IMTC 2004 - Instrumentation and Measurement
Technology Conference
Como, Italy, May 18–20, 2004

An Instrument for Real-Time Emulation of DSL Binders

Nikolaos Papandreou[1] and Theodore Antonakopoulos[2]

[1]Academic Research Computer Technology Institute - CTI,
Riga Feraiou 61 Str., 26100 Patras, Greece
Phone: +30 2610 997345, Fax: +30 2610 997342, E–mail: npapandr@cti.gr

[2]Department of Electrical Engineering and Computers Technology
University of Patras, 26500 Rio-Patras, Greece
Phone: +30 2610 997345, Fax: +30 2610 997342, E–mail: antonako@ee.upatras.gr

Abstract – *This paper presents an instrument for real-time emulation of digital subscriber loop binders. The instrument is based on the accurate description of the transmission channels characteristics and the binder's crosstalk environment. A key feature of the presented instrument is its ability to support single and multiple lines. In the multiple lines configuration, the pair-to-pair crosstalk injection is based on the actual data transmitted over the disturber lines and is achieved using analytical emulation of the pair-to-pair crosstalk coupling effects.*

Keywords – *Digital Subscriber Lines, Line Emulator, Noise Generators.*

I. INTRODUCTION

The digital subscriber line (DSL) [1] technology is the industry's solution to the last mile problem. DSL modems provide high-speed data communications services over the twisted-pair cables. In contrast to the voice-band modems, the transceivers of the DSL modems utilize higher frequencies for data exchange by applying advanced coding and signal processing techniques, which compensate for the many transmission impairments common to the telephone lines. Due to the increasing demand for higher data rates and the need for new broadband services, DSL technology has attracted a lot of attention over the last years. Several DSL transmission techniques have resulted from the evolution of this technology, e.g. HDSL, ADSL, SHDSL, VDSL, usually referred under the term xDSL [1], [2].

Verification and testing of xDSL systems involves transmission performance measurements over a predefined set of conditions that represent worst-case scenarios in the loop plant. These scenarios include specific sets of subscriber loops, comprising segments of various types with and without bridged taps, and noise sources that include white thermal noise, crosstalk noise, radio noise and impulsive noise. References [3], [4], [5] and [6] enumerate most of the standardized test loops and noises for evaluating the performance of xDSL systems.

Measuring the performance of xDSL devices during development, is performed using laboratory instruments that consist of line cables and noise injection circuits that simulate the aforementioned interfering sources. Figure 1 describes a general diagram of such an experimental set-up [1]. Using this set-up, we can measure the effects of loop attenuation and noise interference on the performance of xDSL transceivers in terms of bit error rate (BER). In order to avoid long coil cables that complicate the construction of multi-segment loops of different cable types, a real-time line emulator can be used. The line emulator has to provide accurate representation of loop attenuation and of multiple types of interference over a variety of loop lengths and noise sources, usually via a programmable environment.

Existing DSL line emulators are either analog or digital [7]. Analog emulators are based on passive networks and active analog circuitry, which provide reasonable accuracy, but high component demand and limited flexibility. Digital emulators usually rely on digital signal processing (DSP) devices and offer high accuracy and extensibility, i.e. emulation of lines according to current and future standards using the same hardware platform.

In this paper we present a new digital instrument for real-time emulation of DSL binders that can be used to test the performance of xDSL transmission systems over the noise environment of a binder of subscriber lines from a central office (CO) to the customer premises equipment (CPE) side. The presented instrument is based on reconfigurable logic and DSP devices and emulates the loop attenuation and the crosstalk interference environment [8] of a DSL binder. The loop topologies under test, along with various types of noise sources are configured using a host based application environment and an in-

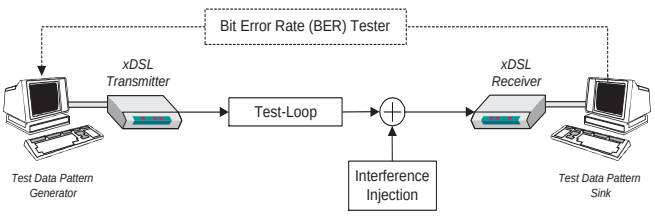

Fig. 1. General measurement setup for xDSL systems.

ternal database. A key feature of the instrument's architecture is its ability to support multiple DSL input lines. In this case, the emulator introduces the crosstalk noise between the emulated lines based on the real data exchanged over these lines using analytical pair-to-pair coupling functions.

Section II provides a concise description of the attenuation and crosstalk environment of the DSL binders. Section III discusses the emulator's functionality regarding the instrument's configurations for single-line or multi-line emulation as well as the method used for generating the signal attenuation and distortion. Finally, Section IV describes the instrument's architecture and gives various implementation details.

II. DSL LOOPS AND NOISE

The twisted pair telephone lines are modeled for frequencies up to 30 MHz by using the well-known two-ports "*ABCD*" theory [9]. The ABCD values are related to the line's primary parameters of resistance *R* (Ω/km), inductance *L* (μH/km), capacitance *C* (nF/km) and conductance *G* (Mho/km), which are frequency dependent and are best determined by measurements over the frequency band of interest. Reference [6] provides analytical tables with measurement data for typical cables used in DSL systems. From these primary parameters, we can calculate the line's secondary parameters of propagation constant γ and characteristic impedance Z_0.

The line's complex frequency response is determined as the output to input voltage ratio. For qualification and interoperability testing purposes of xDSL equipment, both ANSI and ETSI have specified sets of test loops comprising several segments with or without bridged taps and often of different cable types. In this case, the subscriber loop is modeled as cascade of two-port networks and the line's "*ABCD*" matrix is defined as the product of the individual matrices corresponding to each loop segment. References [3], [4] and [5] specify typical test loops.

The main limiting factor on the maximum achievable throughput of xDSL systems is the crosstalk coupling of signal energy from transmission systems on other pairs in the same binder. This type of noise is the result of the capacitive and inductive coupling between the twisted pairs and occurs as near-end-crosstalk (NEXT) and far-end-crosstalk (FEXT) [8]. NEXT is generated between transmitter and receiver pairs at the same end of a cable section, while FEXT is generated between transmitter and receiver pairs at the opposite end of the cable section. Statistical models are adopted for both NEXT and FEXT crosstalk, that expect 1% worst case power sum noise of the cables tested [6]. In general, the NEXT increases with a power of 1.5 of frequency, while the FEXT increases with the square of frequency, but also depends on the squared magnitude of the line transfer function over the length of the crosstalk coupling path. The Full Access Service Networks (FSAN) [6] method defines the models for calculating the noise

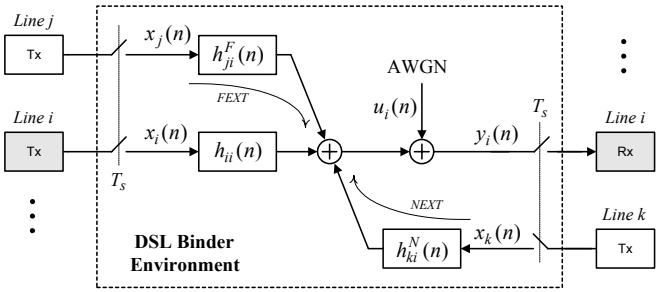

Fig. 2. The DSL downstream noise environment.

of NEXT and FEXT contributions from groups of unlike disturbers in the complete system.

Based on the above loop and crosstalk models, we can calculate the transfer function of standard or custom test loops and the total noise that a DSL receiver experiences under standard or custom noise scenarios.

III. EMULATOR'S FUNCTIONALITY

Figure 2 describes the signal attenuation and crosstalk noise environment experienced in a DSL binder environment. In the following description, we consider the downstream direction of transmission, from the CO to CPE, however a similar discussion also holds for the upstream direction, from the CPE to CO, as well. Considering subscriber line-*i* to be the line of interest, Figure 2 presents the FEXT interference generated from the far-end transmitter of line-*j* and the NEXT interference generated from the near-end transmitter of line-*k* in the binder. It has to be mentioned that the sampling period T_s is referred to the input and output of the binder's emulator and not to the attached DSL modems. The following equation provides the discrete representation of the relation between the input and output signals of line-*i*:

$$y_i(n) = \sum_{m=0}^{L_{ii}-1} h_{ii}(m)x_i(n-m) + \sum_{m=0}^{L_{ji}-1} h^F_{ji}(m)x_j(n-m) + \sum_{m=0}^{L_{ki}-1} h^N_{ki}(m)x_k(n-m) + u_i(n), \quad (1)$$

where x_i and y_i are the input and output signals of line-*i* respectively, h_{ii} represents the impulse response of line-*i* with size L_{ii}, x_j and x_k are the transmit signals of lines-*j* and -*k* respectively, h^F_{ji} represents the FEXT coupling of line-*j* to line-*i* with size L_{ji} and h^N_{ki} represents the NEXT coupling of line-*k* to line-*i* with size L_{ki}, whereas u_i is an additive white Gaussian (AWGN) noise process with variance $\sigma^2_{u_i}$.

The above equation holds at instances $t = nT_s$ ($\forall n = 0, 1, \ldots$), where T_s is the sampling period of the observation, which is in general different from the sampling frequency of each transceiver pair in the binder, but is the same for all emulated lines. When more FEXT and/or NEXT disturbing lines

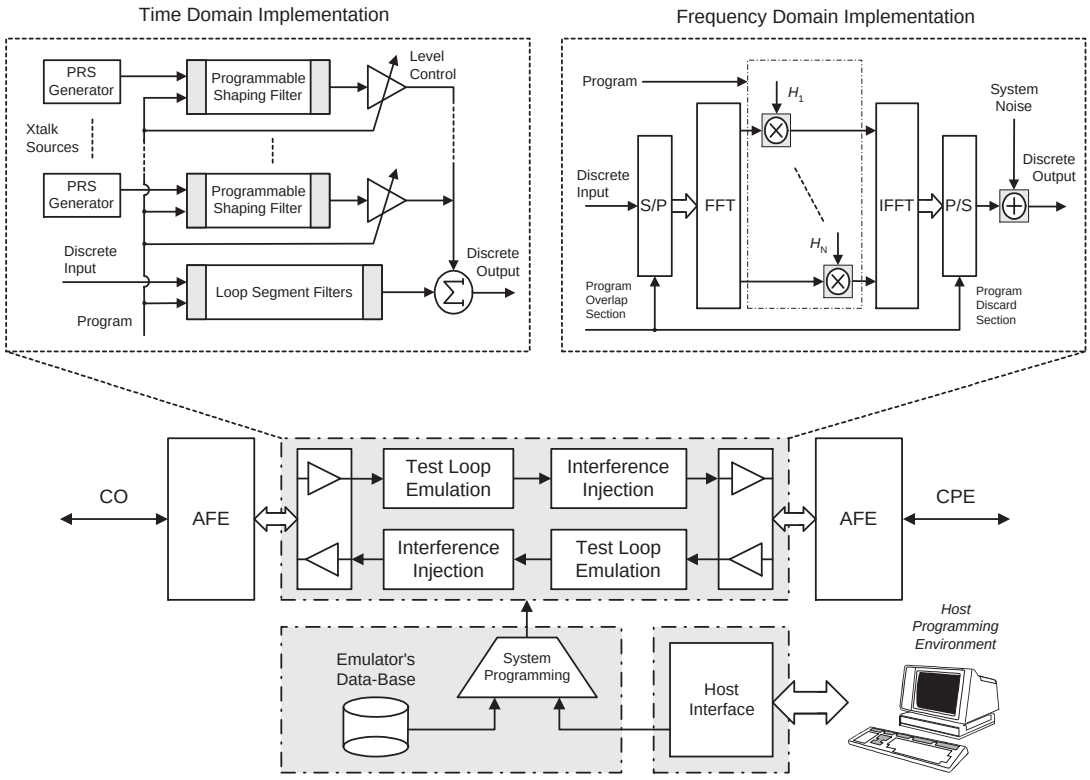

Fig. 3. The architecture of a single DSL line emulation instrument.

exist, the right side of (1) has to be modified respectively in order to account for the total crosstalk interference. Assuming M_F total FEXT disturbers and M_N total NEXT disturbers, equation (1) becomes:

$$y_i(n) = h_{ii}(n) * x_i(n) + \sum_{j=1}^{M_F} h_{ji}^F(n) * x_j(n) + \sum_{k=1}^{M_N} h_{ki}^F(n) * x_k(n) + u_i(n), \quad (2)$$

where $*$ is used for the linear convolution.

The real-time emulator of the DSL binder environment has to implement equation (2) for each of the instrument's input-output lines. This equation can be implemented either in the time domain or in the frequency domain. Figure 3 describes the basic architectural components of the real-time DSL binder emulator, for the two directions of signal transmission with non-overlapping frequency bands. Emulator's configuration in terms of loop types and noise parameters is provided by a host-based application that also uploads emulation results for further processing and graphical presentation. Moreover, the system maintains an internal database where information about standard test-loops and cable types is registered (e.g. primary and secondary parameters for various copper cables, spectral parameters of the xDSL technologies).

A. The Time Domain Approach

The emulator's implementation in the time domain is straightforward and is realized by passing each input through a parallel filters bank and summing all the filter outputs along with an AWGN or other type of noise components. Each filter output is the linear convolution of the input signal with a finite impulse response (FIR) filter that represents either the direct channel or the crosstalk coupling function.

In the single line emulation, noise injection is performed using pseudorandom (PRS) sources, appropriate shaping filters that provide the interfering signals' power spectral density according to the disturbing transmission technologies, and a cascade crosstalk coupling filter in order to create the interference noise samples. White background noise is directly added to the line samples from a high-uncorrelated PRS stream (usually in the order of -140 dBm/Hz).

The general architecture described in Figure 3 can also be extended to support multiple lines. In Figure 4 we present the total binder noise emulation environment, considering two separate lines, named *i* and *j*, in the downstream direction. This configuration includes also the particular filters for the FEXT coupling between the line signals. The FEXT filters are configured in order to account for the coupling path length and the transfer function of the disturbed line. These parameters determine the far-end crosstalk effects as indicated by the analytical models [6].

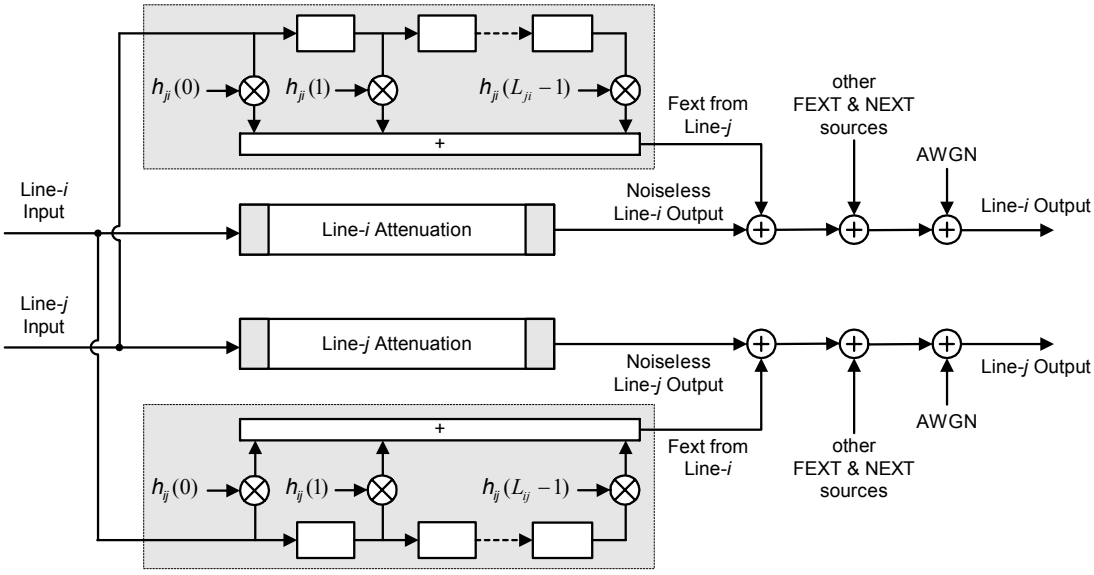

Fig. 4. Multiple lines DSL binder emulation in the time domain.

B. The Frequency Domain Approach

In the frequency domain, the input signal of each line is segmented into blocks of size *N* and then an *N*-point Fast Fourier Transform (FFT) is used to produce the input signal's frequency components. An efficient method to obtain the linear convolution via calculations performed in the frequency domain is the "*overlap-and-save*" [10] method, which divides the corresponding line input into blocks of size *N*, so that each block *overlaps* the preceding block by $L-1$ points, i.e. each block consists of $N-L+1$ new points and $L-1$ points *saved* from the previous block. We assume that $L < N$ (usually $L << N$), where L is the size of the channel's filter or the crosstalk coupling function's filter. Each block is passed through the FFT and the output is multiplied by the complex frequency response of the filter. These results are then passed through the inverse FFT and the N point time domain output is obtained. This output sequence is equivalent to the cyclic convolution of the input block sequence with the filter's impulse response and is equal to the linear convolution only in the last $N-L+1$ points. The first $L-1$ points are discarded as they contain aliasing. Since the linear convolution is a linear time-invariant operation and due to the construction form of the N points input blocks, the last $N-L+1$ points of the successive output blocks are abutted to form the real filtered output.

Figure 5 describes the implementation details for the frequency domain approach considering two separate input downstream lines (*i* and *j*). The presented architecture incorporates the actual FEXT crosstalk coupling functions between the two lines. The input samples are grouped in order to construct the N-point blocks according to the "*overlap-and-save*" technique. Each block is passed through the FFT and is multiplied by the corresponding channel or crosstalk transfer function, whose coefficients are stored in the system. For each line the noiseless output frequency components are added with the FEXT interference frequency components and the complex sum feeds the inverse FFT unit. In order to generate a real output signal, the Hermitian symmetry property [11] is imposed to the input of the inverse FFT. As a result only $N/2$ multiplications and $N/2$ additions need to be performed per direction of transmission in the single line emulator. The remaining complex values are generated using the Hermitian symmetry. As described in the previous subsection, noise injection from sources that do not correspond to actual emulated lines is performed in the time domain using PRS sources and appropriate spectral shaping filters.

IV. THE INSTRUMENT'S ARCHITECTURE

Each direction of transmission, independent if it is upstream or downstream, requires the same set of system resources for implementing the respective functionality. Therefore, based on the total available resources, up to $2M$ directions of transmission can be supported. That results to two possible emulator configurations: either support of M full-duplex lines emulation or $2M$ simplex lines emulation. The existing implementation of the DSL binder emulator supports either the two directions of a single line or the downstream directions of two separate DSL lines. In both cases, loop emulation is performed in the frequency domain, while noise injection is performed in the time domain using programmable filters.

The emulator was prototyped using a custom platform that consists of two FPGAs, 1 million gates each, two high performance floating point DSPs, analog front end circuitry and the appropriate interfacing units.

The analog front-end units (AFEs) are based on 14-bits fast A/D and D/A converters for digitizing the incoming signal and

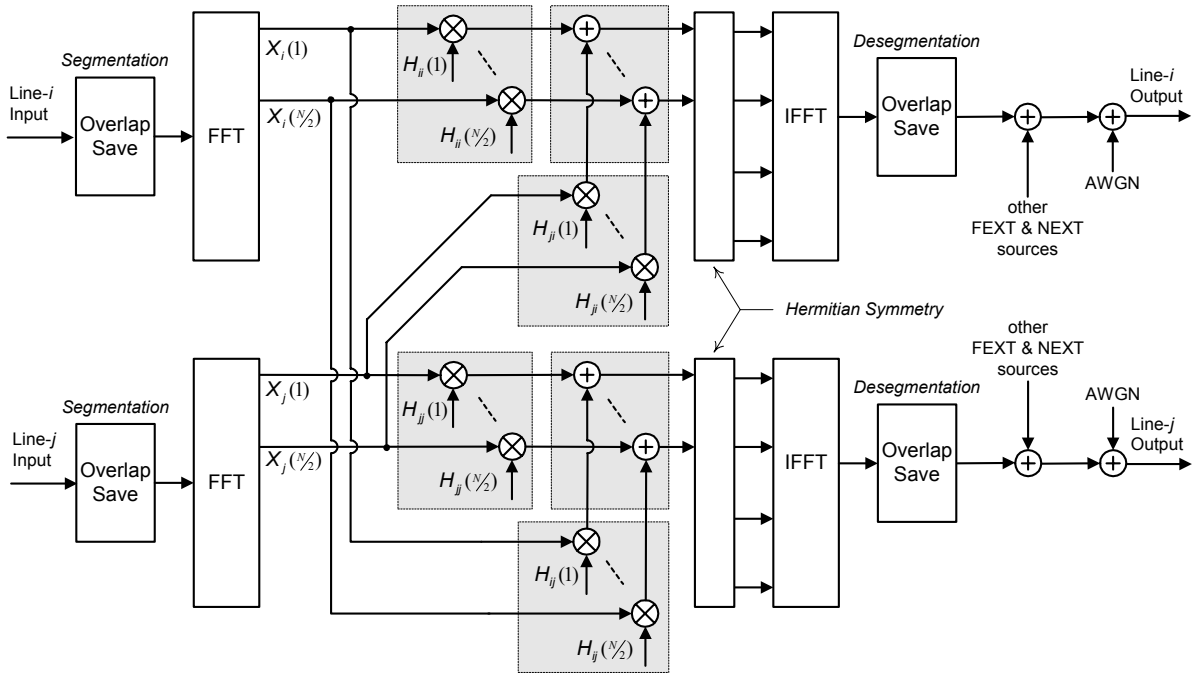

Fig. 5. Multiple lines DSL binder emulation in the frequency domain.

for generating the emulator's output signal. Line interface is performed using commercially available ADSL AFEs. The emulator uses embedded decimation/interpolation circuits for determining a user defined rate on the data stream that is transferred to the emulator's internal logic. At the emulator's output, the reverse operation is performed for generating the analog output signal at the transmitter AFE unit.

The emulator's internal logic follows the configuration shown in Fig. 5. An FPGA is used for implementing the FFT and IFFT modules, used for frequency decomposition of the incoming and recomposition of the outgoing data streams of a single line and the *overlap-save* circuits that are used for counteracting the circular convolution effects imposed by the used FFT/IFFT. Due to the variable size of the *overlap-save* method, a DPLL is used for varying the internal clock and adjusting the timing of the various processing stages.

The incoming digital samples are stored in a local memory and when $N - L + 1$ new samples become available, a new block of N samples is generated and the FFT engine begins its processing. The first $N/2$ frequency domain complex samples are stored in a dual-port RAM (DPRAM) and the DSP is informed about the availability of a new FFT output block. As the DSP starts processing the results of the current data block, the FFT unit proceeds with the next available incoming data block, repeating the whole process sequentially. At this point we have to point out that the emulator uses the same overlapping parameter L in all functions. The value of L is determined as the maximum length of all channel and crosstalk functions of the emulated lines.

The DSP engine is responsible for the complex multiplications of the incoming FFT samples with the line's transfer function, stored in an internal table. The same procedure is used for generating the crosstalk noise that is based on the incoming data of each disturber line. At the end of this procedure, the frequency components that correspond to each subchannel are added and a block of N complex samples that satisfies the Hermitian symmetry is generated. This block is transferred to the DPRAM of the IFFT engine that transforms them into N time domain samples and the $N-L+1$ last samples are used for feeding the noise generation circuit. In order to add additive white Gaussian noise, an DSP-controlled noise generation module is used that creates samples according to a given noise power spectral density and these samples are added to the signal samples at the emulator's output, before feeding the D/A converter.

This procedure is performed on each direction of transmission. The aforementioned hardware platform was designed to comprise a second set of interconnected FPGA and DSP units, so that both directions of a channel or the downstream directions of two separate lines can be emulated.

Figure 6 demonstrates the effectiveness of the presented loop emulator for a single downstream line. We have used the frequency domain approach in order to emulate subscriber loops of 26-AWG cable and of different lengths. Our system consist of a downstream ADSL transmitter sending data over the DSL emulator to the far-end ADSL receiver. The transmitted signal corresponds to the wideband pseudorandom sequence, named *MEDLEY*, which is transmitted during modem

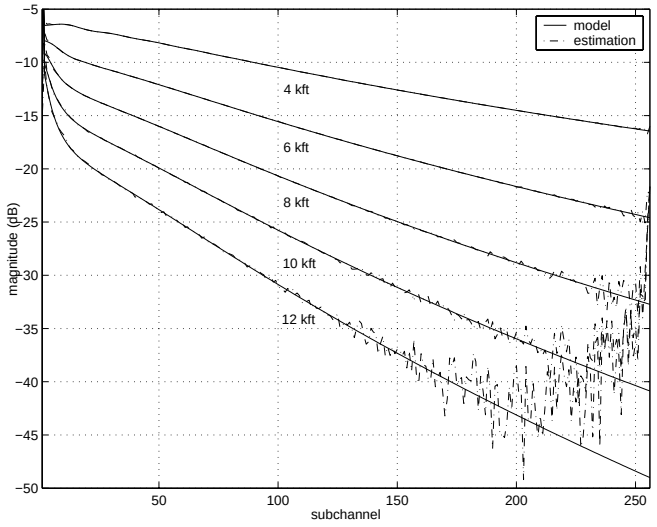

Fig. 6. Emulator's performance example.

initialization [4], and is used by the downstream receiver in order to estimate the channel's transfer function and noise. The transmitted signal is processed by the DSL emulator and the emulated channel output feeds the receiving modem, which performs channel identification using a convergence estimation method [1]. In this example, the DSL emulator uses $L = 54$ and does not inject any noise, for not introducing any errors on the estimation process. Figure 6 presents the analytical and the estimated loop transfer functions for various lengths. Channel estimation was performed using 1000 DMT symbols. The plots of the analytical transfer functions correspond to the standard models provided for DSL system analysis [6]. The estimated transfer function is generated at the downstream receiver and depends on the estimation method's accuracy as well as the DSL emulator's accuracy. For short loops, the accuracy between the analytical and the estimated curves is extremely high and the 1000 training symbols are sufficient. This result proves the accuracy of the DSL emulator. The longer is the loop, the greater is the attenuation and as a result, more training symbols need to be processed in order to receive a good transfer function estimate.

V. CONCLUSIONS

This paper presented a detailed description of an instrument for real-time emulation of DSL binders. This instrument can be used for testing and verifying the performance of transmission techniques used in DSL systems. The instrument is configurable in terms of DSL binder parameters (e.g. test-loop topology and noise sources) and can support crosstalk noise generation based on the actual data transmitted over a disturber line. This feature enables the performance measurement of new transmission techniques, such as techniques that rely on the coordination of CO DSL devices [12], [13]. This option is not possible with classical line emulators based only on random noise generators.

VI. ACKNOWLEDGMENTS

This work was partially supported by the "Karatheodoris" R&D program of the University of Patras and Project 00BE33 entitled "Digital Subscriber Lines Technology" of the Greek Ministry of Industry.

REFERENCES

[1] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding Digital Subscriber Line Technology*, Upper Saddle River, NJ: Prentice Hall, 1999.
[2] T. Starr, M. Sorbara, J. M. Cioffi, and P. J. Silverman, *DSL Advances*, Upper Saddle River, NJ: Prentice Hall, 2003.
[3] High-Bit-Rate Digital Subscriber Lines (HDSL), ANSI Technical Report TR28, 1994.
[4] ANSI, *Asymmetric Digital Subscriber Line (ADSL) Metallic Interface*, ANSI Standard, T.413-1995.
[5] J. M. Cioffi, "Very-high-speed Digital Subscriber Lines System Requirements," ANSI Contribution T1E1.4/96-153, Nov. 1996.
[6] ANSI, *Spectrum Management For Loop Transmission Systems*, ANSI Standard, T1.417-2001.
[7] T. Fliess and H. -J. Jentschel, "Digital real time line simulator," Proc. *14th Int'l Conference on Digital Signal Processing - DSP 2002*, July 2002, vol.2, pp. 1039-1042.
[8] J. W. Cook et al., "The noise and crosstalk environment for ADSL and VDSL systems," *IEEE Commun. Mag.*, vol. 37, pp. 73-78, May 1999.
[9] J. J. Werner, "The HDSL Environment," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 785-800, Aug. 1991.
[10] Al. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Upper Saddle River, NJ: Prentice Hall, 1999.
[11] J. M. Cioffi, "A Multicarrier Primer," ANSI Contribution, T1E1.4/91-159, Nov. 1991.
[12] Kee Bong Song et al., "Dynamic spectrum management for next-generation DSL systems," *IEEE Commun. Mag.*, vol. 40, pp. 101-109, Oct. 2002.
[13] G. Ginis and J. M. Cioffi, "Vectored transmission for digital subscriber line systems," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1085-1104, June 2002.