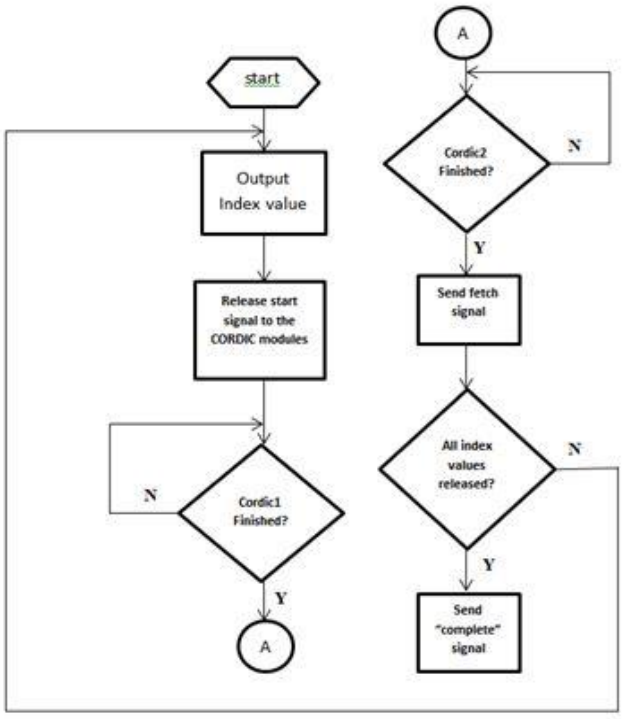

Figure 5: The Controller Behavior

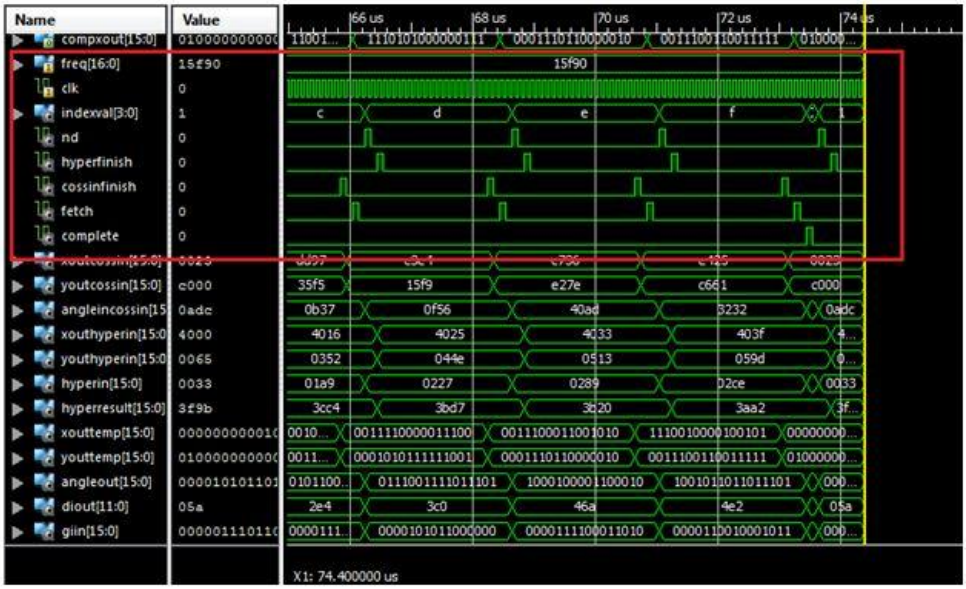

Figure 6: Timing Relation of Various Signals

Table 1
Theoretical vs. Actual Simulation Result for The Cosine Part At 90khz

Excel Results	HDL Results	Matlab
0.027174496	0.02717044	0.0272
0.039575515	0.03957801	0.0396
0.0930534	0.09306645	0.0931
-0.049276854	-0.04927663	-0.0493
-0.037780913	-0.03778176	-0.0378
-0.028752234	-0.02875309	-0.0288
0.020778567	0.0207798	0.0208
-0.012978887	-0.0129782	-0.0130
0.001473603	0.00147786	0.0015
0.00923024	0.0092283	0.0092
-0.033535821	-0.03352481	-0.0335
0.049016701	0.04901389	0.0490
-0.034870784	-0.03486999	-0.0349
0.023823243	0.0238225	0.0238
-1.78654E-07	0	0.0000

Table 2
Theoretical vs. Actual Simulation Result for The Sine Part At 90khz

Excel Results	HDL	Matlab
- 0.009590971	- 0.00958791	-0.0096
- 0.016015248	- 0.01601414	-0.0160
- 0.042226521	- 0.04221674	-0.0422
0.029477543	0.02947835	0.0295
0.02357878	0.02357588	0.0236
0.027000111	0.02699574	0.0270
-0.030991771	-0.03098638	-0.0310
0.034809779	0.03480579	0.0348
-0.068965023	-0.06896725	-0.0690
0.032532463	0.03252398	0.0325
-0.052697717	-0.05268674	-0.0527
0.017995923	0.01798982	0.0180
0.018044288	0.01804248	0.0180
-0.04901803	-0.049012098	-0.0490
0.044883505	0.04488705	0.0449

V. CONCLUSION

In this paper, we successfully designed a hardware model for the multipath channel model developed by Zimmermann and Dostert. The design is implemented as a pipeline structure with a controller serving as a timer for the entire process. The input uses 17 bits for the frequency value while every term generated in the summation series used 48 bits. Each term to be generated used up around 24 clock pulses mostly due to the timing needed by the CORDIC module for computing the sine/cosine values of the input angle. The HDL model developed shows promising result because it successfully synthesized to a hardware representation for a target FPGA library.

ACKNOWLEDGEMENT

The authors would like to thank the University Research Coordination Office (URCO) and Science Foundation of De La Salle University for funding this research.

REFERENCES

[1] M. M. Zimmermann and K. Dostert, "A multipath model for the powerline channel," *Communications, IEEE Transactions on*, vol. 50, no. 4, p. 553–559, Apr 2002.

[2] Cypress, "What is power line communication?" August 2011. [Online]. Available: http://www. eetimes.com/document.asp?doc id=1279014.

[3] H. Ferreira, L. Lampe, J. Newbury and T. Swart, Eds., Power Line Communications: Theory and Applications for Narrowband and Broadband Communications over Power Lines., John Wiley & Sons, Ltd, 2010.

[4] S. Galli, "A simplified model for the indoor power line channel," in *Power Line Communications and Its Applications, ISPLC 2009. IEEE International Symposium on*, March 2009.

[5] N. Weling, "Flexible FPGA-based powerline channel emulator for testing MIMO-PLC, neighborhood networks, hidden node or VDSL coexistence scenarios," in *Power Line Communications and Its Applications (ISPLC), 2011 IEEE International Symposium on*, April 2011.

[6] W. Liu, *Emulation of narrowband powerline data transmission channels and evaluation of PLC systems*, Karlsruher Instituts f¨ur Technologie, 2013.

[7] M. Gotz and K. Dostert, "A universal high speed powerline channel emulation system," in, 2002.," in *Broadband Communications Access, Transmission, Networking. 2002 International Zurich Seminar on*, 2002.

[8] M. Bauer, W. Liu and K. Dostert, "Channel emulation of low-speed plc transmission channels," in *Power Line Communications and Its Applications, 2009. ISPLC 2009. IEEE International Symposium on*, March 2009.

[9] F. Canete, L. Diez, J. Cortes, J. Sanchez-Martinez and L. Torres, "Time-varying channel emulator for indoor power line

communications," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM2008. IEEE*, Nov 2008.

[10] H. Eslami, S. Tran and A. Eltawil, "Design and implementation of a scalable channel emulator for wideband mimo systems," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 9, p. 4698–4709, Nov. 2009.

[11] S. Galli, A. Scaglione and Z. Wang, "For the grid and through the grid:: The role of power line communications in the smart grid,," *Proceedings of the IEEE*, vol. 99, no. 6, p. 998–1027, June 2011.

[12] A. Tonello, "Advances in power line communications and applications to the smart grid," in *European Signal Processing Conference*, Aug 2012.

[13] K. Kirkpatrick and C. Stimmel, "Power line communications for smart grids: PRIME, G3-PLC, IEEE P1901.2 proprietary OFDM, Proprietary N-PLC, and other narrowband solutions for utility NAN connectivity," Pike Research, Tech. Rep., 2012.

[14] W. Zhu, X. Zhu, E. Lim and Y. Huang, "State-of-art powerline communications channel modeling," in," in *Procedia Computer Science, First International Conference on Information Technology and Quantitative Management , pp. 563*–, 2013.