Time-Varying Channel Emulator for Indoor Power Line Communications

F.J. Cañete, L. Díez, J.A. Cortés, J.J. Sánchez-Martínez
Departamento Ingeniería de Comunicaciones
Universidad de Málaga (Spain)
Email: francis@ic.uma.es

Luis M. Torres
Design of Systems on Silicon (DS2)
C/ Charles Robert Darwin 2, Parc Tecnológic,
46980 Paterna, Valencia, SPAIN.
Email: luismanuel.torres@ds2.es

***Abstract*—In this paper, an FPGA-based channel emulator for indoor Power Line Communications (PLC) is described. It has been designed for a band up to 30MHz and is able to emulate time-varying channels and to generate a great variety of disturbances.**

I. Introduction

In the last two decades, a remarkable research effort has been put in PLC systems. These systems cover three main applications: *access*, the use of the low voltage power network as a last mile solution; *indoors*, the use of power grids inside small offices and homes as support for local area networks and *in-vehicle*, the use of on-board power grid for control and data transmission purposes. Nowadays, it is the second one that attracts more commercial interest, at least in the developed countries. In particular, this technology is very appropriate for the triple play service, i.e. to distribute telephone, internet access and digital video (that are provided by means of ADSL or cable) inside the consumer premises. This work is oriented to indoor broadband PLC systems and consists in the design and implementation of a channel emulator that helps in the development of future devices for this specific application.

Although the digital transmission over power lines is an old issue, it was mainly focused on low bit-rate applications that require little bandwidth, below 500kHz. It was not until the nineties when the characterization and modeling of PLC channels in a broader band was carried out. Since then, PLC channels started to reveal a quite interesting behavior. At the beginning, the research was centered on Linear Time Invariant (LTI) channel models [1][2], the simplest approach, but soon the time varying character of the channel was spotted [3]. However, regarding the noise characterization, since pioneer works, non stationary components were considered [4].

Other PLC channel emulators have been proposed [5] [6], but, to the authors' knowledge, this is the first one designed to implement a time-varying channel. In particular, the basis of this emulator is the *Cyclic Channel* model synchronous with the mains frequency described in [7].

The organization of the paper is as follows. The next section is a general description of the emulator characteristics, both of the hardware and the functionality implemented on it. The third section is devoted to explain the details of the programmed algorithms and in the fourth one conclusions are given.

This prototype is the result of a joint project between DS2 (Design of Systems on Silicon) and the Departamento Ingeniería de Comunicaciones - Universidad de Málaga, supported by DS2.

II. Emulator Description

The channel emulator here presented consists of a hardware platform and a software application running in a host PC. The hardware is an FPGA-based card for digital signal processing that includes Analogue to Digital Converters (ADC) and Digital to Analogue Converters (DAC). The software application has been developed in Java and C and its main purpose is the configuration of the hardware parameters. It also manages data bases that contains behavioral parameters of the channels, which will be later loaded in the emulator. In the remaining of this paper, unless otherwise noted, the term emulator will be exclusively used to refer to the system programmed in the hardware part.

A. Hardware Platform

The main elements included in the hardware platform are:

- Virtex-4 FPGA XC4VSX35-10FF668 by Xilinx [8], for signal processing and management purposes
- Host PC interfacing via PCI or USB v1.1 interfaces
- Spartan-II FPGA by Xilinx, only for interfaces management
- Two independent input channels with the ADC AD6645 by Analog Devices (14 bits up to 105 MSPS)
- Two independent output channels with the DAC AD9772 by Analog Devices (14 bits up to 160 MSPS)
- Two banks of memory ZBT-SRAM (133MHz, 512Kx32 bits per bank)

The anti-aliasing filter used in the converters is a third order low-pass filter with a cutoff 3dB frequency of 58MHz and attenuation level of 60dB at 350MHz. In the current implementation, the channel emulator only employs one ADC and one DAC.

B. Emulator Overview

The main features of the emulator can be summarized as follows:

- time-varying channel filtering

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

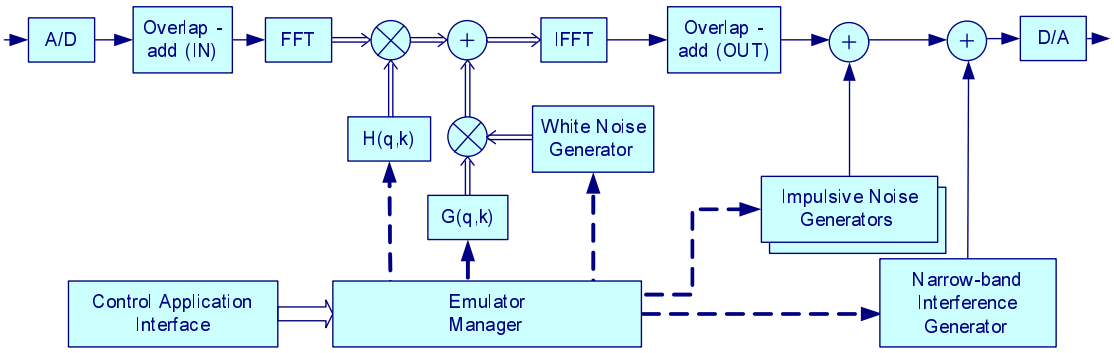

Fig. 1. Emulator Scheme

- cyclostationary noise generation
- impulsive noise generation
- narrow band interference generation
- sampling frequency of 100MHz
- approximate dynamic range of 70dB
- useful band up to 30MHz
- configurable mains period value, to be adapted to PLC in countries with 50Hz/60Hz AC.

C. Functional Description

The structure of the channel emulator can be seen in fig.1. The main functional modules are: the channel filter, the cyclostationary noise generator, the impulsive noise generators, the interference generator and the emulator manager. The latter controls the timing of the different modules according to the configuration loaded in the emulator memory (in the so-called Control Application Interface) by the Control Application.

All the algorithms included in the design for signal processing purposes have been developed ad-hoc (programmed in Verilog), but the FFT and IFFT algorithms have been implemented by means of Xilinx LogiCore [9].

D. Disturbances classification

The disturbances considered in this work are a variation of the conventional classification of noise in PLC channel characterization [10]. The emulated components are,

- *Cyclostationary noise*. It is colored Gaussian noise that generally exhibits a decay of level in its instantaneous power spectral density (PSD) as frequency increases.
- *Impulsive noise*. Three classes of impulsive noise are distinguished: *synchronous*, which is periodic and synchronous with the mains period; *asynchronous*, which is periodic but not synchronous with the mains period (although its parameters usually exhibit a cyclostationary behavior along the mains period); and *sporadic*, which is not periodic and appears randomly.
- *Narrow-band interference*. It models the coupling of radiofrequency signals, either from commercial broadcasting, amateur radio or oscillators inside consumer electronics devices next to the receiver.

As described later, the strategy to create these classes of disturbances is different. The cyclostationary noise generation is performed in the frequency domain by means of time-varying filtering of white Gaussian noise. However, the impulsive noise and the narrow-band interference generation is made in the time domain by direct digital synthesis.

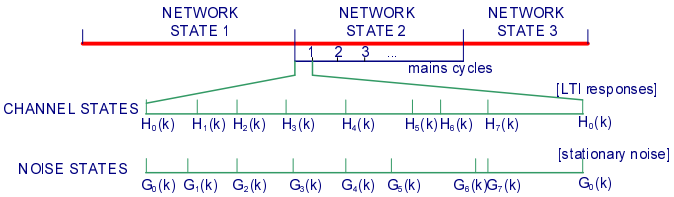

Fig. 2. Modeling procedure of the time variation

E. Time variation

In the emulator design, two levels of channel time variation have been considered:

- *Long-term changes*, among which the power network is assumed to be invariant, i.e. electrical devices keep their working state. This situation constitutes what is called a *network state*. From a network state to the following one an ideal transition is suppose to occur, what means an abrupt change of channel.

- *Short-term variation*: inside each network state, the channel is considered LPTV (Linear Periodically Time-Varying) and part of the noise cyclostationary as indicated before. Both features are modeled with the *channel states* and the *noise states*. This temporal organization is depicted in fig.2.

Following a practical approach, the channel time-varying filtering is based on the slow-variation approximation of an LPTV channel [7]. On one hand, the channel is assumed underspread, i.e. its coherence time is larger than the effective duration of its impulse response. On the other hand, the variation is slow enough so that the channel variant response can be sampled few times in a mains cycle. According to this, the channel evolution has been represented with several snapshots of the time-varying channel response which are taken at different instants of the mains period (not necessarily equally spaced). The same idea is applied to the instantaneous PSD of the cyclostationary noise, which varies along the mains period. These snapshots constitute the channel states (LTI) and

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

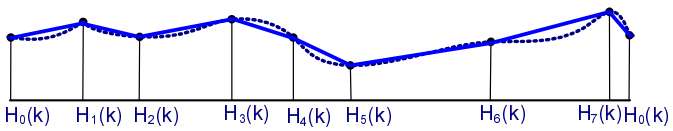

Fig. 3. Sampling of the channel time-varying response

the noise states (stationary), respectively. As seen in fig.2, just eight states per mains cycle have been contemplated.

III. Detailed Design

In this section, the implementation of the different functional modules is described.

A. Channel Filtering

The signal filtering is carried out by using block convolution in the frequency domain with the overlap-add method [11]. This implementation is more efficient than the use of a convolution in the time domain. For this purpose, an FFT (Fast Fourier Transform) of $N = 4096$ samples is performed. The maximum channel impulse response length is fixed to L =1536 samples (that represents 15.36μs at 100MHz sampling rate), which implies that the number of samples from the input signal per FFT block is $M = N - L = 2560$ samples. The number of FFT blocks per mains period results in $Q = 782$ or 652, depending on the mains frequency (for 50 or 60Hz respectively). Hence, this procedure establishes a time structure in frames or FFT blocks and an integer relation between the mains period and the frames duration is forced to guarantee the periodicity.

As mentioned, the variation of the channel is modeled with Z=8 channel states per mains cycle. These states do not need to be uniformly distributed in time along the mains period, but each of them must last an integer number of frames. Hence, the data base of channel behavior parameters will contain the channel responses and the duration of each state.

An expansion with a linear interpolation is performed to these channel states in order to implement the varying filtering with a time-frame resolution. This idea is illustrated in fig.3. The slope represents an increment, denoted by ΔH_i, used to interpolate the i-th channel state and is frequency selective. This increment is calculated from the values of the channel state duration, x_i (measured in number of frames), its frequency response and the one of the following state. That is,

$$\Delta H_i(k) = \frac{H_{i+1}(k) - H_i(k)}{x_i} \tag{1}$$

The linear interpolation is performed according to,

$$H[q,k] = H[q-1,k] + \Delta H_i(k) \tag{2}$$

where, $i = 1 \ldots Z$ is the channel state index; $q = 1 \ldots Q$ is the frame or FFT-block index and $k = 1 \ldots N$ is the frequency index. The initial value of $H[q = 0, k] = H_0(k)$ is the first snapshot of the channel response. As result, a sequence of Q invariant states per mains period is obtained and a periodical evolution is assumed, i.e. the state zero appears again at the end of the sequence. Consequently, there are hundreds of locally invariant states per mains period, because at each frame the channel filter updates its frequency response.

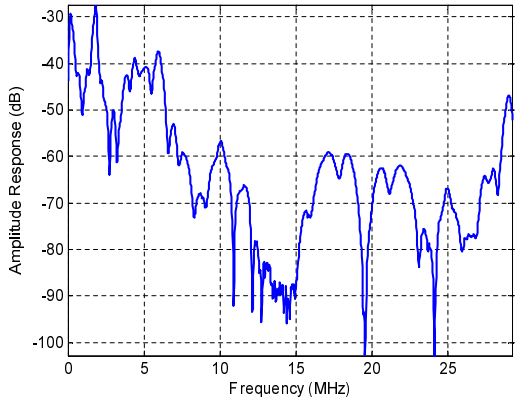

Fig. 4. Mean amplitude response for an example of channel response

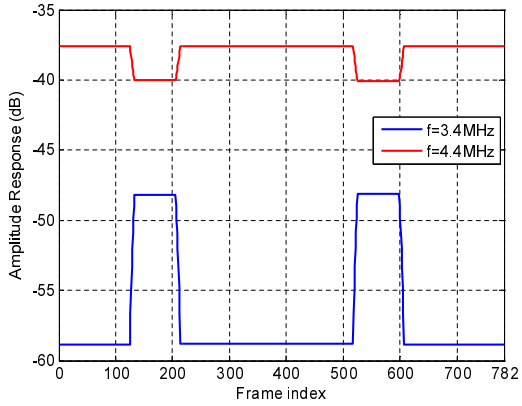

Fig. 5. Time evolution, in frames, of the amplitude response at two frequencies for the previous channel

This is an operations-efficient implementation of the filtering in the frequency domain that avoids the need for multiplications. The increment at each frequency is maintained during each of the Z channel states and the channel response is accumulated at each frame.

The next two figures are included as an implementation example of the channel time-varying behavior. The selected channel response has been taken from measurements at an apartment and included in the emulator. In fig.4, the mean amplitude response of this channel, averaged in time, is plotted. In fig.5, two certain frequencies of the channel have been selected and their amplitudes evolution along the mains cycle are shown. A quite common feature is observed: a two-state behavior with a periodicity of double the mains frequency. This data is loaded in the emulator and, when it is excited with an input sinusoid (e.g. at any of these frequencies), the output signal will exhibit a sort of modulation due to the channel filtering. This can be seen also in the frequency domain, as a spectral broadening due to the channel Doppler effect.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

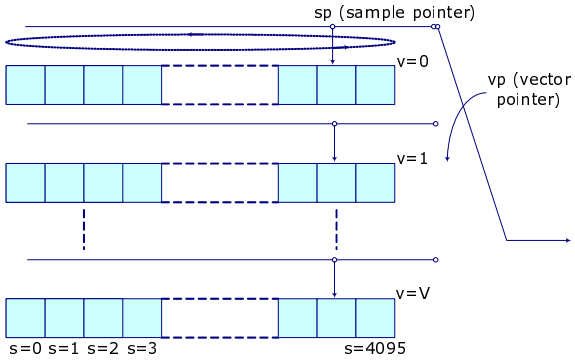

Fig. 6. White noise generator scheme. Several vectors of noise samples in frequency domain are addressed by a random pointer.

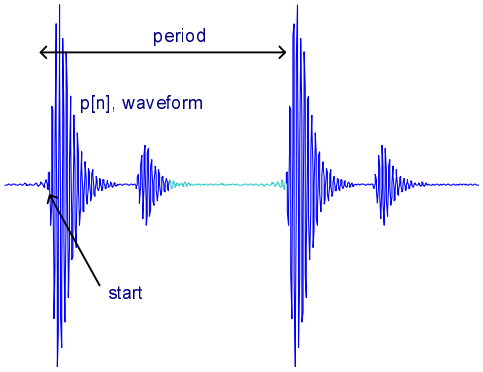

Fig. 7. An example of impulsive noise

B. Cyclostationary Noise Generation

In the channel emulator, the cyclostationary background noise is obtained from a pseudo-random *white noise generator* performed in the frequency domain. The generator contains a set of noise vectors of length N, whose complex samples are the FFT of M samples of white Gaussian noise padded with L zeros, i.e. as processed by the overlap add method.

To address the N samples to be read from the bank at each FFT-block, a Linear Feedback Shift Register (LFSR) is employed. The LFSR is implemented with a Fibonacci structure of nineteen stages [12]. The output of the LFSR is used to: randomly select one of the stored vectors (*vp* pointer) and to select a random initial position (*sp* pointer) from which to apply a circular shift (of length N). The scheme is depicted in fig.6.

The values of the noise samples and the polynomial of the LFSR have been selected to get the most Gaussian and uncorrelated character of the generated noise.

In order to obtain the cyclostationarity of the noise, synchronized with mains period, an approach analogous to the signal time-varying filtering is employed. The noise PSD is sampled Z times in a mains period, at instants around which it can be assumed approximately stationary. At each of these noise states, a different filter is applied to color the white noise source. The filter frequency response is calculated from the desired PSD [13]. A time-varying filtering is constructed based on these noise filters responses also with a linear interpolation among them,

$$G[q,k] = G[q-1,k] + \Delta G_i(k) \tag{3}$$

The indices have the same meaning as in (2) and the increments are calculated as described in (1).

C. Impulsive Noise Generation

As mentioned, the impulsive noise generation is performed in the time domain by direct digital synthesis of pulses with different waveforms and inter-arrival times. The same generator structure is used for the three classes of impulsive noise, but with a different parameter selection.

The generator have three configurable parameters: *period*, the inter-arrival time between impulses, $p[n]$, the impulse waveform and *start*, which allows to control the impulses timing. The waveforms are, in all cases, selectable from a data base of pulses obtained from measurements. In fig.7, an example of these parameters on a impulse waveform that corresponds to asynchronous impulsive noise, can be observed.

The settings for the three classes of impulsive noise considered are described below.

1) Synchronous Impulsive Noise: The *period* is half a mains cycle, to model the common case of a double mains frequency periodicity (i.e. 100Hz in Europe). The *start* time is used to place the impulses along the mains cycle at any sampling period. These parameters can be set on a network state basis.

2) Asynchronous Impulsive Noise: The *period* is selectable, and allows to have a wide range of repetition rates, the default one is [20kHz-500kHz] (to cover the frequencies commonly used in switched power supplies). The period and the waveform can be set on a noise state basis, so at different noise states they can be changed. The *start* time is superfluous in this case, because the impulses are considered to start at the beginning of the noise state.

3) Sporadic Impulsive Noise: Since this kind of impulses are aperiodical, the *period* is superfluous. The *start* time is selectable at any sampling period of any mains cycle. These parameters can be set on a network state basis.

D. Narrow-band Interference Generation

The narrow-band interference generator is implemented by means of a pulse-amplitude (PAM) modulator. Binary modulation is used with pulses that have a pass-band waveform. The generated signals have a selectable bandwidth (the minimum value is around 1kHz) and center frequencies. The scheme of the generator is plotted in fig.8.

The parameters of the generator are:

- *Pulse waveform*, $p[n]$. A rectangular pass-band pulse is assumed, with a certain carrier frequency and amplitude, or group of them (this way several interference terms are

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

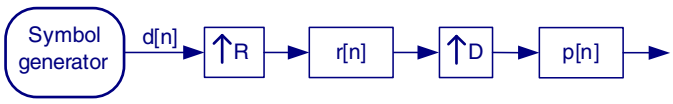

Fig. 8. PAM modulator scheme

generated simultaneously). That is,

$$p[n] = \sum_i A_i sin[\omega_i n], \quad 0 \leq n \leq D-1 \tag{4}$$

$$\omega_i \in \{\text{set of carrier frequencies}\}$$

$$A_i \in \{\text{set of amplitudes}\}$$

- *Pulse duration*, D, in number of samples. It must contain an integer number of periods for all carriers, to guarantee a continuous phase transition between pulses.

- *Modulating data*, $d[n]$. It is a random sequence of binary symbols {+1,-1} that modulates the pulses. It is fixed, stored in the emulator memory and is periodically used.

- *Repetition factor*, R. It is an interpolation factor of the random symbols to extend the effective pulse duration and consequently to reduce the interference bandwidth. It leads to a more efficient implementation, saving memory. The repetition factor is implemented by means of an expansion with a rectangular window $r[n]$=1, 0$\leq n \leq R$-1.

E. Mains frequency jitter

It is interesting to emulate the jitter commonly exhibited by the mains frequency. For this purpose, a vector that represents a sequence of relative jitter is stored in the emulator memory. The resolution in time of this mains jitter is of one frame, or FFT block, to guarantee that mains period always has a multiple integer of frames. The jitter vector has ternary values {0,+1,-1} that indicate whether the current mains cycle contains the nominal value of frames, Q=782 or 652, one frame more or one frame less. This leads to a range of variation over the nominal mains frequency, 50 or 60Hz, of $\pm 0.13\%$ or $\pm 0.15\%$ respectively. This jitter is applied to the last channel state and noise state of each mains cycle. The jitter sequence can be modified by the user to emulate different jitter ensembles.

F. FPGA Utilization Summary

The explained design, once synthesized in the Virtex-4 FPGA (and after mapping and place & route procedures), has reached the degree of device utilization summarized in table I. The indicated resources refers to: slices (configurable logic blocks), LUT (Look-Up Tables), memory blocks, DSP (Digital Signal Processing) slices and DCM (Digital Clock Managers). The total equivalent gate count for the design has resulted in 10,789,057.

TABLE I
FPGA RESOURCE UTILIZATION

occupied slices	83%
4-inputs LUTs	62%
RAM blocks	81%
DSP slices	41%
DCMs	50%

IV. CONCLUSION

In this paper, a channel emulator for broadband PLC channels has been described. In the design, a comprehensive channel model has been used that includes a channel time-varying filtering with short-term and long-term dynamics. Different components of noise and narrow-band interference are contemplated as well. The algorithms of the different functional modules programmed in the FPGA have been explained to outline its working principle. Finally, the resource utilization in the device has been summarized. This channel emulator will be very useful to test modem prototypes and to assist in the development of new transmission techniques.

Although the prototype is focused on indoor PLC channels, it could be adapted to outdoor channels as well. However, some parameters should be modified and some algorithms could require a re-design. For instance, the length of the channel impulsive response considered (15.36 μs) should be extended.

Additionally, as the hardware platform has two DACs and ADCs, a future work would be to use the remaining converters to implement a return path. That way, a full-duplex transmission could be performed in the emulator on the same hardware.

REFERENCES

[1] H. Philipps, "Performance measurements of power line channels at high frequencies," in *International Symposium on Power-Line Communications and its Applications (ISPLC)*, 1998, pp. 229–237.

[2] ——, "Modelling of power line communication channels," in *International Symposium on Power-Line Communications and its Applications (ISPLC)*, 1999, pp. 14–21.

[3] F. Cañete, J. Cortés, L. Díez, and J. Entrambasaguas, "Modeling and evaluation of the indoor power line channel," *IEEE Communication Magazine*, vol. 41, pp. 41–47, Apr 2003.

[4] O. Ohno, M. Katayama, T. Yamazato, and A. Ogawa, "A simple model of cyclostationary power-line noise for communication systems," in *International Symposium on Power Line Communications and Its Applications (ISPLC)*, 1998.

[5] G. Bumiller, "Power-line analysing tool for channel estimation, channel emulation and evaluation of communication systems," in *International Symposium on Power-Line Communications and its Applications.*, 1999.

[6] H. Philipps, "A hardware fading simulator for poweline communication channels," in *International Symposium on Power-Line Communications and its Applications (ISPLC)*, 2001, pp. 241–246.

[7] F. Cañete, J. Cortés, L. Díez, and J. Entrambasaguas, "Analysis of the cyclic short-term variation of indoor power-line channels," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 7, pp. 1327–1338, July 2006.

[8] Xilinx, "Virtex-4 family overview. preliminary product specification," www.xilinx.com, Tech. Rep., 2006.

[9] ——, "Xilinx logicore. fast fourier transform v4.1. product specification." www.xilinx.com, Tech. Rep., 2007.

[10] M. Zimmermann and K. Dostert, "Analysis and modeling of impulsive noise in broad-band powerline communications," *IEEE Trans. on Electromagnetic Compatibility*, pp. 249–258, Feb 2002.

[11] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[12] R. Peterson, R. Ziemer, and D. Borth, *Introduction to Spread Spectrum Communication*. Prentice Hall, 1995.

[13] W. Gardner, *Introduction to Random Processes*. MacMillan, 1986.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.