

Channel Emulation of Low-Speed PLC Transmission Channels

Michael Bauer, Wenqing Liu, Klaus Dostert
Institute of Industrial Information Technology
Universität Karlsruhe (TH)
Karlsruhe, Germany

Abstract—Test and verification of the physical layer of power line communication systems are very difficult to accomplish, because the mains grid exhibits stochastic and time variant behavior. In this paper, we propose a channel emulator that allows for emulating a freely configurable PLC channel scenario in the frequency range between 9 kHz and 500 kHz. The emulator can be deployed for system verification during the design and development phase as well as for standardization purposes.

Keywords—power line transmission channel, channel emulation, smart metering

I. INTRODUCTION

PROPER system verification procedures are mandatory to every successful development. In the case of power line communication (PLC), it is very difficult to verify a modem, since parameters of the PLC transmission channel that may be crucial to system reliability expose a highly time-variant behavior. Therefore, tests on communication links via live mains grids lack the statistical significance that would be necessary to prove the reliability of a PLC modem. In addition, the sources of failures in complex systems, such as PLC modems, can hardly be traced under realistic scenarios as they are encountered on a live mains grid.

PLC technology seems to gain importance in the context of automatic meter reading (AMR) or even advanced metering infrastructure (AMI) services. The directive on energy end-use efficiency and energy services (2006/32/EC) formally adopted by the EU council in early 2006 advises “intelligent metering systems” as an energy efficiency improvement measure that will result in energy savings. AMR and even more so AMI systems depend on highly reliable and available communications networks. A great number of distributed meters must be managed remotely, thus data must be transferred reliably and securely between spatially separated points.

At the same time, there is no public regulation or standardization body that evaluates and certifies technology available on the market. Customers need to trust technology providers, which is especially difficult regarding the high investment cost at stake.

Addressing the need for test and verification equipment, we propose a PLC transmission channel emulator in this paper. By means of this emulator it is possible to emulate a freely

configurable PLC scenario including access impedance, channel transfer function and noise scenario in a reproducible manner. The currently envisaged frequency spectrum for metering services in Europe ranges from 9 kHz up to 95 kHz (CENELEC A band) [1], therefore we provide an overview of the channel characteristics in this frequency range. Outside the EU, frequencies up to 500 kHz are used for such services. Therefore, our channel emulator covers a frequency range from 3 kHz to 500 kHz.

Our paper comprises six sections. In Section II, we describe the channel model that the emulator is based on. Section III contains the overview of the relevant channel characteristics mentioned before. The schematic structure of the channel emulator is laid out in Section IV, and the evaluation of the overall emulator is presented in Section VI.

II. LOW-SPEED PLC TRANSMISSION CHANNEL MODEL

In order to provide an exhaustive description of the PLC channel properties for test and verification of a modem, we need to distinguish between physical channel properties and channel properties related to signal processing.

A. Signal Processing Channel Model

The model in Fig. 1 has been proposed e.g. in [2].

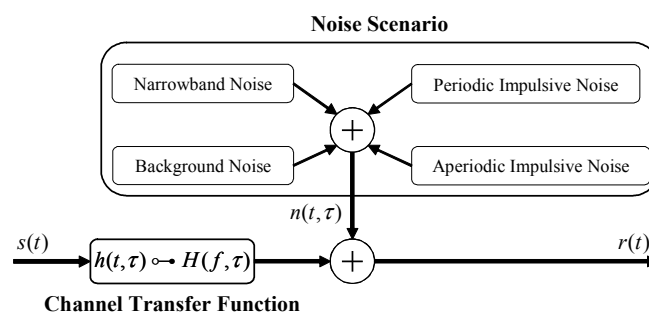


Fig. 1. Signal Processing Model of a PLC Channel

The transmit signal $s(t)$ is affected by the channel impulse response $h(t, \tau)$ that is generally time variant. The receive signal is then obtained by adding time variant noise signals $n(t, \tau)$ to the modified transmit signal $s(t) * h(t, \tau)$.

Neglecting the time variance for the sake of clarity, the relation between transmit signal and receive signal is described by

$$r(t) = s(t) * h(t) + n(t). \quad (1.1)$$

Equation (1.1) is the most fundamental equation describing a PLC transmission channel and can be implemented digitally.

B. Physical Channel Model

So far, we have considered the effects of the PLC channel on the transmit signal from a signal theoretic perspective. The physical properties of the mains grid, however, have significant influence on PLC transmit signals at low frequencies in the order of kHz. Due to the low signal frequencies, the injection of transmit signals into the power line can only be achieved by capacitive signal coupling. Therefore, the access impedance of the low voltage transmission line has significant impact on the transmit signal. It is thus necessary to extend the emulation of the signals provided in Fig. 1 by including a model of the access impedance as proposed in [5]. As can be seen in Fig. 2, this model covers the same features as the model in Fig. 1, but

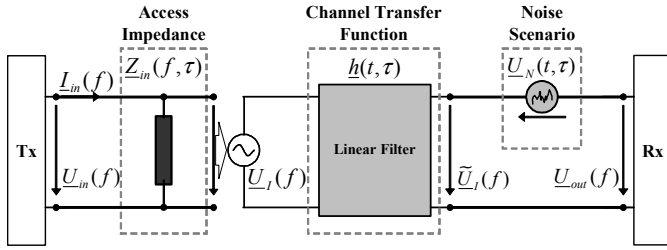


Fig. 2. Physical Channel Model

extends the channel model by the access impedance $Z_{in}(f, \tau)$. This impedance is presented to the power output stage of a PLC modem at the transmitter side (Tx in Fig. 2) by the mains grid. Since the modulus of this impedance can be challengingly low, especially at low frequencies, the correct design of a modem's power output stage states a crucial design requirement. It is indispensable for tests and is therefore included with the design of our emulator.

III. Assumptions on the Channel Characteristics

To date, the properties of the low-speed PLC transmission channel have been analyzed in various publications, e.g. [3] and [4], [5], and [6]. [5] focuses on the in-house domain, but many of the findings can be transferred to the access domain. In the following, we provide an overview of the properties of the transmission channel in the CENELEC A band:

A. Access Impedance

As described in [5], the access impedance is generally complex-valued and exhibits non-linear behaviour. The modulus of the access impedance can assume very low values, in extreme cases even below 1Ω [6].

B. Channel Transfer Function

Due to the relation between the length of a transmission line and transmit signal wavelength, we assume that signal reflections and scattering do not severely affect the quality of the transmit signal. This leads to the assumption that the channel transfer function of the PLC channel will most likely not exhibit sharp notches. Frequency selectivity of the channel transfer function only plays a minor role. However, as described in [6] and [5], transmit signals may be heavily influenced by strong attenuation, even more so, if the receiver is coupled to a different phase of the three-phase mains grid than the transmitter. The attenuation may reach up to -60dB.

C. Noise Scenario

As in the channel model in Fig. 1, we generally differentiate between several kinds of additive noise, i.e. colored background noise, periodic impulsive noise (either synchronous or asynchronous to the mains frequency) and aperiodic impulsive noise. Measurements show that the power spectral density of the colored background noise decreases significantly [5]. So far, narrowband noise has not been discovered in the access domain, neither periodic impulsive noise. However, there is severe aperiodic impulsive noise. Such pulses exhibit only sporadic occurrence in the order of minutes rather than seconds, and the pulse amplitudes may reach up to peak voltages of even more than 15V. Although it is rather difficult to specify the parameters of such short pulses exactly, measurements reveal waveforms according to

$$n_{imp}(t) = \underbrace{A_0 \cdot \exp(-d \cdot t)}_{x_0(t)} \cdot \underbrace{B_0 \sin(2\pi f_0 t)}_{y_0(t)} \quad (1.2)$$

as the predominant waveform for impulsive noise. Fig. 3 shows a selection of typical waveforms with three such impulses.

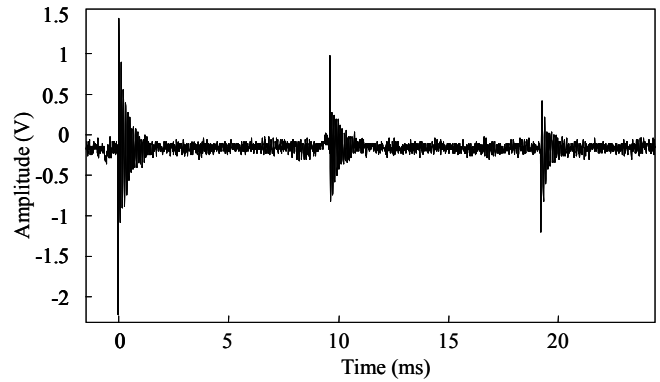


Fig. 3. Measured Impulsive Noises

IV. CHANNEL EMULATOR IMPLEMENTATION

So far, the general channel model for low-speed PLC transmission channels has been described. The low-speed channel emulator implements all of the named features of the PLC channel. We first describe the general structure of the channel emulator. Afterwards, the structure of the digital logics will be described.

A. Hardware Design

The hardware of the channel emulator consists of five modules: a mains access stage, an input stage, an analog front end, an output stage and a FPGA board, as shown in Fig. 4.

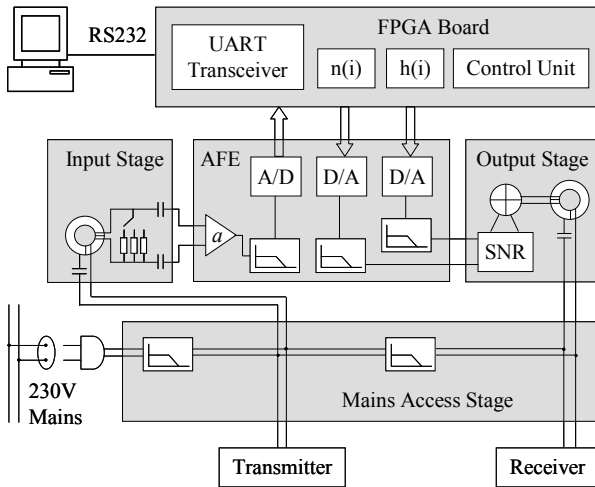


Fig. 4. Block Diagram of Emulator Hardware Implementation

Most PLC modems obtain the power supply directly from the mains network, so they need to be connected to the mains power directly. This also allows for testing modems that rely on the zero-crossing point of the mains voltage for frame synchronization, as proposed in [7]. The emulator described in this paper offers an access interface to the devices under test (DUT) that imitates the original mains access stage. Both, transmitter and receiver, can thus be tested under realistic conditions.

The mains voltage for the *Mains Access Stage*, e.g. 230V, can be obtained from an ordinary wall socket. We use one low-pass filter with a very low cut-off frequency to protect the test setup from unwanted noise and disturbance from the mains network. A second low-pass filter is used for ensuring that transmit signals do not directly pass from transmitter to receiver.

The input stage comprises two modules, a coupling circuit and a collection of five resistor arrays. While the coupling circuit blocks the 230V mains voltage and couples the transmit signal into the emulation system, the resistor arrays emulate the modulus of the access impedance $|Z_{in}(f)|$. Each of the resistor arrays is connected to the input through an electronically controlled relay. The switching state of the relays determines the modulus of the access impedance presented to the transmitter. Designing the emulator in this way forces the power output stage of the DUT to drive loads that are as low as they would be in the case of a real connection to the mains grid.

The field programmable gate array (FPGA) board is equipped with an FPGA that realizes the logic design in VHDL. We use the FPGA for emulating the channel transfer function, for generating the complete noise scenario $n(t, \tau)$ and for controlling components in other modules.

The *Analog Front End* (AFE) converts the incoming analog signal to digital words and conveys them to the FPGA. Having been processed by the digital filter, the digitized transmit signal is sent back to the AFE and reconstructed to an analogue signal by a digital-to-analog converter (DAC). The emulated noise scenario generated by the FPGA is also conveyed to the AFE and converted to analogue waveforms by another DAC device. At the output stage, the signal and the emulated noise signal each pass through an independent variable gain amplifier (VGA). The amplification factors of the VGA are configured digitally by the control unit on the FPGA. The signal outputs of the VGAs are then added together and coupled out to the receiver. This allows for implementing a configurable signal-to-noise ratio (SNR). The output stage houses a coupling circuit of the same structure as the coupling circuit at the input stage. We use a PC with an easy-to-use graphical user interface (GUI) for controlling all channel emulator parameters and functions via the FPGA.

B. Digital Design

As mentioned above, we emulate the PLC channel and the noise scenario digitally by running VHDL code on the FPGA. Fig. 5 shows the top-level design.

According to Equation (1.1), the channel transfer function is implemented by convolving the input signal $s(t)$ (circled 1) and the channel impulse response $h(t, \tau)$. Since the envisaged application of the channel emulator is the verification of modems under predefined scenarios, we neglect the time variance of the channel transfer function, thus $h(t, \tau) = h(t)$. The convolution is carried out by a discrete-time finite impulse response (FIR) filter $h_1(i)$, where the discrete impulse response sequence is stored as coefficients of the filter. It is either possible to use a measured channel impulse response for emulation or to define a custom amplitude response graphically by means of the PC-based GUI. In the latter case, the impulse response is calculated by taking the inverse Fast Fourier Transform (IFFT) of the amplitude response in the emulator console, and then conveyed to the FIR filter via the serial communication interface (circled 2). Since the convolution yields results at a higher bit resolution than can be processed by the DAC, the *Bit Matching* module extracts the most relevant bits from the convolution result (circled 3).

As described in Section II, the noise scenario consists of different types of noise. Despite the fact that not all kinds of noise are relevant to low-speed PLC channels, the channel emulator offers the possibility to emulate all known classes of noises. TABLE I lists the correspondence between emulated noise types and VHDL modules.

Component *WN* generates 12-bit random numbers representing white noise with the help of PN-Sequences. The colored background noise is generated by passing the white noise through a second FIR filter $h_2(i)$ in component *BGN*.

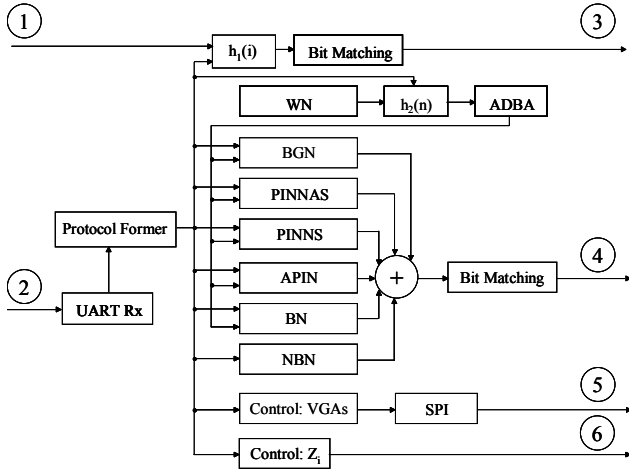


Fig. 5. Block Diagram of Digital Functions

Component *ADBA* matches the output of the filter automatically so that the colored noise is always full-scaled, independent of the current filter coefficients of $h_2(i)$. Component *NBN* emulates narrowband noise by utilizing direct digital synthesis (DDS). Components *PINNAS*, *PINNS*, *APIN* and *BN* control the appearance, pulse width as well as amplitude of impulsive noises. The control algorithm details can be found in [8]. A sum function adds all emulated noises together. The *Bit Matching* module behind the sum function extracts the most relevant bits out of the sum to match the bit width of the DAC (circled 4). Component *Control: Zi* controls the modulus of the access impedance at the input stage by delivering five-bit control signals (circled 6). Component *Control:VGAs* configures the SNR at the output stage over an SPI interface (circled 5). The convolution modules $h_1(i)$ and $h_2(i)$, the control units and the modules for noise scenario are configured over a configuration bus. The component *UART Rx* de-multiplexes the serial bit stream to bytes. *Protocol Former* assembles these bytes according to a predefined protocol and launches them on the configuration bus.

TABLE I
VHDL COMPONENTS AND CORRESPONDING NOISE TYPES

Component	Emulated Interference
<i>WN</i>	white noise
<i>BGN</i>	colored background noise
<i>PINNAS</i>	periodic impulsive noise asynchronous to mains frequency
<i>PINNS</i>	periodic impulsive noise synchronous to mains frequency
<i>APIN</i>	aperiodic impulsive noise
<i>BN</i>	burst noise
<i>NBN</i>	narrowband noise

C. Implementation of Impulsive Noises

Two types of impulsive noise are often observed in measurement practice. The one type features stochastic waveforms. The appearance of such an impulse boosts the

power of the background noise shortly. It can be realized by switching the full-scaled background noise on and off according to the control pulse sequence. The detailed implementation can be found in [8]. The other type can be described by an exponentially decaying oscillation. As shown in equation (1.2) there are three parameters characterizing the waveform: decay factor d , maximal amplitude $C = A_0 \cdot B_0$ and frequency of the oscillation f_0 . The decay factor d decides how quickly the oscillation drops and so determines the pulse width. Fig. 6 shows the principle of the implementation. A pattern of exponential decay and one cycle of a sinusoidal signal are stored as look-up tables. The access to both look-up tables are controlled by address steps Δm and Δn respectively. The discrete-time samples \underline{x}_0 and \underline{y}_0 are the respective outputs of both look-up tables. Element-wise multiplication of both sequences delivers the exponentially decaying oscillation

$$\underline{n}_{\text{Imp}} = \underbrace{A_0 \cdot e^{-d_0 \cdot (\Delta m \cdot T_s)}}_{\underline{x}_0} \cdot \underbrace{B_0 \cdot \sin(\Delta n \cdot T_s)}_{\underline{y}_0}, \quad (1.3)$$

Where T_s is the period of the clock at which samples are read out.

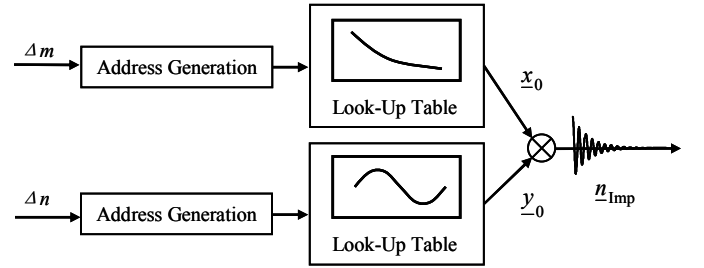


Fig. 6. Emulation of Impulsive Noise with Exponentially Decayed Waveform

If we rewrite Equation (1.3) as

$$\underline{n}_{\text{Imp}} = A_0 \cdot e^{-d_1 T_s} \cdot B_0 \cdot \sin(2\pi \cdot f_1 \cdot T_s), \quad (1.4)$$

where $d_1 = d_0 \cdot \Delta m$, and $f_1 = f_0 \cdot \Delta n$, we can see that changing Δn and Δm results in a variation of the sampling period, which can also be interpreted as an alteration of the decay factor and of the frequency of the oscillation respectively. In this way, we can emulate impulses with different decay factor and oscillation frequencies.

V. VERIFICATION

A. Access Impedance

The access impedance is measured with a vector network analyzer (HP 8753E RF). Fig. 7 shows one of the emulated impedances. The first plot shows the modulus $|Z(f)|$.

It begins with 8.5Ω at 30 kHz, drops to its local minimum of about 0.73Ω at 67.6 kHz rapidly, and then rises to about 35.4Ω at 500 kHz monotonously. The second and third plot show the real and imaginary part, respectively.

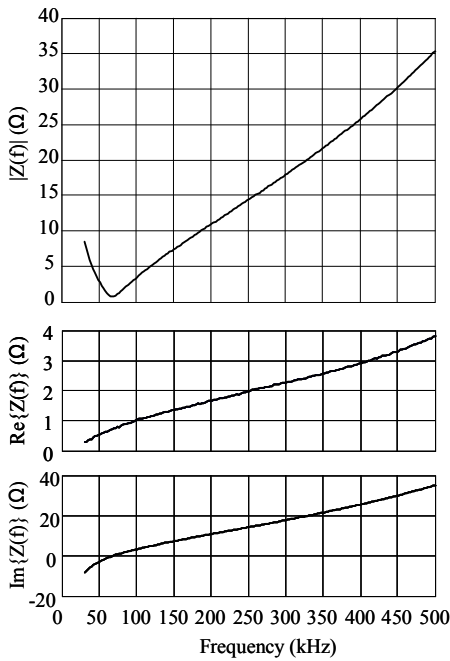


Fig. 7 Emulated Access Impedance

The real part has a rising curve with neither local minima nor local maxima, ranging from 0.3Ω to 3.8Ω . The imaginary part features a zero at the same 67.6 kHz at which the modulus changes its polarity from negative to positive. The imaginary part has much larger values than the real part, contributing dominantly to the absolute impedance.

B. Channel Transfer Function

The channel transfer function is also verified by means of a network analyzer (HP 8753E). Fig. 8 shows the amplitude response of an example channel transfer function.

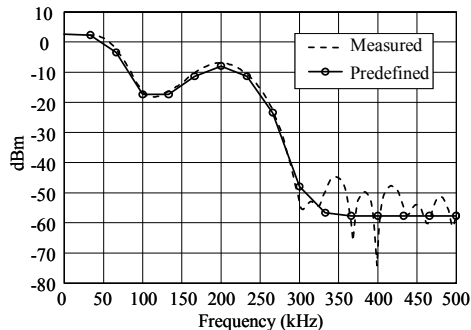


Fig. 8 Predefined and Measured Amplitude Response

The circled solid line represents the amplitude response specified by means of the GUI. It begins with 2.4 dBm at frequency 0 , drops to a local minimum of -17.5 dBm at about 133.3 kHz, reaches the next peak of about -8.1 dBm at 200 kHz, and stays below -50 dBm between 300 kHz and 500 kHz. The dashed line represents the measured result, which agrees very well with the desired amplitude response. Sidelobes between 300 kHz and 500 kHz result from the finite number of samples (60 points) of the discrete impulse response. If necessary, they

can be reduced by applying proper windows, such as Hamming or Blackman.

C. Noise Scenario

As mentioned before, the PLC transmission channel features different classes of noises. The channel emulator allows to generate all different classes at the same time. For the sake of convenience, however, their verifications are introduced separately.

The background noise and the narrowband noise are verified with a spectrum analyzer (R&S® ESPI 3). Fig. 9 displays the noise spectrum generated by the channel emulator at $50, 70, 100, 120, 200, 220, 300, 350$ and 400 kHz, which corresponds very well with the frequencies specified in the control GUI.

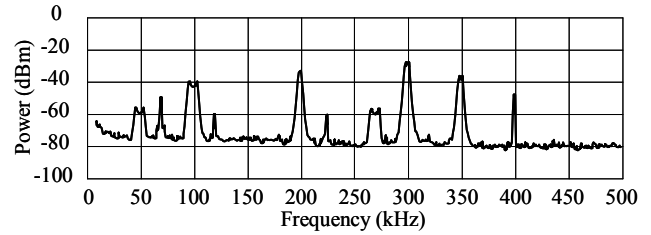


Fig. 9 Spectrum of background noise and narrowband noise

Both types of emulated impulsive noise waveforms are depicted in Fig. 10. They have first been captured by means of an oscilloscope. Their spectra have then been calculated offline by means of the Fast Fourier Transform. The upper plot of Fig. 10 (a) illustrates a single pulse of 150 μ s where the waveform is an exponentially decaying harmonic oscillation. Its spectrum is shown in the upper plot of Fig. 10 (b). There is an obvious peak at 100 kHz corresponding to the frequency of the oscillation.

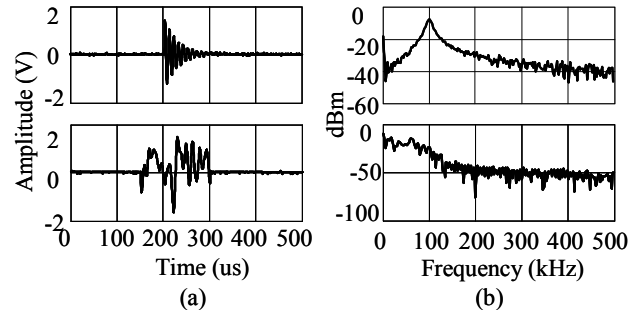


Fig. 10 Waveforms and Spectra of Impulsive Noises

The lower plot in Fig. 10 (a) shows a pulse with stochastic waveform of the same duration of 150μ s. Its spectrum is shown in the lower plot in Fig. 10 (b). Since this kind of noise is implemented by boosting the colored background noise for a while, it does not have a predictable waveform. The power spectrum of the background noise is raised significantly at the moment the noise waveform appears, but keeps its general characteristics over frequency.

Fig. 11 shows an example of emulated periodic impulsive noise. It exhibits a repetition rate of 100 Hz, while its amplitude varies stochastically between -1.5 V and $+1.5$ V.

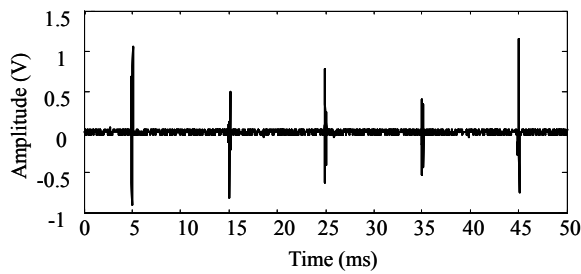


Fig. 11 Periodic Impulsive Noise

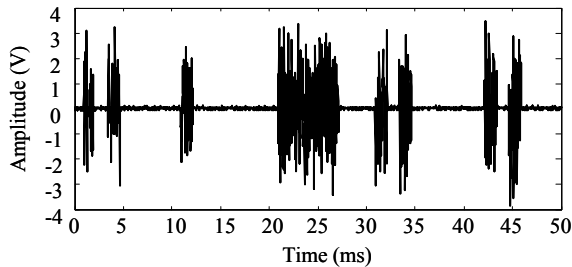


Fig. 12 Aperiodic Impulsive Noise

Fig. 12 shows a snapshot of aperiodic impulsive noises with stochastic waveforms. They appear at random time instants, their pulse widths and amplitudes are random. The pulse width can be configured freely with values ranging from tens of microseconds to as long as several milliseconds. Table II summarizes the configurable ranges of values for the different kinds of noises.

TABLE II
PARAMETER OVERVIEW

Emulated Feature	Parameter	Value
Channel Impulse Response (Amplitude Response)	Length (Frequency resolution)	30 us (33.3 kHz)
	Maximum number	3
Periodical Impulsive Noise Synchronous to Mains Frequency	Cycle time	20ms or 10 ms
	Phase shift	0~1ms, resolution 50us
	Pulse Width	0~200us, resolution 50us
Periodical Impulsive Noise asynchronous to Mains Frequency	Maximum number	3
	Cycle time	2~500 us, resolution 0.5 us
	Pulse width	0~16 us, resolution 0.5us
Narrowband Noise	Maximum number	10
	Middle frequency	0~500 kHz, resolution 0.5 kHz
	Bandwidth	0~20 kHz, resolution 1 kHz
SNR	Range	0~75 dB
	Resolution	0.37 dB
Access Impedance	Number of Types	7

VI. CONCLUSION

For applications like smart metering services, reliability and availability of communications solutions are key requirements. PLC would be most interesting as a physical layer solution for such services, but it is difficult to prove the reliability of

PLC-based communication systems. As a solution to this problem, we have presented a channel emulator for low-speed PLC modems in this paper. It covers the frequency range of the CENELEC A band that is currently envisaged in the EU for intelligent metering services, but can also be reliably used for systems operating in frequency ranges up to 500 kHz.

Validation by measurements verifies that the channel emulator emulates all relevant PLC channel parameters with very high accuracy, including very low access impedances.

REFERENCES

- [1] CENELEC EN 50065-1, Signaling on low-voltage electrical installations in the frequency range 3 kHz to 148.5 kHz, 1991
- [2] M. Goetz, K. Dostert, A Universal High Speed Powerline Channel Emulation System, *International Zurich Seminar on Broadband Communications*, 2002
- [3] M. Zimmermann, K. Dostert, A Multipath Model for the Powerline Channel, *IEEE Transactions on Communications*, Vol. 50, NO.4, 2002
- [4] M. Zimmermann, K. Dostert, Analysis and Modeling of Impulsive Noise in Broad-Band Powerline Communications, *IEEE Transactions on Electromagnetic Compatibility*, Vol. 44, NO. 1, February 2002
- [5] J. Bausch, T. Kistner, M. Babic, K. Dostert, Characteristics of Indoor Power Line Channels in the Frequency Range 50-500 klz, *IEEE International Symposium on Power Line Communications and its Applications*, 2006
- [6] K. Dostert, M. Zimmermann, T. Waldeck, M. Arzberger, Fundamental Properties of the Low Voltage Power Distribution Grid Used as a Data Channel, *IEEE International Symposium on Power Line Communications and its Applications*, 1997
- [7] T. Kistner, M. Bauer, A. Hetzer, and K. Dostert, Analysis of Zero Crossing Synchronization for OFDM-Based AMR Systems, *IEEE International Symposium on Power Line Communications and its Applications*, 2008.
- [8] M. Goetz, Mikroelektronische, echtzeitfähige Emulation von Powerline-Kommunikationskanälen, 2004.