Adaptive Transmit-Side Equalization for Serial Electrical Interconnects at 100 Gb/s Using Duobinary

Michiel Verplaetse, Timothy De Keulenaer, Arno Vyncke, Ramses Pierco, Renato Vaernewyck, Joris Van Kerrebrouck, *Student Member, IEEE*, Johan Bauwelinck, and Guy Torfs, *Member, IEEE*

***Abstract*— The ever-increasing demand for more efficient data communication calls for new, advanced techniques for high speed serial communication. Although newly developed systems are setting records, off-line determination of the optimal equalizer settings is often needed. Well-known adaptive algorithms are mainly applied for receive-side equalization. However, transmit-side equalization is desirable for its reduced linearity requirements. In this paper, an adaptive sign–sign least mean square equalizer algorithm is developed applicable for an analog transmit-side feed-forward equalizer (FFE) capable of transforming non-return-to-zero modulation to duobinary (DB) modulation at the output of the channel. In addition to the derivation of the update strategy, extra algorithms are developed to cope with the difficult transmit–receive synchronization. Using an analog six tap bit-spaced equalizer, the algorithm is capable of optimizing DB communication of 100 Gb/s over 1.5-m Twin-Ax cable. Both simulations and experimental results are presented to prove the capabilities of the algorithm demonstrating automated determination of FFE parameters, such that error-free communication is obtained (BER $< 10^{-13}$ using PRBS9).**

***Index Terms*— Duobinary, feed-forward equalizer, analog equalization, adaptive equalization, least mean squares.**

I. Introduction

THE current demand for high data rates in network applications pushes high speed serial electrical interconnects to the limits. Traditional non-return-to-zero (NRZ) on-off-keying, although being easy to implement and power efficient, is no longer feasible due to bandwidth limitations of the copper channels. To alleviate this, more bandwidth efficient modulation schemes are explored, such as duobinary (DB) and 4-level pulse-amplitude modulation (PAM4) [1]. However, equalization is still required to compensate for frequency dependent loss and to optimize the channel performance. Duobinary has shown to be a very efficient modulation format over high loss channels because part of the channel loss can be included in the creation of the channel modulation format [2]. Using DB, transmission up to 100 Gb/s over copper has already been achieved over high loss channels [1]. To achieve this high data rate, a transmit-side equalizer is introduced. The transmit-side equalizer settings of [1] are, however, determined offline.

To obtain a practical implementation at these high data rates, adaptive tuning of the transmit-side equalizer filter is required. Only limited work has been performed on adaptive equalizer algorithms for DB used in high speed communication systems. In [3], 12 Gb/s signaling is achieved with an adaptive Sign-Sign-Least Mean Squares (SSLMS) algorithm using a $\times 2$ oversampled receive equalizer. However, when going to higher transmission rates, oversampling is no longer feasible. In [4], 20 Gb/s DB communication is achieved with a 2 tap transmit side equalizer. The adaptation is based on power measurements of reflected signals at the receiver to update the second tap parameter. This is, to our knowledge, the fastest reported adaptive DB equalization so far. However, the length of this FFE will not be sufficient for realistic channels when transmitting 100 Gb/s and the adaptation method is not scalable to longer finite impulse response (FIR) configurations.

Most implemented equalizer update strategies use the data-oriented Least Mean Squares (LMS) based strategies for different equalizer architectures. A brief overview of receive-side adaptive equalization methods can be found in [5]. Other examples of existing transmit-side LMS architectures are discussed in [6] and [7], based on an approximated system identification, and in [8] using an indirect learning architecture. The aforementioned data-oriented based strategies, minimize the error that is defined as the difference between the received and reference waveforms. This can be easily translated into update equations for FFE filters and clearly reflects the quality of the communication channel.

Other quality indicators can be used as well as error signal. For example, Eye-Opening Monitors (EOM) can be constructed to observe signal quality. They are used in [9] for a CTLE equalizer, in [10] for an analog FFE equalizer and in [11] for the adaptation of a CTLE and DFE. In [12], a solution is proposed with asynchronous undersampling histograms to guide an adaptive continuous time linear equalizer (CTLE).

Manuscript received November 8, 2016; revised January 25, 2017; accepted February 16, 2017. This work was supported in part by the Research Foundation-Flanders and in part by the UGent Industrial Research Fund through BiFast and DestIO Projects. This paper was recommended by Associate Editor G. Jovanovic Dolecek.

M. Verplaetse, J. Van Kerrebrouck, J. Bauwelinck, and G. Torfs are with the Internet Technology and Data Science Laboratory, Department of Information Technology, Ghent University–imec, Zwijnaarde 9052, Belgium (e-mail: michiel.verplaetse@ugent.be).

T. De Keulenaer, A. Vyncke, R. Pierco, and R. Vaernewyck are with the Internet Technology and Data Science Laboratory (IDLab), Department of Information Technology, Ghent University–imec, Zwijnaarde 9052, Belgium, and also with BiFast, IDLab, Department of Information Technology, Ghent University–imec, Zwijnaarde 9052, Belgium (e-mail: timothy@bifast.io).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2017.2672788

Although error values obtained with EOMs or histograms can provide useful information, the calculation with the available hardware and translation towards LMS update equations is non-trivial.

The introduction of a transmit-side equalizer forces to abandon the direct implementation of well-known data-oriented receive-side equalization methods. From the reported equalization methods, only the method in [7] promises to achieve tuning in a reasonably short time interval by using a gradient-based method and has to deal with fewer architectural constraints compared to the EOM or histogram based methods. The extra constraints added by the used architecture will, however, hinder straightforward implementation. Added to these constraints, the usage of DB communication will add extra challenges.

This paper presents an adaptation scheme for a transmit-side FFE for DB communication. This scheme is used to equalize high loss channels, which in turn will support 100 Gb/s DB signaling. Added to the equalization problem, a robust transmitter (TX) - receiver (RX) synchronization method for low quality links is developed. This method is required as data-oriented equalizer algorithms rely on correct synchronization between transmitter and receiver. Section II will discuss the system architecture including an introduction towards DB signaling and the topology of the FFE. In section III, the adaptive algorithm for equalization of the high speed signals is proposed. Section IV discusses the stability and validation of the algorithm through simulation, followed by section V, where the experimental results are presented.

II. System Architecture

Development of an adaptive equalizer algorithm will be influenced by the architectural design of the transceiver. First, the DB communication scheme is discussed. Afterwards, the necessary details of the DB chipset are highlighted to introduce the constraints it implies.

A. Duobinary Signaling

The considered architecture uses DB signaling which provides bandwidth reduction (compared to NRZ) and simplicity in decoding (compared to PAM4) [13]. Duobinary was first introduced in [14] and is a form of partial response signaling [15], [16]. A DB symbol is obtained by summing two subsequent NRZ symbols (represented as $1 + z^{-1}$ in the z-domain) [15]. This operation represents a low pass filter, halving the bandwidth of the NRZ signal. Hence, when the overall channel response has this filter characteristic, an NRZ signal at the input will be transformed into a 3-level DB signal. Furthermore, part of the actual channel loss can be used to form this filter and only a part needs to be compensated by the equalizer [13]. A typical DB eye-diagram is shown in Fig. 1 together with the three signal levels and the two decision thresholds in the middle of both eye openings.

To avoid error propagation during the decoding of a duobinary stream, the input binary stream should be precoded [17]. Sending a precoded binary stream through the $1 + z^{-1}$ filter forms a DB stream which can easily be decoded via the usage

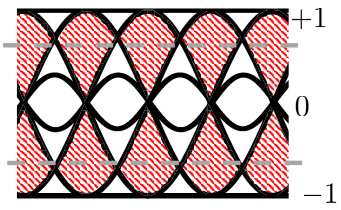


Fig. 1. Duobinary eye-diagram (hatched eye openings) with the two decision thresholds shown.

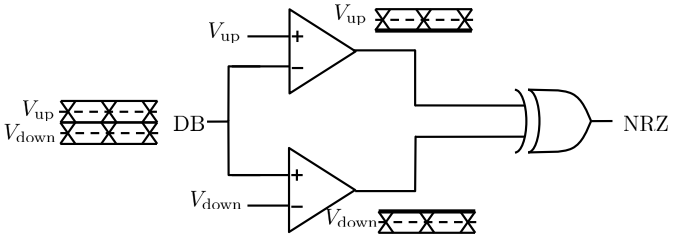


Fig. 2. Decoding stage of a precoded DB sequence to obtain the original transmitted NRZ output stream.

of an XOR operation on the two binary streams resulting from to the upper and lower eyes. Both streams can be extracted using decoding thresholds V_{up} and V_{down}. This decoding system is illustrated in Fig. 2 [18].

B. Transceiver Chipset

An overview of the entire transceiver architecture is found in Fig. 3a [18], [19]. A separate die-photo of the TX IC ($1555\,\mu m \times 4567\,\mu m$) is found in Fig. 3b and of the RX IC ($1926\,\mu m \times 2585\,\mu m$) in Fig. 3c. Both chips are fabricated in a 0.13 μm SiGe BiCMOS technology. The system can process four parallel streams with bitrates up to 25 Gb/s, which are sent through a multiplexer (MUX). The multiplexed signal is equalized and sent over the channel. In the receiver, a DB waveform is demodulated to a binary output stream as explained in section II-A using the decoding levels. Both levels can be set with a precision of 1 mV with a range of 127 mV in either direction. The high-speed binary stream is deserialized to obtain four quarter-rate data streams to complete the SerDes link.

The FIR filter is build-up as a six tap analog equalizer, similar to the equalizer used in [20] and schematically depicted in Fig. 3a. The six tunable gain elements with 8 bits of precision, are interconnected with transmission lines resulting in a fixed tap delay between 9 and 10 ps [19]. With the given delay values, the FFE results approximately in a bit-spaced equalizer at 100 Gb/s. The delays in the FFE are calibrated by a redesign of the chip, taken the design in [20] into account. At a serial rate of 100 Gb/s, the transmitter chip consumes approximately 1 W of which 0.64 W is used for the MUX and 0.36 W is used by the FFE. The complete receiver chip consumes 1.2 W [19].

The choice for a transmit-side equalizer has certain practical advantages compared to receive-side equalization. A first advantage concerns the dynamic range of the RX, which can be made much smaller compared to receive-side equalization. The reason for the reduced dynamic range is twofold.

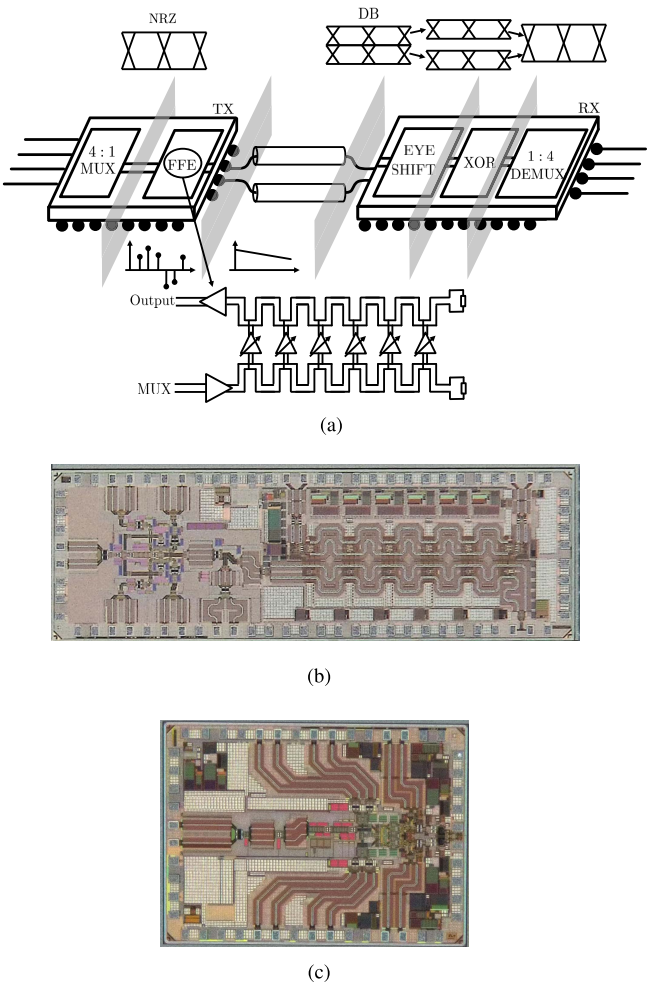


Fig. 3. Architectural overview of the used chipset (a), the TX IC die is shown in (b), the RX IC die is shown in (c).

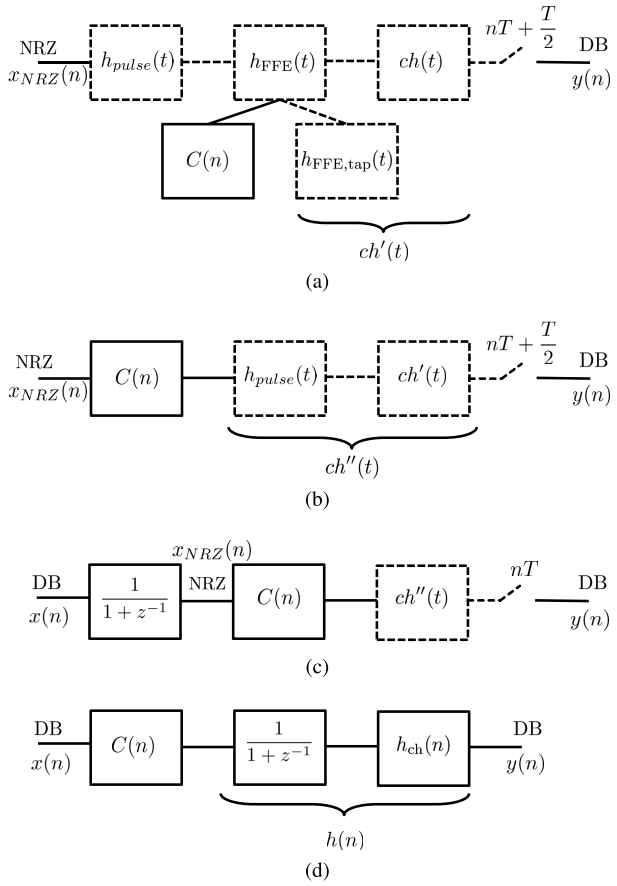


Fig. 4. Equivalent schematic representations of the architecture (analog = dotted), (a) provides the analog representation of the architecture with an analog channel and FFE response, (b) splits the FFE in a digital and an analog part, (c) adds a DB input to the representation and (d) is a reordered equivalent digital representation.

First, the amplitude of the incoming signal is smaller as the low frequency components are attenuated by the transmit equalization. Second, the output noise level can be somewhat higher as no noise enhancement due to filtering is present. With receive-side equalization, the dynamic range should be of the same order as the loss of the highest useful frequency components, which will result in a much higher dynamic range. A second advantage, is the fact that the input stage of the equalizer and the tap amplifiers only process NRZ modulated input data, such that these amplifiers can behave nonlinear. The summation node is the first component in the chain that must be linear. A disadvantage of transmit-side equalization is the need for a linear driver at the transmitter. However, comparing the high speed design effort, the reduced dynamic range at the receiver and the linearity requirements in the FFE are much more stringent than the linear driver, hence favoring transmit-side equalization.

III. Optimization Algorithm for Duobinary

To shape the transmitted signal correctly, an adaptive equalizer algorithm is required to optimize the filter coefficients of the transmit-side equalizer. The update strategy for bit spaced equalizers is derived first to stress the important approximations made. Afterwards, implementation challenges and solutions are discussed.

A. Bit Spaced Equalizer Updates

To derive a coefficient update strategy applicable to the architecture discussed in section II, an equivalent representation of the system is desired. The block diagram in Fig. 4a shows the used analog system, with an NRZ input signal $x_{\mathrm{NRZ}}(n)$. The generation of the analog signal from the digital stream is represented by a pulse shaping filter $h_{\mathrm{pulse}}(t)$. This signal is sent through the equalizer response $h_{\mathrm{FFE}}(t)$ defined by (1) as the cascade of a digital filter $C(n)$ and an analog impulse response $h_{\mathrm{FFE,tap}}(t)$, which represents the analog response of a single FFE tap. The value N_{FFE} corresponds to the length of the FFE.

$$h_{\mathrm{FFE}}(t) = \sum_{j=0}^{N_{\mathrm{FFE}}-1} C(j) h_{\mathrm{FFE,tap}}(t - jT) \tag{1}$$

After the equalizer, the signal passes through the physical channel impulse response $ch(t)$. At the output, the signal is sampled at the instances $nT + \frac{T}{2}$. The $\frac{T}{2}$ term is added to compensate for the NRZ to DB conversion. The block diagram in Fig. 4b splits the analog FFE response $h_{\mathrm{FFE}}(t)$ into its digital and analog part. This analog impulse response

is included in an equivalent channel impulse response $ch'(t)$. Now both $h_{pulse}(t)$ and $ch'(t)$ can be combined in $ch''(t)$. In Fig. 4c, a DB input $x(n)$ is introduced with the addition of an inverse DB filter $\frac{1}{1+z^{-1}}$. Finally, Fig. 4d implements a completely digital equivalent $h_{\mathrm{ch}}(n)$, defined as the symbol spaced samples of $ch''(t)$. Additionally, the blocks are reordered such that a transmit-side FFE architecture is obtained. The inverse DB filter $\frac{1}{1+z^{-1}}$ and $h_{\mathrm{ch}}(n)$ can be combined in one digital channel response $h(n)$.

1) Transmit-Side LMS Architecture: With the help of Fig. 4d, we can write the output $y(n)$ as:

$$y(n) = \sum_{j=0}^{N_{\mathrm{FFE}}-1} \sum_{i=0}^{\infty} C(j)h(i)x(n-i-j) \tag{2}$$

The goal of the update strategy is to minimize the mean squared error (MSE), where the error is defined by (3).

$$e(n) = x(n) - y(n) \tag{3}$$

For this purpose the cost function $S(n)$ in (4) must be minimized where the operator $E[.]$ calculates the expected value.

$$S(n) = \frac{1}{2} E\left[|e(n)|^2\right] \tag{4}$$

Calculating the gradient of $S(n)$ with respect to the components of $C(n)$ yields equation (5):

$$\frac{\partial S(n)}{\partial C(j)} = -E\left[e(n) \sum_{i=0}^{\infty} h(i)x(n-i-j)\right] \tag{5}$$

As expected, this result is similar to the formula for a receive-side FFE equalizer [21].

The notation in (6) can now be used to simplify the expressions,

$$\sum_{i=0}^{\infty} h(i)x(n-i-j) = (h*x)(n-j) = x'(n-j) \tag{6}$$

and an unbiased estimator for $\frac{\partial S(n)}{\partial C(j)}$ can be found in (7) [21].

$$\frac{\partial S(n)}{\partial C(j)} \approx -e(n)x'(n-j) \tag{7}$$

Using (7), an iterative update for the coefficients of the filter $C(n)$ can be proposed. Each update iteration k, the values of the N_{FFE} non-zero coefficients of $C(n)$ are combined in the vector $\overline{C}(k)$, defined in (8).

$$\overline{C}(k)^T = [C(N_{\mathrm{FFE}}-1)\ldots C(0)] \tag{8}$$

After combining the N_{FFE} corresponding samples of $x'(n)$ in $\overline{x'}(k)$, defined by (9), the update equation for the $\overline{C}(k)$ is obtained in (10).

$$\overline{x'}(k)^T = [x'(k-N_{\mathrm{FFE}}+1)\ldots x'(k)] \tag{9}$$

$$\overline{C}(k+1) = \overline{C}(k) + \lambda e(k)\overline{x'}(k) \tag{10}$$

Up to this moment, the derivation for the LMS algorithm is equal to a receive-side LMS update engine. The difference with our setup is the inability to observe the signal $x'(n)$ directly. To remedy this, the Maximum Peak Approximation method is utilized [7]. Using this method, an estimate of $x'(n)$ can be made.

We first adopt a SSLMS update strategy which alters the update equation to (11):

$$\overline{C}(k+1) = \overline{C}(k) + \lambda\ sgn(e(k))sgn(\overline{x'}(k)) \tag{11}$$

Each of the components of the vector $sgn(\overline{x'}(k))$ can be approximated by (12), similar to [7]. The sample corresponding to h_c is the most significant sample of $h(n)$.

$$sgn(h*x)(n-j) = sgn(h_c x(n-c-j)) \quad \text{when } |h_c| \geq \sum_{i=0, i\neq c}^{\infty} |h_i| \tag{12}$$

Under the assumption that the impulse response $h(n)$ has a positive maximum, the term h_{c} can be omitted as only the sign is important. In this way, only known or measurable quantities are left in the update equation of (11). A better approximation can be obtained if more samples of $h(n)$ are used.

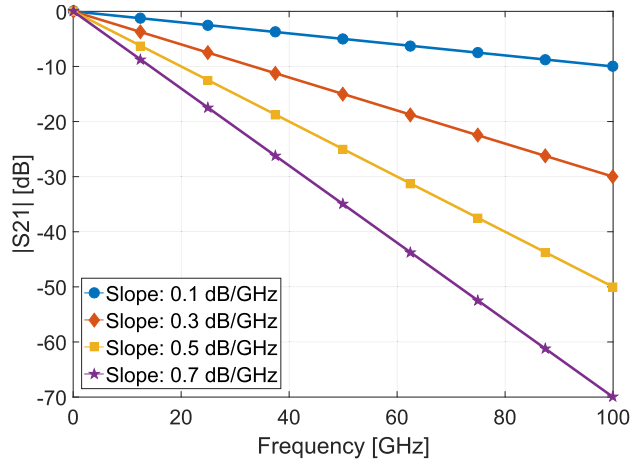
According to Fig. 4, $h(n)$ is the cascade of the channel $h_{\mathrm{ch}}(n)$ with the inverse DB filter ($\frac{1}{1+z^{-1}}$). To observe the direct influence of the channel, (12) can be rewritten in function of the input NRZ stream $x_{\mathrm{NRZ}}(n)$:

$$sgn(h*x)(n-j) \approx sgn\Big(\sum_{i=0}^{1} h_{\mathrm{ch}}(c+i-\frac{1}{2})x_{\mathrm{NRZ}}(n-c-i-j)\Big) \tag{13}$$

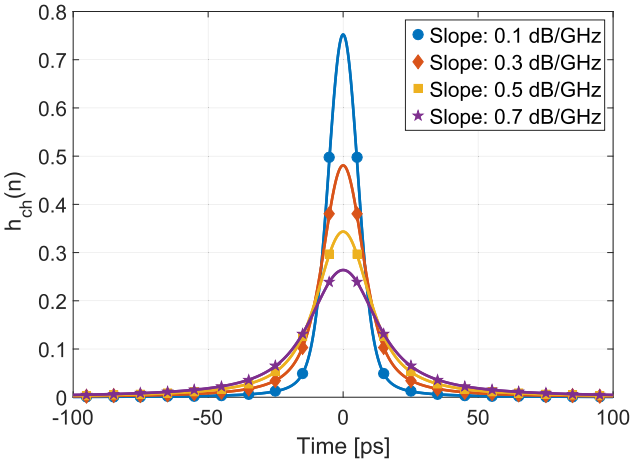
This implies that two samples of the channel impulse response have to be used. In an actual implementation, both samples are not exactly known, and hence a good approximation is needed. In practice, both $h_{\mathrm{ch}}(c-\frac{1}{2})$ and $h_{\mathrm{ch}}(c+\frac{1}{2})$ will be set equal which is a good approximation for many channels. However, in this case, the approximation of (13) will be zero when two subsequent values of x_{NRZ} have opposite signs. This will prohibit an update of the particular coefficient, hence slowing down the system, but on average, the gradient will still point to the optimal location. If more known samples of $h_{\mathrm{ch}}(n)$ are used, the chance of a sign estimation of 0 will lower with increased amount of samples in the approximation.

2) Errors in the Estimation: The condition in (12) is only true for channels with little loss, which translates in a relatively short impulse response. However, not meeting the condition does not imply a malfunctioning of the algorithm as the condition in (12) only considers the worst-case scenario. As long as the amount of sign errors is negligible, the averaging effect of the update strategy will make it converge in the correct direction. To verify the amount of sign errors, the sign error rate (SER) is estimated for different idealized channels with a flat loss profile expressed in dB/GHz, depicted in Fig. 5a.

The frequency responses of Fig. 5a are converted to the time domain (cascaded with $h_{\mathrm{pulse}}(t)$ having a Gaussian impulse response with $\sigma = 4.7$ ps) and sampled at the correct instances in Fig. 5b. With the sampled version of the impulse response, an estimation for $sgn(h*x)(n)$ is calculated with the help of (13) using the center two or four samples (using an extension of (13) for more samples). The estimated sign is compared to the correct sign of $(h*x)(n)$ (sign estimation



(a)



(b)

Fig. 5. The different channels used for the SER evaluation are shown in (a), (b) shows the impulse responses of the channels (cascaded with $h_{\text{pulse}}(t)$ having a Gaussian impulse response with $\sigma = 4.7\ ps$) where the used samples of the impulse response are marked.

of 0 is considered as correct). In Fig. 6, the SER for the outer transitions ($+1$ and -1) and middle transitions (only for the four-sample estimation) are given for different kinds of input bit sequences. Both a random sequence (2^{18} values taken from a discrete uniform distribution) and a 2^7-1 pseudo random binary sequence (PRBS7) sequence are used for the values of $x_{\text{NRZ}}(n)$. A PRBS7 sequence is introduced as training sequence in the practical execution of the algorithm for its simplicity in implementation.

For both input sequences, the simulated SER at the outer transitions are equal for both the two and four-sample estimation. For all simulated channels and input sequences, the SER is below 7% which results in a relatively small error that will be averaged out without major impact on the stability of the algorithm. However, only channels with a loss up to 0.6 dB/GHz will be used in our system as they have an impulse response which is short enough to be compensated with the available FFE. For these channels, a PRBS7 stream will result in maximum 1 sign error per period in the worst case. Simulations using PRBS7 have shown that the impact of this low amount of errors is negligible.

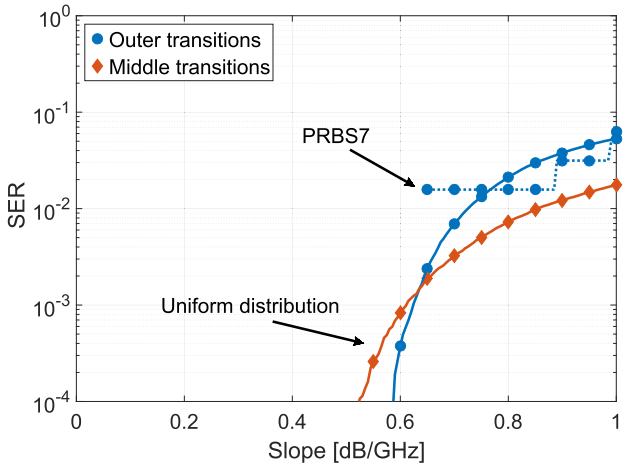


Fig. 6. Validation of the sign error rate (SER) of the used approximation for the different channels of Fig. 5a.

The estimations are performed with the knowledge of the exact values of the impulse response, which are not available in practice. If incorrect values are used in the estimation, the SER will increase very quickly, especially with the four-sample estimation. Using the two-sample estimation, only symmetry between both samples is important. Estimating both samples to have symmetric values will in practice not deviate a lot from the actual response. For the four-sample estimation, it is more difficult to estimate correct values for the impulse response samples, hence, a larger SER can be expected. For this reason, the two-sample approximation will be used in the remainder of this paper.

B. Practical Implementation of the Algorithm

To create a practical implementation of the algorithm, certain constraints defined by the architecture in section II must be taken into account. First, the hardware limitation on the calculation of $e(n)$ is discussed followed by practical methods to synchronize the transceiver chain and the calculation of a synchronized version of $x(n)$ at the receiver. At last, an additional algorithmic update for the threshold levels is introduced.

1) Alternative Error Definition: In the receiver chip, only two decoding thresholds are available to generate the error data for the algorithm while three of them are theoretically required ($+1$, 0 and -1). This introduces the constraint that only one of both DB eyes can be analyzed at the same time. Thanks to symmetry, this is not a problem. Discarding one of both eyes can be implemented by implying (14) for $e(n)$:

$$\begin{cases} e(n) = x(n) - y(n) & x(n) \in \{+1, 0\} \\ e(n) = 0 & x(n) \in \{-1\} \end{cases} \tag{14}$$

The calculation of $e(n)$ with only 2 decoding levels is illustrated in Fig. 7. The value of $x(n)$ will determine which comparator value is used.

2) Quarter-Rate Synchronization: Data-oriented adaptive equalizer algorithms (like our LMS engine) rely on a correct synchronization between the transmitter and the receiver to calculate the error values. For this purpose, a locked reference

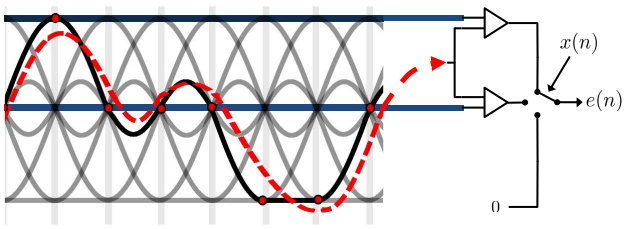


Fig. 7. Illustration of the calculation of $e(n)$ from the incoming data stream. The signal $x(n)$ determines the used comparator value.

stream, which is synchronized to the incoming data is desired. Synchronization can be performed by transmitting a quarter-rate NRZ signal instead of a full rate NRZ (or DB) signal. As the BER will be much lower due to the reduced bandwidth of the signal, synchronization is now much easier. A full rate NRZ signal can be derived from this quarter-rate signal at both the transmitter and receiver such that a synchronization on the full rate is obtained. For this purpose, properties of PRBS signals can be used, hence, an important reason to introduce a PRBS sequence as training data. However, still an ambiguity of four delay positions can exist in the synchronization at the full rate. To resolve this delay uncertainty, a method based on the average error is proposed in the following section.

3) Duobinary and Delay Estimation: The update equation from (11) depends on the calculation of $e(n)$ and $\overline{x}$. To calculate these parameters, the correct DB samples are needed at the receiver. As observations are only possible after the decoding step from DB to binary data, these DB samples are not readily available. Assuming the BER is sufficiently low such that the transmitter and the receiver can be synchronized at the full data rate, the DB stream $x(n)$ must still be calculated. To calculate the samples, an inverse XOR operation is needed. However, this operation is ambiguous, as a sign inversion of the DB stream will not change the binary output stream. For this reason, two versions of the DB stream $x(n)$ are calculated, each with a different polarity. To determine the correct $x(n)$, this sign ambiguity should be resolved by calculating the $e(n)$ values corresponding to estimated values of $x(n) = +1$ for both polarities (called e_{up}). For the $x(n)$ stream which is estimated with the wrong polarity, e_{up} will mainly have the same sign for each value. As each estimated $x(n) = +1$ value will correspond to an actual waveform sample which is negative or close to zero (as it should be estimated as $x(n) = -1$ to be correct), all samples will lay on the same side of the decision threshold. For the correct polarity, most of the actual waveform samples will be positive, and will cross the decision threshold more often, changing the sign of e_{up}. Averaging e_{up} will make it possible to determine correct polarity based on the minimum absolute value.

As mentioned in section III-B.2, the assumption of a correct synchronized NRZ stream will only be correct with an ambiguity of four delay positions. To solve this issue, e_{up} can be calculated for reference streams which have different delays. Only for the correct delayed stream, the e_{up} values will have constantly the same sign when the DB stream has the inverted sign. For the other delayed streams, e_{up} will flip sign more often due to the decorrelation between PRBS streams of different delay.

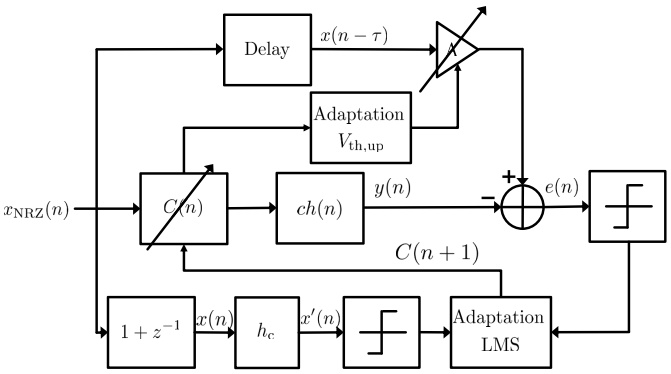


Fig. 8. Blockdiagram of the combined LMS and threshold update loop.

4) Automated Threshold Optimization: To calculate $e(n)$, the incoming signal must be compared with two thresholds to create the sign information. The threshold at the top of the signal is used to compare with $x(n) = +1$. Adaptively updating this threshold is important to ensure the algorithm functions properly over channels with different loss profiles. The update can be guided via the extra control loop of (15) which will adapt the threshold such that the maximum FFE tap value is driven towards a certain value C_{set}.

$$V_{\text{th,up}}(k+1) = V_{\text{th,up}}(k) - \lambda_{th}(\max(\overline{C}(k)) - C_{set}) \quad (15)$$

The value C_{set} should be chosen close to the maximum tap gain to obtain the maximum eye-height. However, choosing C_{set} equal to the maximum gain will stall the entire system if one of the tap gains wants to go higher than the maximum value (as in this case the update of the taps is stopped). A value of C_{set} which is marginally lower than the maximum gain will prohibit this stall. However, for values that are very close to the maximum gain, convergence time of the algorithm will increase significantly. A value of 0.95 times the maximum gain is chosen in our case as a trade-off between convergence time and final eye-height. Both the adaptive system for the threshold in (15) and the coefficients in (11) will execute simultaneously, hence creating a complex system. Both systems can be combined in the block diagram of Fig. 8 (a delay τ is introduced to mimic the delay of both the channel and FFE). Both loops influence each other's performance, possibly jeopardizing stability of the combined system. In section IV, the stability of the system in Fig. 8 will be analyzed.

5) Practical Execution: To correctly perform the optimization, all practical steps highlighted above should be executed. Therefore, during execution of the algorithm, the state machine in Fig. 9 is used.

One should note that it is not necessary to update $\overline{C}(k)$ each iteration. The product $sgn(e(k))sgn(\overline{x'}(k))$ can be averaged, which will lower the variance on the estimated gradient [22]. This average can be defined as:

$$\overline{u}(k) = \frac{1}{D}\sum_{i=1}^{D} sgn(e(kD-i))sgn(\overline{x'}(kD-i)) \quad (16)$$

Increasing D has a similar effect as decreasing λ. Increasing D is preferred over a decrease in λ because of the large update

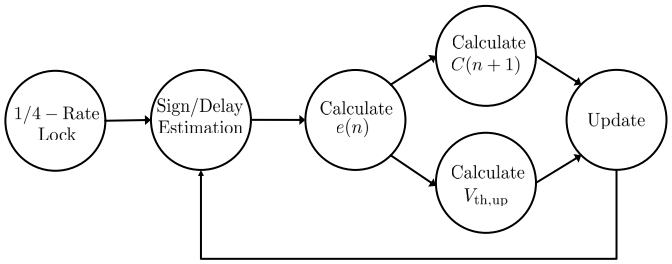


Fig. 9. Overview of the different states during the optimization of the algorithm.

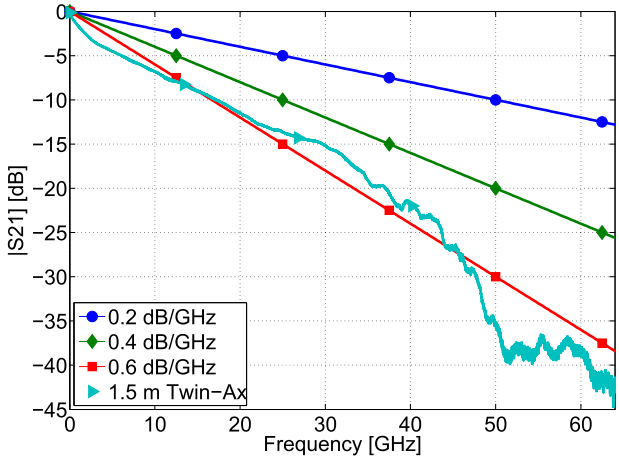


Fig. 10. Different channel responses used in the simulation.

delay (as the update values must be passed from RX to TX). During the update, no useful calculations can be performed at the RX. Increasing D, lowers the amount of updates and hence increases the useful measurement time at the RX. In practice, a value of D = 127 is used, for the ease of implementation, as this value corresponds to a complete PRBS7 period which is the training data that will be used.

IV. Simulation and Convergence Analysis

In section III-A and III-B, two update algorithms have been introduced resulting in the system of Fig. 8. To validate the performance and stability of the designed optimization algorithm, a simulation environment is built based on the subblocks in Fig. 4a. The FFE is modeled with the use of a measured response (S-parameter measurement) for $h_{\text{FFE,tap}}(t)$ and an idealized tap delay of 10 ps. The channel $ch(t)$ can take a variety of forms. The frequency response of the channel models used in simulation are shown in Fig. 10. Three of the channel models use idealized channels with a flat loss profile expressed in dB/GHz. Next to these, a measured channel consisting of 1.5 m Twin-Ax cable, and identical to the channel used in the measurement setup, can be simulated. First, the stability of the system is analyzed such that appropriate values for λ and λ_{th} can be proposed. Afterwards, more detailed simulation results are presented using the channels from Fig. 10.

A. Convergence Analysis

Stability and convergence time of LMS update systems are very important in practical applications. The stability in LMS loops (e.g. [21], [23]) and SSLMS loops (e.g. [24]) is dependent on the value of the step-size compared to the properties of the training data and channel. In our case, the introduction of the threshold update loop gives rise to an extra influence on the stability. As both λ and λ_{th} can be chosen, the stability will be a two-dimensional problem. To investigate this stability problem, simulations are performed for various values of λ and λ_{th}. For each combination of both parameters, simulations are performed over at least two times the amount of iterations needed to reach a stable regime solution, with a minimum of ten thousand iterations. The used training data is a PRBS7 sequence and a channel of 0.6 dB/GHz is used to filter the transmitted data. The point where stable regime is achieved, is defined by the iteration, from which, on average, no updates are performed on $V_{th,up}$ nor on coefficients of $\overline{C}$. To evaluate the quality of the solution in regime, the MSE of each iteration (calculated over $D = 127$ bits) is averaged over the last 1000 iterations of each simulation. In regime, the MSE should stabilize towards a minimal value, called the steady-state excess MSE [21]. For standard SSLMS algorithms (without the simplification of (12)), the excess MSE can be theoretically calculated based on the statistics of the data [21], [25].

In Fig. 11, the results of the simulation in function of both step-sizes are provided by contour plots showing the number of iterations (Fig. 11a) to achieve regime and the final average MSE (Fig. 11b). For high λ and λ_{th} values (much greater than 1), the system becomes unstable, hence, these regions should be avoided at any time. From Fig. 11a, it can be concluded that the speed of the system is mainly influenced by the value of λ when $\lambda_{th} > 10^{-4}$. From Fig. 11a, an optimal region in terms of speed can be distinguished. However, comparing this same region with Fig. 11b, shows that a fast convergence corresponds to a high average MSE. This high value is mainly related to fast and large fluctuating MSE values in regime due to the large step-size. In terms of MSE, the results are almost independent from λ_{th}, as could be expected by the fact that the threshold loop is only used to keep the final filter coefficients bounded. Hence, the high MSE forces us to use lower values for λ.

Similar contour plots can be made for the channels with lower loss gradients which result in an increased minimum value for the MSE and a small decrease in convergence time, which will be illustrated in section IV-B. The unstable regions become slightly smaller if lower loss channels are used.

In practice, it is not needed to keep a constant step-size during the optimization. Gradually lowering the step-size creates the possibility to decrease the algorithm run time significantly while keeping a low final MSE. As the MSE is independent from λ_{th} in Fig. 11b, it makes sense to keep λ_{th} fixed and vary λ.

Several nonidealities are not included in the simulations of Fig. 11, however, the convergence properties can be influenced by them. First, the convergence can be influenced by a non-linear response of the gain of an FFE tap with respect to the digital control settings. In Fig. 12, the measured gain response of one tap is provided. Both offset with respect to 0 and third order nonlinearity are present. The offset will have no

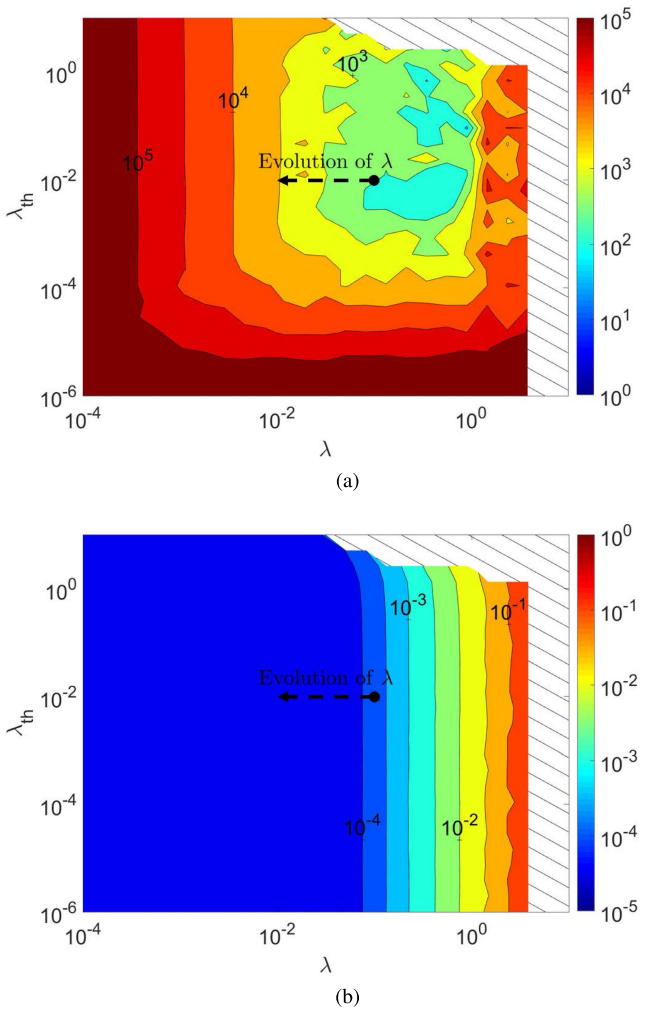


Fig. 11. The amount of iterations needed to achieve a stable solution (a) and the final MSE, averaged over 1000 iterations after achieving a stable solution (b) in function of the step-sizes λ and λ_{th}. Simulations are performed using a channel of 0.6 dB/GHz. Unstable regions are hatched. The used evolution for the values of λ are marked with the arrows.

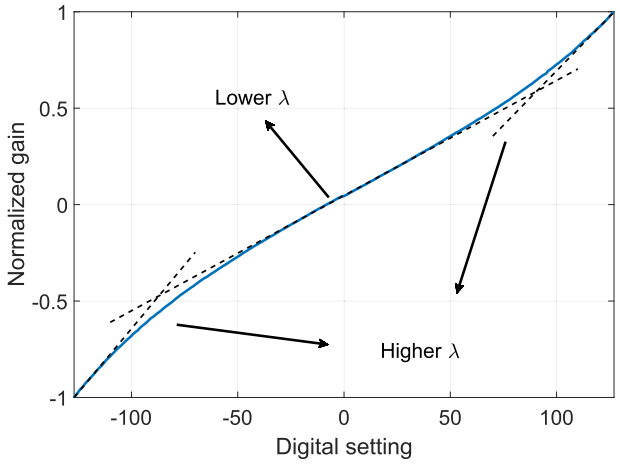


Fig. 12. Normalized gain response with respect to the digital settings of the the tap amplifiers.

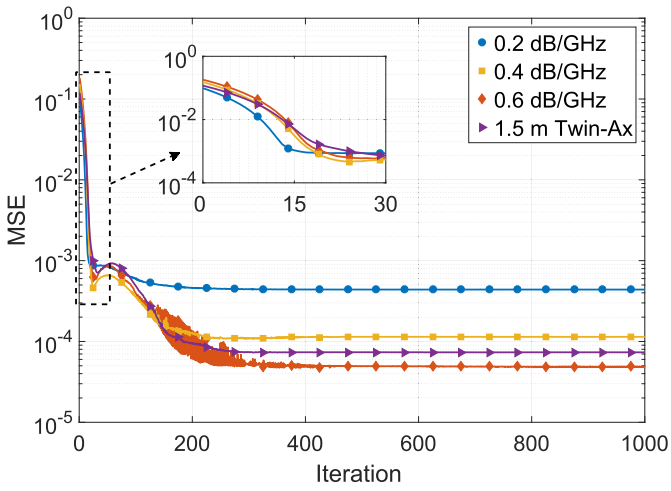


Fig. 13. The MSE evolution for a simulated DB link at 100 Gb/s over various channels with an insert showing the curves during the first iterations.

influence on the convergence. As indicated in Fig. 12, the non-linearity will lead to different effective λ values, dependent on the tap value. In our case, this leads to a reduced λ for the smallest taps which leads to slower convergence but higher stability of the overall system.

The adaptive LMS loop can be influenced by variation in the delays between the taps. As mentioned in section II-B, the delays vary between 9-10 ps. The effect of an LMS loop on an equalizer, where the delays are slightly lower than the symbol spacing, is briefly analyzed in [26]. Our simulations show that in our case, a smaller tap spacing will still result in a successful cancellation of the intersymbol interference (ISI), however, the obtained signal swing will be smaller. For values larger than 10 ps, signal quality starts degrading very quickly which is the same conclusion as in [26].

The influence of finite precision of the tap coefficients is for example discussed in [21]. Apart from the degradation of the minimum reachable MSE, simulations show that for lower coefficient accuracy, an increase of the variation of the MSE in regime can be observed. However, the available accuracy of 7 bit is high enough such that perturbations are still limited and degradation of the minimum MSE is limited.

The effect of nonlinear distortion in the channel (or the transmit driver) will be similar as discussed in [27], decreasing the quality of the final solution (increased MSE) but decreasing convergence time. As the driver in the transmitter is designed to be very linear, the influence of nonlinear distortion will be limited.

The cope with the possible variations due to nonidealities, some margin on the used values for the step-sizes must be taken into account. From Fig. 11, safe starting values (far enough from the unstable region) for λ and λ_{th} can be proposed. For example respectively the values 0.1 and 0.01 can be used as step-sizes. Lowering the value of λ gradually towards 0.01 brings the algorithm in a regime state with the minimum average MSE. These values are graphically shown on the contour plots of Fig. 11.

B. Simulation on Various Channels

Using the values for λ and λ_{th} from section IV-A, simulations over the different channels of Fig. 10 can be performed to

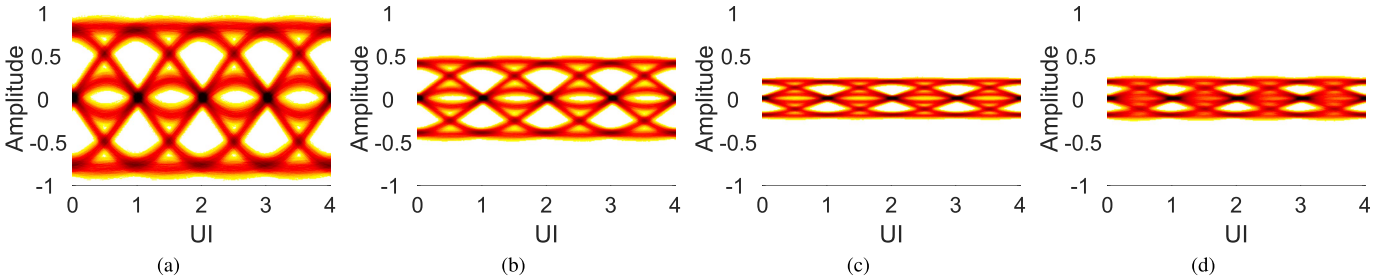


Fig. 14. Final simulated eye-diagrams at 100 Gb/s for different channel loss profiles. (a) 0.2 dB/GHz channel. (b) 0.4 dB/GHz channel. (c) 0.6 dB/GHz channel. (d) 1.5 m Twin-Ax channel.

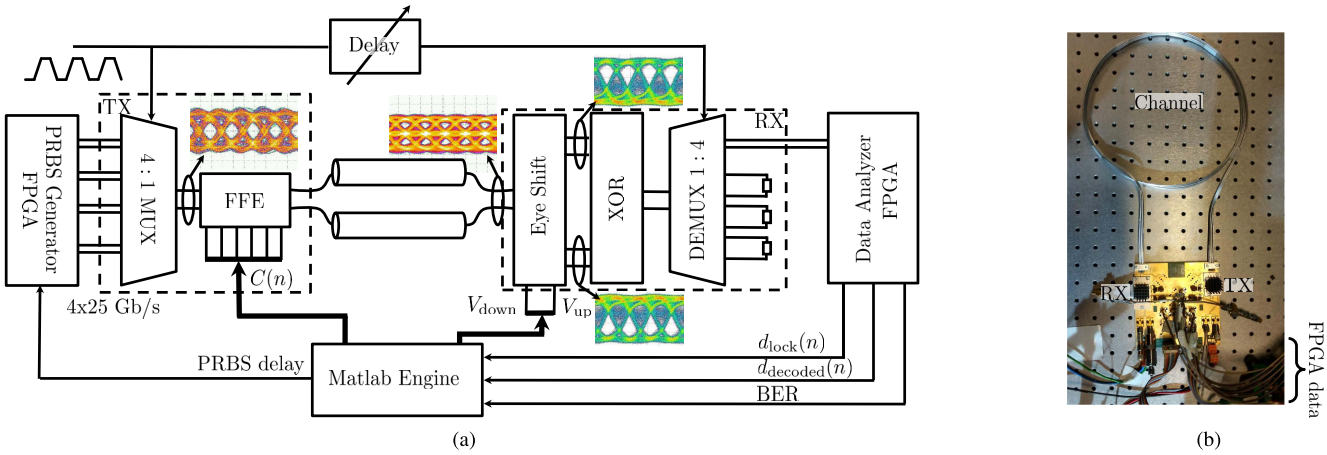


Fig. 15. Schematic overview of the test setup used to evaluate the developed LMS algorithm update (a) with a photo of the used setup (b).

verify operation for different channel losses. The initial value of $V_{th,up}$ is scaled relatively to the channel loss, such that a fair comparison can be made between the different simulations. The value for λ is made variable such that it changes linearly from 0.1 to a value of 0.01 in 500 iterations.

The MSE evolution of the simulations is found in Fig. 13. The MSE stabilizes at a minimum value in all four cases. A lower channel bandwidth, leads to a lower excess MSE. While this seems counter-intuitive, the lower bandwidth will force the eye height to be smaller, creating a lower equivalent value for $x(n)$. This lower value directly translates into lower MSE values. In the insert of Fig. 13, the faster convergence for lower loss channels is visible, especially when making the comparison between the 0.2 and 0.4 dB/GHz channel, supporting the statement in previous section. During the first iterations of the MSE evolution for the 0.6 dB/GHz channel (up to iteration 300), the MSE is very noisy due to the high value of λ that is relatively close to the unstable region. For later iterations, the lower λ removes the noise on the MSE. As other channels do not show this behavior for any iteration, they are assumed to be much more stable using the specific value for λ and λ_{th}. For all channels, the long-term MSE evolution is not monotonic, which deviates from the behavior of normal LMS algorithms. The extra variations are caused by the dynamics of the threshold loop.

In Fig. 14, the eye-diagrams at 100 Gb/s after complete convergence of the algorithm for the different channels are given. In all cases, a clean eye-diagram is obtained. However, the longer, lower bandwidth channels are forced to trade gain for bandwidth. For this reason, the final eye-diagrams are smaller in the higher loss channels, which reflects itself in the lower MSE values in Fig. 13.

Combining the stability results from section IV-A and the simulations over various channels, it can be concluded that the algorithm converges into a stable solution, in a relatively short time, given that the values for λ and λ_{th} are correctly chosen. Observing the different eye-diagrams in Fig. 14, this stable solution provides a correctly optimized DB eye, as intended by the update system.

V. Experimental Verification

To verify the developed LMS update strategy with the actual tranceiver chipset, a test environment was built. In this section, the experimental setup is discussed and followed by actual algorithm runs.

A. Experiment Setup

An overview of the lab setup is shown in Fig. 15. At the transmit side, PRBS7 (or higher order PRBS) streams are generated by an FPGA. The mutual delays between the four different PRBS streams can be separately tuned to obtain either a quarter-rate serial stream or a full rate serial PRBS7 stream. These four streams are multiplexed on chip using a half rate clock creating a serial stream of 100 Gb/s. This signal is sent through the integrated six tap analog FFE for which the coefficients can be set via a serial interface. At the receiver, the signal is amplified and both eyes are extracted as explained in section II. After a XOR and demultiplexing stage (more details in [18]), four quarter-rate streams are obtained. Due to the absence of CDR circuitry in the receiver prototype,

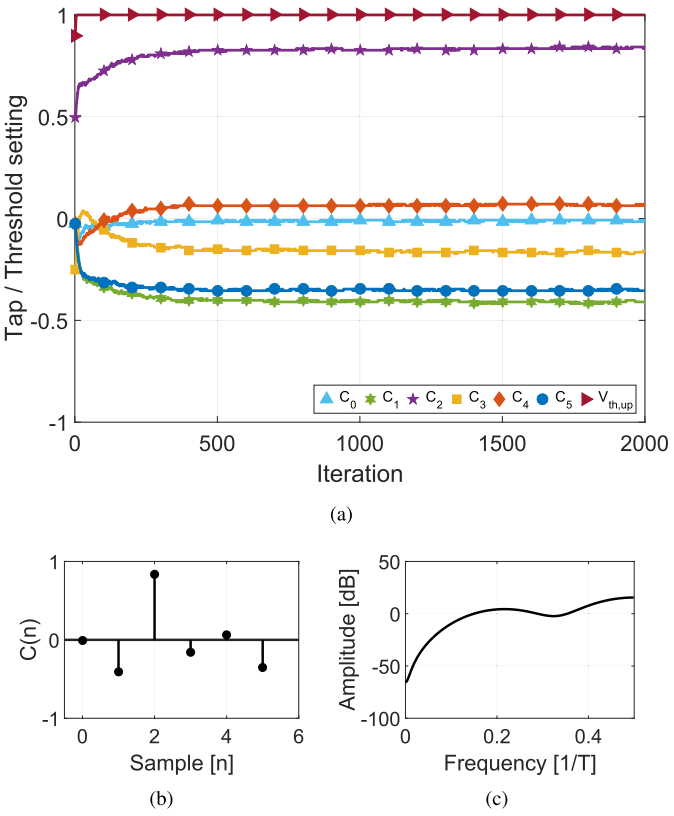


Fig. 16. Parameter evolution for DB communication optimization at 100 Gb/s over 1.5 m Twin-Ax cable (a). The final impulse response is given in (b), the amplitude response of this filter is given in (c).

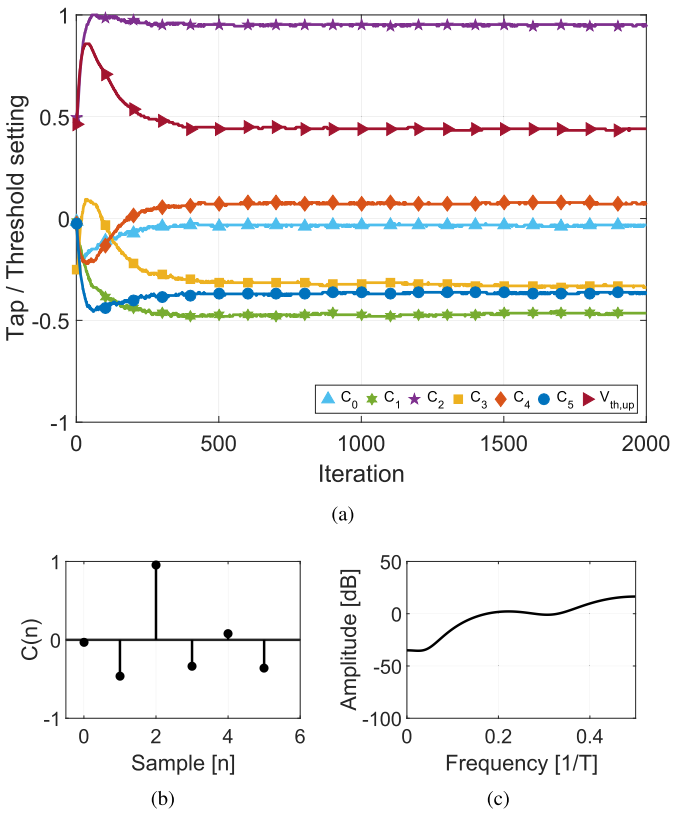


Fig. 17. Parameter evolution for DB communication optimization at 100 Gb/s over 1.5 m Twin-Ax cable with half of the transmitter swing (a). The final impulse response is given in (b), the amplitude response of this filter is given in (c).

the decision clock of the receiver is a delayed version of the transmitter clock. The delay of the receive clock is tuned manually to obtain optimal link performance. At the output of the demultiplexer (DEMUX), one out of four PRBS streams is analyzed in the FPGA. Due to the properties of the PRBS code, the analysis of one stream will give the same performance as the analysis of all four combined streams as the full rate stream can be estimated. This estimation will yield the signal $d_{\text{decoded}}(n)$. In the FPGA, a PRBS aligner is implemented to lock on the incoming PRBS stream to generate $d_{\text{lock}}(n)$ and to calculate the BER. The signal $d_{\text{lock}}(n)$ is used for the estimation of $x(n)$. All three signals can be read each iteration during which one PRBS7 period is analyzed at a time. As 127 bits are read each iteration, an averaged update over these bits is performed. Both the FFE parameters $C(n)$ and the decoding thresholds V_{up} and V_{down} are updated at the end of each iteration which closes the feedback loop. All update calculations are performed in Matlab®. In the future, a hardware implementation of the algorithm is desired. The amount of needed resources is comparable to digital LMS implementations, however, the complexity can be lowered considerably by serializing all calculations as short calculation times are not critical due to the already large timing overhead of the feedback channel.

B. Experimental Results

Using the developed chipset and test environment, the designed algorithm is tested at 100 Gb/s. For the test run, a 1.5 m Twin-Ax cable is used. The insertion loss profile of the channel is already given in Fig. 10 and shows a loss of −36 dB at 50 GHz.

To start an experiment, synchronization between the TX and RX is performed using a starting high pass filter FFE response ($C = [0\ 0\ 0.5\ -0.25\ 0\ 0]$) to obtain a clear quarter-rate eye diagram. These filter parameters are not critical and can be obtained using an initial codebook search. Next, the full rate data communication is started where the clock is manually tuned such that the lowest BER is achieved.

As in simulation, the value of λ starts at 0.1 and drops each iteration such that a value of 0.01 is reached after 500 iterations. The threshold update loop has a fixed step-size value λ_{th} of 0.01. As mentioned previously, the optimization is performed using a PRBS7 training sequence. The evolution of the six FFE coefficients and the level of $V_{\text{th,up}}$ are shown in Fig. 16a. Stable operation is achieved after approximately 450 iterations. The final impulse response and frequency response are respectively given in Fig. 16b and Fig. 16c. The response of Fig. 16c shows the required high pass behavior. The slow ripple on the frequency response follows the ripple in the channel response, shown in Fig. 10. The BER over the link after optimization is below 10^{-13} using PRBS7 and PRBS9 as test sequence. Using PRBS15, the BER increases to a value of 5×10^{-11} due to the additional ISI components. Analyzing the evolution of the value for $V_{th,up}$ in Fig. 16a, saturation is visible after a few iterations. At this point, the signal at the input of the RX is too high to place $V_{\text{th,up}}$ inside the noise band. This saturation removes the

influence of the threshold update loop completely. To observe the influence of the threshold loop in practice, the gain of the FFE amplifiers is halved such that saturation cannot occur anymore. For this experiment, the initial value of $V_{\text{th,up}}$ is also halved compared to the previous experiment to make a fair comparison. The result of the experiment is shown in Fig. 17, showing unsaturated adaptation of $V_{\text{th,up}}$. Due to nonlinearities in the gain response of the taps, the final impulse response is not a scaled version of the results in Fig. 16b, however, the same trends are visible. Due to the halved signal swing, the BER has increased significantly to a value of 2.3×10^{-5} using PRBS7 (6.7×10^{-5} using PRBS9).

VI. Conclusion

In this paper, an adaptive equalizer algorithm is developed, intended for transmit-side equalization at a rate of 100 Gb/s. It shapes the NRZ input data into a DB wave at the output of the channel. The adaptation is performed under stringent constraints enforced by the architecture of the chipset which is optimized for high speed communication. The adaptation algorithm is a SSLMS based algorithm. It is possible to perform NRZ to DB transmit-side equalization based on a crude a priori estimation of the channel loss. In addition to the development of the adaptation engine, practical issues concerning the architecture are solved. The limited number of decoding thresholds is handled by an alteration of the error definition. The synchronization problem between the transmitter and receiver is solved via the usage of a synchronization at the quarter-rate and an estimation method to evaluate the synchronization. The designed algorithm is simulated to perform a thorough analysis of the convergence behavior of the algorithm used for adaptively equalizing DB signals at 100 Gb/s. Experimental hardware runs are executed, completing a challenging 100 Gb/s serial link over a 1.5 m Twin-Ax cable. With this experimental verification, the capabilities of an adaptive algorithm for shaping DB with a transmit-side equalizer are shown.

References

[1] J. Van Kerrebrouck *et al.*, "100 Gb/s serial transmission over copper using duo-binary signaling," in *Proc. DesignCon*, 2016, pp. 1–25.

[2] J. H. Sinsky, A. Adamiecki, and M. Duelk, "10-Gb/s electrical backplane transmission using duobinary signaling," in *IEEE MTT-S Int. Microw. Symp. Dig.*, vol. 1. Jun. 2004, pp. 109–112.

[3] K. Yamaguchi *et al.*, "12 Gb/s duobinary signaling with $\times 2$ oversampled edge equalization," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2005, pp. 70–71.

[4] Y.-M. Ying, I.-T. Lee, and S.-I. Liu, "A 20 Gb/s adaptive duobinary transceiver," in *Proc. IEEE Asian Solid State Circuits Conf. (A-SSCC)*, Nov. 2012, pp. 129–132.

[5] G. Malik and A. S. Sappal, "Adaptive equalization algorithms: An overview," *IJACSA Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 3, pp. 62–67, 2011.

[6] Y. Fang, L. Chen, A. Jaiswal, K. Hofmann, and P. Gregorius, "Adaptive equalizer training for high-speed low-power communication systems," in *Proc. Euromicro Conf. Digit. Syst. Design*, Sep. 2013, pp. 745–751.

[7] E. Coersmeier and E. Zielinski, "Comparison between different adaptive pre-equalization approaches for wireless LAN," in *Proc. 13th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, vol. 3. Sep. 2002, pp. 1136–1140.

[8] A. Hekkala *et al.*, "Predistortion of radio over fiber links: Algorithms, implementation, and measurements," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 3, pp. 664–672, Mar. 2012.

[9] T. Suttorp and U. Langmann, "A 10-Gb/s CMOS serial-link receiver using eye-opening monitoring for adaptive equalization and for clock and data recovery," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2007, pp. 277–280.

[10] B. Analui, A. Rylyakov, S. Rylov, M. Meghelli, and A. Hajimiri, "A 10-Gb/s two-dimensional eye-opening monitor in 0.13-μm standard CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2689–2699, Dec. 2005.

[11] H. Won *et al.*, "A 28-Gb/s receiver with self-contained adaptive equalization and sampling point control using stochastic sigma-tracking eye-opening monitor," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 3, pp. 664–674, Mar. 2017.

[12] W.-S. Kim, C. Seong, and W.-Y. Choi, "A 5.4-Gbit/s adaptive continuous-time linear equalizer using asynchronous undersampling histograms," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 9, pp. 553–557, Sep. 2012.

[13] J. H. Sinsky, M. Duelk, and A. Adamiecki, "High-speed electrical backplane transmission using duobinary signaling," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 1, pp. 152–160, Jan. 2005.

[14] A. Lender, "The duobinary technique for high-speed data transmission," *Trans. Amer. Inst. Elect. Eng. I, Commun. Electron.*, vol. 82, no. 2, pp. 214–218, May 1963.

[15] T. De Keulenaer *et al.*, "56+ Gb/s serial transmission using duo-binary signaling," in *Proc. DesignCon*, vol. 10TH-3. 2015, pp. 1–23.

[16] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. 23, no. 9, pp. 921–934, Sep. 1975.

[17] J. Buiting, "Introduction to duobinary encoding and decoding," *Elektor Electron.*, pp. 50–52, Jan. 1990.

[18] T. D. Keulenaer *et al.*, "84 Gbit/s SiGe BiCMOS duobinary serial data link including serialiser/deserialiser (SERDES) and 5-tap FFE," *Electron. Lett.*, vol. 51, no. 4, pp. 343–345, 2015.

[19] M. Verplaetse *et al.*, "Real-time 100 Gb/s transmission using 3-level electrical duobinary modulation for short-reach optical interconnects," *J. Lightw. Technol.*, to be published.

[20] Y. Ban *et al.*, "A wide-band, 5-tap transversal filter with improved testability for equalization up to 84 Gb/s," *IEEE Microw. Wireless Compon. Lett.*, vol. 25, no. 11, pp. 739–741, Nov. 2015.

[21] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349–1387, Sep. 1985.

[22] K. H. Mueller and D. A. Spaulding, "Cyclic equalization—A new rapidly converging equalization technique for synchronous data communication," *Bell Syst. Tech. J.*, vol. 54, no. 2, pp. 369–406, Feb. 1975.

[23] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2008.

[24] C. Rohrs, C. Johnson, and J. Mills, "A stability problem in sign-sign adaptive algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 1986, pp. 2999–3001.

[25] B.-E. Jun, D.-J. Park, and Y.-W. Kim, "Convergence analysis of sign-sign LMS algorithm for adaptive filters with correlated Gaussian data," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 2. 1995, pp. 1380–1383.

[26] E. Mammei *et al.*, "Analysis and design of a power-scalable continuous-time FIR equalizer for 10 Gb/s to 25 Gb/s multi-mode fiber EDC in 28 nm LP CMOS," *IEEE J. Solid-State Circuits*, vol. 49, no. 12, pp. 3130–3140, Dec. 2014.

[27] M. I. Mossi, N. W. D. Evans, and C. Beaugeant, "An assessment of linear adaptive filter performance with nonlinear distortions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 313–316.

Michiel Verplaetse was born in Ghent, Belgium, in 1993. He received the degree in electrical engineering from Ghent University, Belgium, in 2016, where he is currently pursuing the Ph.D. degree. In 2016, he joined the IDLab Design Group, Ghent University. His technical interests are mixed signal circuit design, mainly focusing on equalization structures.

Timothy De Keulenaer received the M.Sc. and Ph.D. degrees in applied electronical engineering from Ghent University, Ghent, Belgium, in 2010 and 2015, respectively. He is currently with BiFAST, a spin-off of Ghent University IDLab. His main interests are high speed BiCMOS integrated circuit design for 112Gb/s, the development of active copper cables using duobinary technology to enable serial data rates of 112 Gb/s within cloud data centers, and signal integrity aspects for back-plane communication. He holds three duobinary related patents.

Arno Vyncke received the M.Sc. and Ph.D. degrees in applied electronical engineering from Ghent University, Ghent, Belgium, in 2010 and 2016, respectively. He is currently with BiFAST, a spin-off of Ghent University IDLab. His research is focused on clock and data recovery circuits for nextgeneration low-power passive optical networks and the business development for active copper cables enabling high-speed in-rack data center connectivity.

Ramses Pierco received the master's degree in applied electrical engineering and the Ph.D. degree in applied electrical engineering from Ghent University, Belgium, in 2010 and 2015, respectively. He was with the INTEC Design Laboratory, Department of Information Technology, Ghent University. He is currently with BiFAST, a spin-off of Ghent University IDLab. He is currently responsible for the technology development within BiFAST in which his focus lies on further integration of a duobinary transceiver chipset aiming at data rates ranging from 40 Gb/s up to 112 Gb/s.

Renato Vaernewyck received the M.Sc. and Ph.D. degrees in electronical engineering from Ghent University, Ghent, Belgium, in 2010 and 2014, respectively. He is currently with Bi-FAST, a spin-off of Ghent University IDLab. He has been a Research Assistant with the INTEC Design Laboratory, Ghent University, since 2010. His research focuses on high speed, high-frequency (opto-)electronic circuits and systems, and further integration of a duobinary transceiver chipset aiming at data rates ranging from 40 Gb/s up to 112 Gb/s within BiFAST.

Joris Van Kerrebrouck (S'14) was born in Ghent, Belgium, in 1989. He received the M.S. degree in electronics engineering from Ghent University, Belgium, in 2014, where he is currently pursuing the Ph.D. degree. In 2014, he joined the IDLab Design Group, Ghent University, where he was involved in high-speed electrical transceivers. His current fields of interest are high-speed SiGe BiCMOS analog circuits and systems.

Johan Bauwelinck received the Ph.D. degree in applied sciences and electronics from Ghent University, Belgium, in 2005. Since 2009, he has been a Professor with the INTEC Department, Ghent University, and since 2014, he has been leading the IDLab Design Group. He was and is active in the EU-funded projects GIANT, POWERNET, PIEMAN, EuroFOS, C3-PO, Mirage, Phoxtrot, Spirit, Flex5Gware, Teraboard, Streams, and WIPE conducting research on advanced electronic integrated circuits for next generation transport, metro, access, datacenter and radio-over-fiber networks. He has promoted 18 PhDs and co-authored over 150 publications and ten patents in the field of high-speed electronics and fiber-optic communication. His research focuses on high-speed, highfrequency (opto-) electronic circuits and systems, and their applications on chip and board level, including transmitter and receiver analog front-ends for wireless, wired and fiber-optic communication or instrumentation systems.

Guy Torfs (M'13) received the B.E. degree in applied electronics and the Ph.D. degree in applied sciences, electronics from Ghent University, Belgium, in 2007 and 2012, respectively. Since 2011, he has been with IMEC, associated with Ghent University, where he became an Assistant Professor in 2015. His research focuses on high-speed mixed signal designs for wireless baseband and fiber-optic and backplane communication systems, including digital signal processing and calibration, analog equalization circuits and clock and data recovery systems. He was a co-recipient of the 2015 Design Con Best Paper Award in the High-Speed Signal Design category. In 2014, as part of the Bi-PON and Cascaded Bi-PON Team, he was rewarded with the Greentouch 1000x Award.