DesignCon® 2025
WHERE THE CHIP MEETS THE BOARD

Genetic Algorithm-Driven IBIS-AMI Optimization for Robust 200G SerDes Design

Adrien Auge, Alphawave Semi
adrien.auge@awavesemi.com

Tripp Worrell, MathWorks
tworrell@mathworks.com

Walter Katz, MathWorks
wkatz@mathworks.com

Richard Allred, MathWorks
rallred@mathworks.com

Eric Brock, MathWorks
ebrock@mathworks.com

Abstract

Copper connectivity is increasingly becoming a bottleneck in the rapidly evolving compute landscape that supports artificial intelligence (AI). The design of 200G SerDes faces aggressive timelines, necessitating a robust pre-silicon simulation methodology. A significant performance limiter in SerDes technology is the adaptation routine that determines the optimal equalization (EQ) for a given channel. The routine's effectiveness, whether in laboratory settings or simulations, heavily depends on the underlying optimization algorithm's robustness. Due to runtime constraints, traditional simulation models often fail to represent hardware precisely when searching for optimal equalization recipe instead focusing on achieving the required simulation throughput these types of electrical simulations demand by adding a certain level of abstraction. Even with modeling simplification, proper adaptation of the model equalization settings necessitates a considerable amount of the total computation to be put on this optimization task. This paper proposes enhancements to existing modeling techniques to enable a faster and more accurate representation of SerDes, thus narrowing the gap between model abstraction and actual silicon/firmware performance. By adopting proven optimization strategies, such as Genetic Algorithms, over traditional brute-force methods for driving adaptation, we aim to improve the efficiency at arriving at an optimal equalizer recipe for a given channel. Along with efficiency improvement in optimizing the SerDes registers, further modeling enhancement is proposed in terms of the adaptation cost-function promising to enhance prediction accuracy across a broader range of channels for the next generation of copper links, offering significant advancements in the field.

Author(s) Biography

Adrien Auge joined Alphawave Semi in 2021 after spending more than eight years in the signal integrity field, working on parallel and serial interface design for memory and GPU products at SK Hynix and Intel. He also worked at Synopsys on SI modeling, where he led the development of industry-standard IBIS-AMI models for their IP portfolio and released the company's first ADC-based IBIS-AMI model. Adrien's current role at Alphawave IP is to provide accurate SI models to IP customers for their off-chip design needs and enablement and signal integrity integration support. He received his BS in IT Engineering from France (IMT, 2010) and his MS in Electrical Engineering from Korea (KAIST, 2012).

Tripp Worrell joined MathWorks in 2017 after spending three years at SiSoft managing the development of their award-winning EDA simulation software. Before this, he worked for seven years at Cisco Systems as a Signal Integrity and High-Speed Design Engineer, where he was responsible for designing and verifying large enterprise networking linecards and backplanes. His current role as Development Manager overseeing SerDes Toolbox, Mixed-Signal Blockset, and Signal Integrity Toolbox leverages both his hardware and software experience to best support the needs of leading-edge clients at MathWorks. Tripp received his BS in both Computer and Electrical Engineering (2005) and his MS in Computer Engineering (2007) from The North Carolina State University.

Walter Katz, Ph.D., is a chief scientist at MathWorks, Inc., with 50 years of experience developing CAD and single integrity software at Mentor, Cadence, and SiSoft. He is one of the principal authors of the IBIS-AMI standard and participated in the IEEE-370 and 802.3 standards. Dr. Katz has developed numerous SerDes channel optimization methods and is an author of the IBIS-AMI backchannel BIRDs. He is currently developing next-generation optimization methods for Power Supply Design. Dr. Katz received his B.S./M.S. (1966) from the Polytechnic Institute of Brooklyn and his Ph.D. (1971) from the University of Rochester.

Richard Allred, Ph.D., is a principal software engineer at MathWorks, Inc., with over a decade of signal integrity design experience at Intel, Inphi, and SiSoft. He has authored dozens of IBIS-AMI models and is currently developing signal integrity tools at MathWorks. Dr. Allred received his B.S./M.S. (2006) and Ph.D. (2021) from the University of Utah.

Eric Brock is an experienced engineer who spent nearly 20 years in high-speed hardware design and testing. He is now developing tools for the software line at MathWorks. In Eric's previous role at SiSoft, he developed IBIS-AMI models and designed DDR and SERDES interfaces. At MathWorks, he contributes to the SerDes and Signal Integrity Toolboxes, where he is responsible for implementing new functionality in MATLAB® and Simulink®. Eric holds dual B.S. degrees in Electrical Engineering and Physics from Portland State University and has authored several papers presented at DesignCon.

What is an IBIS-AMI?

A system integrator could validate a SerDes link in a platform in many ways before final design sign-off and manufacturing. Such tasks aim to de-risk any potential performance issue arising from the integrity of the signals traveling from the source to the decision point.

Various standards have introduced electrical compliance points and metrics facilitating such tasks from an interoperability perspective. AMI is a standard defined by the IBIS (I/O Buffer Information Specification) organization. It allows for creating executable models that simulate the behavior of SerDes components, such as transmitters and receivers, in a high-speed digital link. An IBIS-AMI model can provide more granular feedback and allow designers to potentially use more of the available design margins because it is more closely related to the actual silicon integrated in the system and not a "bare-minimum" representation of a compliant device.

There are two important factors driving the entire modeling strategy behind IBIS-AMI models and determining the added value their usage can bring. First and foremost, models representing a particular piece of silicon integrated into a system, need to closely match (under certain conditions) the performance of the actual device; correlation is primordial. The second and equally important piece is that such models should provide a throughput that is suited for an off-chip signal integrity simulation flow. System designers and integrators must be able to run off-chip design-trade off and predict system performances using these models over large design permutations.

IBIS-AMI simulations can be either statistical or time domain. Statistical simulations use channel impulse response and equalization impulse responses to calculate an Eye Diagram at the receiver latch. In contrast, time domain simulations generate bit-by-bit time domain waveforms at the receiver latch. Statistical simulations typically provide the best thruput, can explore larger interconnect solution space, and are most often used to extract design sensitivities and trends. Time domain simulations are slower but more accurate. They can model non-linear behavior and can be used for more detailed prediction profiles. Both statistical and time domain simulations can include adaptation algorithms to optimize the channel performance. The latest generation of SerDes components have very complex equalization filters making adaptation problematic both in the hardware and in the adaptation modeling.

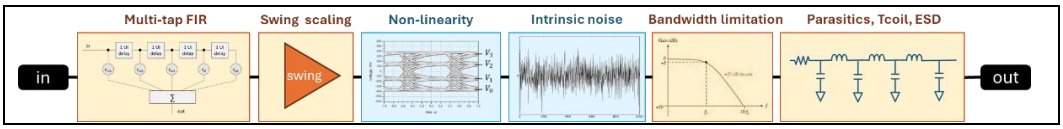
Industry reference transceiver models implemented in tools such as COM, Seasim, and MATLAB already provide a good benchmark for how computationally efficient an IBIS-AMI model should be. This paper discusses a SerDes simulation tradeoff related to IBIS-AMI. It proposes a methodology to scale and address the omnipresent theme of accuracy against the speed dilemma that engineers of all disciplines face.

1. DSP based IBIS-AMI modeling

1.1. Transmitter modeling (high level/block diagram)

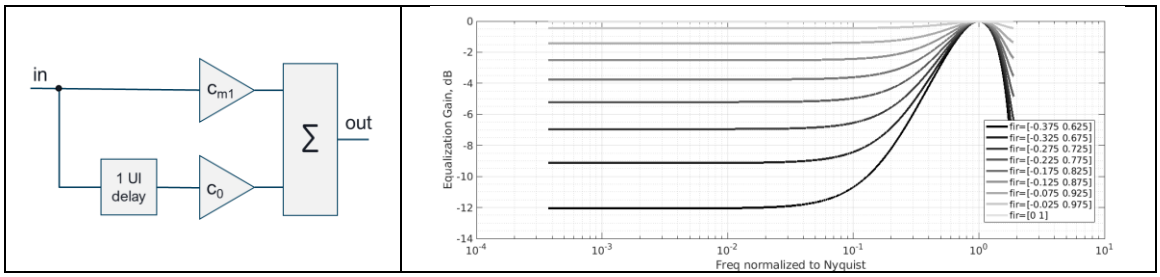
Figure 1 depicts the transmitter modeling for a traditional SerDes (analog or DSP), including its key behavioral blocks.

Figure 1: Transmitter model block diagram



Many of the transmitter blocks represent non-tunable intrinsic impairments of the circuit, such as noise, parasitic, or non-linearity. Their characteristics are extracted through front-end circuit post-layout simulations. Conversely, FIR is the only signal conditioning knob on the transmit side that can provide a significant portion of the total SerDes equalization capability, amounting for up to 30% of the total Nyquist gain required to overcome bump-bump loss for a 40dB channel. This FIR block, composed of multiple pre- and post-cursor correcting taps, needs to be systematically tuned for every channel. Figure 2 shows a sample FIR coarse sweep for a single pre-cursor adjustment yielding an effective Nyquist boost of up to 12dB.

Figure 2: Transmitter FIR transfer function sweep

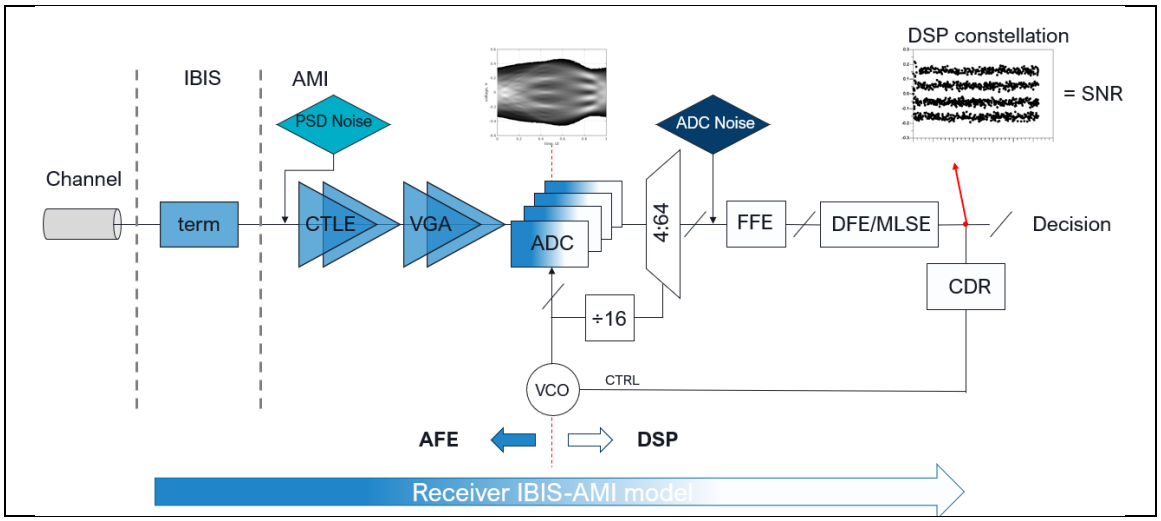


The number of pre- and post-cursors and the correctable range for each, which is dictated by the design and reflected in the behavioral FIR block. The Auto-Negotiating Link Training (ANLT) feature handles this from a firmware perspective for optimizing FIR coefficients for a SerDes IP. A similar approach is taken on the behavioral modeling side, where the coefficient of each FIR tap will be adjusted through the receiver model's "global" adaptation loop. The following sections will present a more detailed overview of the adaptation strategies for IBIS-AMI models.

1.2. Receiver modeling

The receiver modeling for a Digital Signal Processing (DSP) architecture is described below. The data path can be decomposed into two sections, Analog Front End (AFE) and DSP, which roughly equally contribute to the receiver's total equalization power. The boundary between the two areas is delimited by the Analog-to-Digital conversion unit (ADC) and the sampling of the continuous signal at the analog portion's last amplifier/buffer.

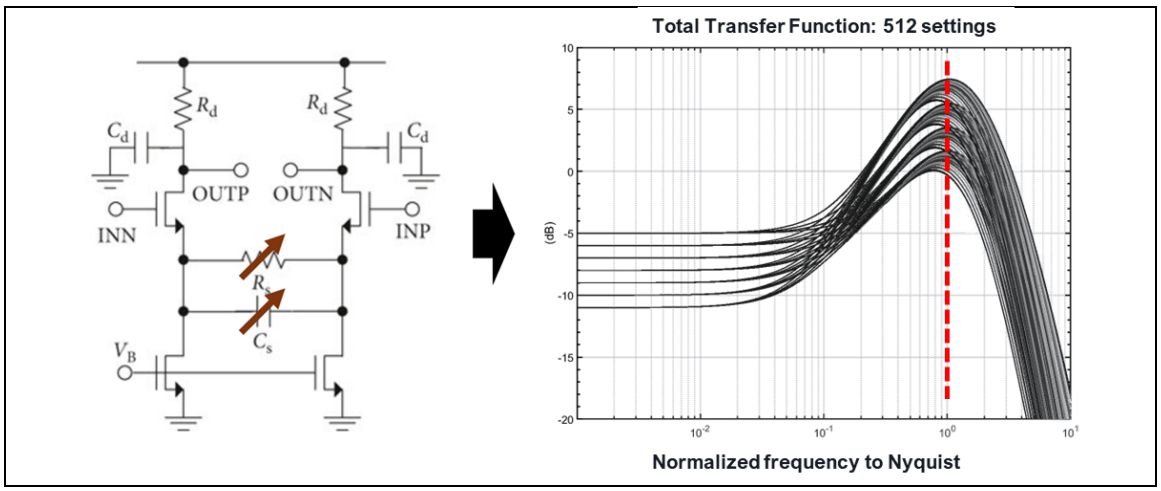
Figure 3: Receiver block diagram



The entire analog front-end provides 30 to 35% of the total signal equalization required for copper-links. The continuous-time-linear equalizer (CTLE) providing the Nyquist boost is typically architected in several stages to optimize peaking range, bandwidth and also provide granular mid-band boosting capabilities. Another function of the analog front-end in DSP transceivers is to properly scale the signal prior to ADC conversion with a Variable Gain Amplifier (VGA). The broadband gain characteristic of the VGA is essential for adjusting the signal concerning the ADC full-scale dynamic range to maximize the performance. Along with the sizing of the Nyquist boost of the CTLE stages, the VGA gains are part of the receiver's global adaptation.

Each analog front-end stage contains several parameters to adjust the incoming signal's overall filtering characteristics. Figure 4 shows an example of a differential amplifier with a degeneration capacitor and resistor adjustment (left) to tune the stage filtering response (right).

Figure 4: Analog equalizer stage filtering



1.3. Adaptation

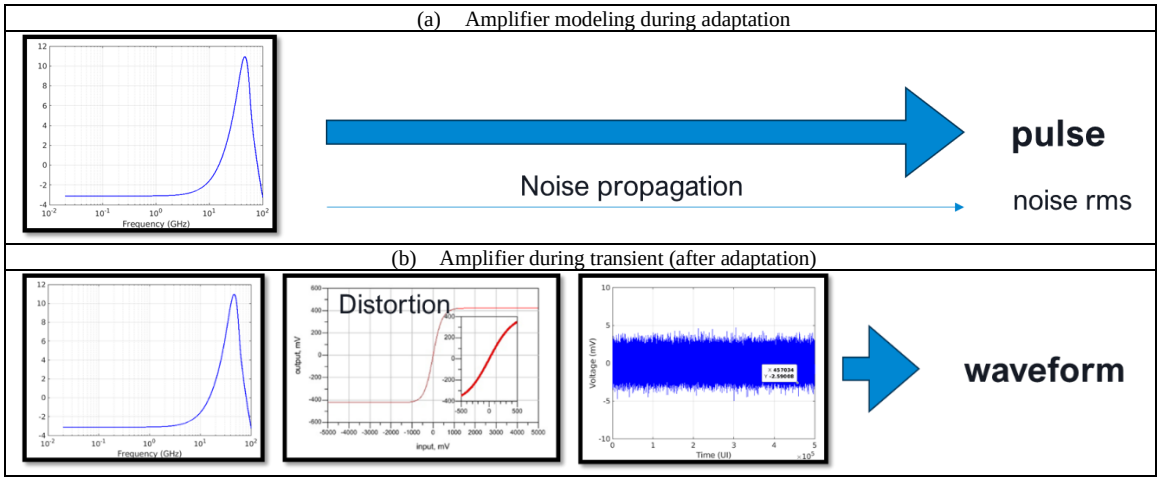
Optimal equalization setting adaptation is central to any SerDes transceiver link. Algorithm robustness is key to achieve the best BER performance. As described in prior sections, both transmit and receive functions contain various tunable parameters to cover different equalization needs for a wide range of channels. It is very important to apply and balance the adequate amount of equalization amongst the transmit, receive analog, and receive DSP blocks for each channel. A signal with too much or not enough equalization applied will underperform. Similarly, relying on a single block to handle the entire equalization will not be efficient either, as every equalizer has its trade-off regarding noise and distortion it introduces to the signal when removing linear residual ISI. Thus, a robust methodology is required.

In hardware, the sampled signal at the ADC output is used to collect statistics and derive equalizer adjustment sequences until certain target metrics are met. This requires collecting millions of data samples to complete and converge. During typical bring-up and validation of IP test chips, a substantial amount of effort is dedicated to fine-tuning these sequences across the whole Nyquist loss range and avoiding performance bottlenecks due to sub-optimal convergence of the equalizer registers.
In simulation, models relying on a 1:1 algorithm match with the firmware are not practical for many reasons, including excessive simulation runtime and limited compute throughput. Instead, the traditional approach employed by many simulators and channel electrical compliance reference transceivers, such as COM for IEEE/OIF and Seasim for the PCIe side, is to use the channel “fingerprint” (transfer function), which is known beforehand in a simulation environment. This fingerprint is run through an exhaustive search of equalizer filter combinations to obtain an “equalized” signal response, from which a Figure of Merit FoM is derived. The FoM is used to grade all the permutations and select an optimal setting.

1.3.1.Fitness calculation

The goodness metric most widely used to optimize the signal conditioning chain is a form of Signal-To-Noise ratio (SNR) derived from pulse response filtering of the channel signal path with consideration for intrinsic/extrinsic noise propagation. One critical distinction to make before going into the details of the fitness calculation is that with IBIS-AMI modeling, there are effectively two equalizer chains used inside the IBIS-AMI model. The first equalizer chain (Figure 5 (a)) is a simplified version meant for linear pulse responses and is used during the simulation's adaptation phase. The second equalizer chain (Figure 5 (b)) contains the same linear response modeling but introduces additional non-ideal impairments and can only be applied to a full-size waveform. This equalizer contains intrinsic non-idealities of the amplifier stage, such as non-linear distortion for large signals, ADC quantization, ADC clipping, and more detailed intrinsic noise injection. From a modeling perspective, it is a closer representation of the physical behavior of the signal propagating through the actual circuit. It is used during the time-domain portion of the simulation after the adaptation is complete.

Figure 5: Amplifier stage modeling

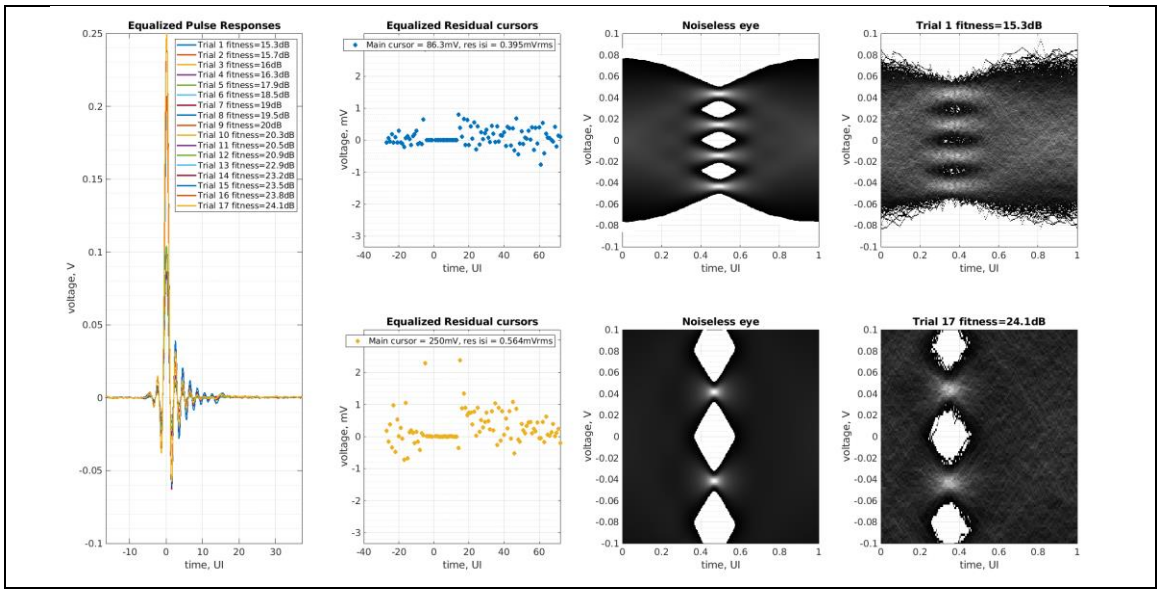


With the above, the actual fitness metric calculation is summarized in Figure 6, which illustrates the pulse responses and their corresponding fitness for a handful of equalizer combinations over a reduced optimization subset. The equalized pulse responses corresponding to several combinations with different performances are shown and superposed on the left side of the figure. Residual cursor ISI and reconstructed eye diagrams, as well as the calculated SNR, are shown on the top and bottom rows for the best- and worst-case EQ settings, respectively.

Both worst- and best-case scenarios show that the resulting equalizer properly cancels the Inter-Symbol-Interference (ISI) on the signal path for this channel (32dB bump to bump). This can be seen clearly in both the residual ISI plot in the second column and in the eye diagrams depicted in the third column. These eye diagrams only contain the residual channel ISI after applying the equalizer. Once factoring in the noise propagating through

the equalizer (both intrinsic to the equalizer and external), the top row (worst case) now turns out to be a sub-optimal setting. It shows an equalizer combination yielding a small signal amplitude (lower than the available ADC range and non-linear region of the Analog-Front-End), resulting in a low signal-to-noise ratio, which translate to a closed eye diagram in the last column. In contrast, the bottom row (best case) shows the optimal settings, providing the maximum signal amplitude with the same residual ISI correction, effectively better signal-to-noise ratio performance, and an open eye once noise is added in the last column.

Figure 6: Fitness calculation example



1.3.2.Search algorithm

The above graphically described fitness calculation is currently embedded in the IBIS-AMI model search algorithm described by the pseudo-code in Figure 7. In this algorithm, a for-loop search is laid out for each of the equalizer blocks in the SerDes chain, including TX FIR, CTLE (Nyquist boost), and CTLE (mid-band boost). The fitness is thus calculated exhaustively across all combinations, and the optimal EQ parameters recipe is retained upon completion.

Figure 7: Equalizer Optimization loop

```
function [ best_tx_fir, best_ctle, best_vga ] = adaptation( pulse_after_channel, pulse_crosstalk )
%% THRU SIGNAL INPUT
% pulse_after_channel <-input
%% AGGRESSORS SIGNAL INPUT
%pulse_crosstalk <- input
%% ADAPTATION PSEUDO-LOOP
fom_best = 0 ;
for tx_fir = 1 : N1 % loop over the n-taps FIR entire range
    for ctle = 1 : N2 % loop over the multi stage CTLEs entire range
        for vga = 1 : N3 % loop over the multi stage VGAs entire range
            pulse_after_tx = filter1( pulse_after_channel, tx_fir ) ; % apply TX FIR  filter
            pulse_after_ctle = filter2( pulse_after_tx, ctle ) ; % apply RX CTLE filter
            pulse_after_vga = filter3( pulse_after_ctle, vga ) ; % apply RX VGA gain
            pulse_final = ffe_dfe_zeros_force( pulse_after_vga, dsp_config ) ; % calculate (zeros force) and apply the DSP filter based on the FFE/DFE tap ranges
            noise = calculate_noise_propagation( pulse_crosstalk, tx_fir, ctle, vga ) ; % calculate the extrinsic (crosstalk) and intrinsic (circuit self-induced) noise propagation and amplifiction
            fom = calculate_snr(pulse_final,noise) ;
            if fom > fom_best
                best_ctle = ctle ;
                best_vga = vga ;
                best_tx_fir = tx_fir ;
                fom_best = fom ;
            end
        end
    end
end

end
```

1.4. Problem scaling

With the demand for longer copper reach and higher Nyquist frequencies, the size and complexities of the amplifier chain are growing. The amplifiers in typical SerDes must handle a wide range of channels (boost) and cover wider Nyquist ranges (peaking frequencies) to support bit rates of up to 200Gb/s. From an implementation perspective (power, area, efficiency), these improvements can be separated using several stages for a single function, so a block like the CTLE can be constructed using multiple amplifier stages, yielding the desired response. The transmitter FIR also requires additional pre- and post-cursor tap coverage. On the receiver side, DSP features are also scaling in complexity and need their own optimization, larger bank of RX FFE roaming taps can cover longer post cursor ranges, features such as the choice between DFE or maximum likelihood sequence estimation (MLSE) for the main post cursor correction and decision circuit needs to be tuned as well. The optimization problem of finding the optimal EQ combination for these larger parameter pools quickly becomes challenging in simulation and requires increasing time and resources to solve.

Below is a rough comparison of the optimization problem based on our current generation 100Gb/s SerDes models and the upcoming 200Gb/s Long-reach SerDes model in terms of EQ permutations that are needed inside the IBIS-AMI model.

Table 1: EQ parameter optimization pool

Block	EQ Parameter	100G permutations	200G permutations
TX	FIR	~4,000	~15,000
RX AFE	Mid-Band Boost	~4	~10
	CTLE	~25	~500
	VGA (is self-adapted)	~1	~1
RX DSP	FFE anchor point	~15	~15
	DFE 1st tap ratio	~6	~6
	FFE roaming	~3	~6
Total		**36 million**	**6.75 billion**

With the current 200Gb/s SerDes adaptation, the problem size changes by 2.5 orders of magnitude. Passing this level of computation resource increase onto the model's end-user is not feasible as it would break the entire IBIS-AMI simulation flow and its usefulness. Using additional modeling abstractions or approximation and reducing the problem size were debated but not pursued as they would sacrifice too much of the modeling accuracy, range of validity, and overall reliability of the model. Ultimately, a shift in the optimization approach was deemed necessary, and this will be the discussion of the remainder of this paper.

2. Optimization techniques

In our search for alternative optimization strategies, we explored a variety of techniques to identify the most effective method for our SerDes Model adaptation use case. The primary focus was on benchmarking the performance and convergence of alternative methods against the Brute Force approach currently used in our modeling and described in the previous section. The following three candidates were short listed:

- Hybrid Search
- Random Search
- Genetic Algorithm

2.1. Technique Overview

Including the original brute force approach used previously, the four optimization techniques are described in a bit more detail below:

- **Brute Force**: This is the currently used exhaustive search method that evaluates the fitness of every possible solution in the equalizer search space.
    - Pros: Guarantees finding the global optimum due to its comprehensive nature.
    - Cons: Computationally expensive and impractical for large or complex search spaces due to exponential growth in required evaluations.
- **Hybrid Search** (Coarse/Fine Sweep): This method combines a coarse search to identify promising regions of the search space, followed by a finer sweep to refine the solution within these regions.
    - Pros: Balances thoroughness and efficiency by reducing the number of evaluations needed compared to brute force.
    - Cons: It may still require significant computational resources, and the initial coarse sweep may miss some optimal regions.

  With fewer EQ blocks to optimize and a clear understanding of each gradient impact on the fitness function, the hybrid search is a strong candidate and can achieve significant gain over an exhaustive search. However, it becomes difficult to develop an efficient and robust hybrid search algorithm with multiple complex data path EQ blocks interacting with each other in terms of fitness function. Moreover, the underlying equalizer characteristics and architecture of the overall RX data path will likely necessitate this approach to be revisited and re-evaluated for validity when changes occur.

- **Genetic Algorithm** (GA): Inspired by the principles of natural selection, GAs use a population of potential solutions that evolve over generations through selection, crossover, and mutation.
    - Pros: Effective at exploring complex, high-dimensional search spaces and escaping local optima.
    - Cons: Requires careful tuning of parameters and may involve higher computational costs than some heuristic methods.
- **Random Search** (Considered, but not benchmarked): This technique involves randomly sampling the search space to find an optimal solution.
    - Pros: Simple to implement and does not require gradient information or problem-specific knowledge.
    - Cons: Typically inefficient for large or complex problems due to its unguided nature, often resulting in suboptimal solutions.

The Genetic Algorithm emerged as a superior and more versatile method through our initial research, the. Given the successful implementation of this algorithm within our silicon validation team's test chip register tuning process during the silicon bring-up-and-characterization phase, we ultimately decided to pursue this approach as our main candidate and initiate the integration within the IBIS-AMI modeling framework.

The following sections delve into the details of GA, a comparative analysis of the techniques under study, and highlight the strengths and limitations observed during our testing.

2.2. Genetic Algorithm details

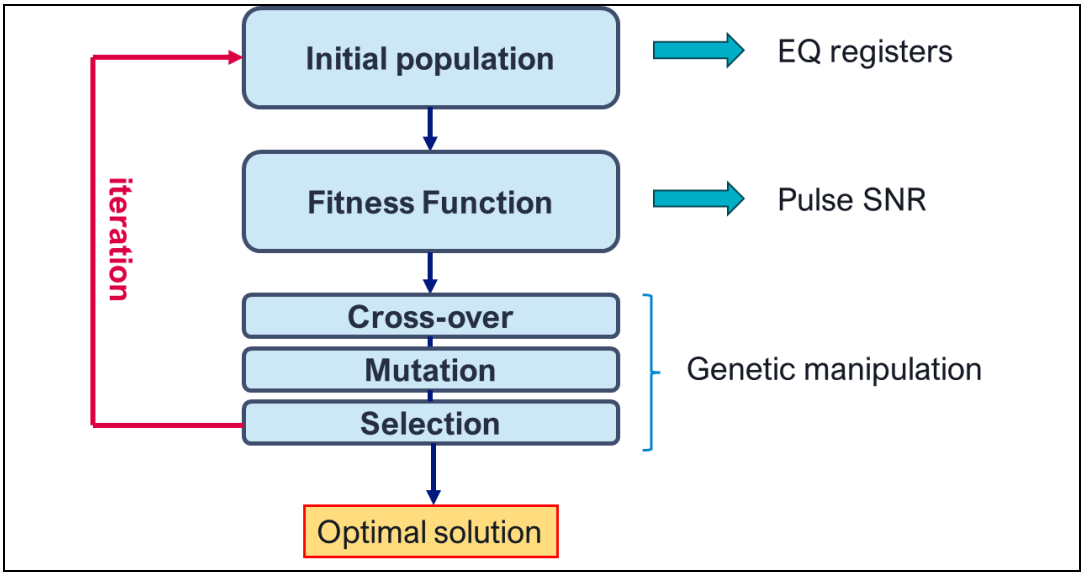
A genetic algorithm (GA) is a search heuristic inspired by the natural selection process. It solves optimization and search problems by mimicking the evolutionary principles of selection, crossover, and mutation. GA is used across many industries due to its ability to solve complex optimization problems such as:

Engineering Design:
- Structural Optimization: Used to optimize the design of structures, such as bridges and aircraft components, to minimize weight while maximizing strength and stability.
- Control Systems: Helps designing robust control systems by optimizing parameters and configurations for performance and stability.

Biotechnology and Bioinformatics:
- Protein Folding: Applied to predict protein structures by optimizing the folding process to achieve the lowest energy conformation.
- Gene Expression Analysis: Used to identify patterns in gene expression data, aiding in understanding genetic interactions and disease mechanisms.

Figure 8 describes the GA iteration process highlighting the key steps in the optimization loop.

Figure 8: Genetic algorithm diagram



Further details on the terminology specific to the GA algorithm are provided below:

Population: A set of potential problem solutions. Each solution is usually encoded as a chromosome.

Chromosome: A solution representation. Each chromosome consists of genes, which are the register values controlling the various equalizer blocks.

Fitness Function: A function that evaluates and assigns a figure of merit (FoM) score to each chromosome in the population. The fitness score determines how well the chromosome solves the problem (improving the link performance). As much as the optimization algorithm itself, the fitness function is essential and needs to proxy the true system performance (BER) across a wide range of system conditions/inputs. The following section will provide further discussion around an appropriate simulation-based fitness.

Selection: The process of choosing the best-fit individuals from the current population to be parents of the next generation.

Crossover (Recombination): A genetic operator combining two parent chromosomes to produce one or more offspring. This mimics biological reproduction and allows for the exchange of genetic material.

Mutation: A genetic operator that introduces random changes to individual genes in a chromosome. This helps maintain genetic diversity within the population and can prevent premature convergence on suboptimal solutions.

Generations: The iterative process of evolving the population over time. Each generation consists of selection, crossover, and mutation steps, producing a new population.

Survivors: Individuals or solutions selected to continue into the next generation of the evolutionary process.

Here's a breakdown of how genetic algorithms work:

The implementation of the genetic algorithm we used was specifically designed for SerDes optimization. It can be used in software (IBIS-AMI) models to train the channel conditioning coefficients (see example in [6]) or even implemented in the firmware of the controller. The genes are the register settings of the different equalizers in the transmit and receive data-path. The chromosomes of the initial population are randomly generated. The initial fitness of each chromosome is calculated using a pulse response-based FoM as discussed in Section 1.3.1. A percentage of the population with the best fitness function is selected.

After each generation, the algorithm adjusts the percentage of individuals selected for the next generation, and the percentage of new individuals who use combination methods and mutation methods. The percentage of survivors is initially 50% of the population and is reduced to 20% after several generations. The percentage of recombined survivors starts at 80% of the new members and is reduced to 20% of the new members after several generations.

The combination method randomly selects pairs of selected members. It creates new members by randomly selecting which genes are inherited from one or the other pair, or by randomly choosing a gene value between the pair's value. Survivors are mutated by a Gaussian distribution of random numbers with a sigma calculated from the variance of each gene's history.

The process terminates after fitness stops improving or after a specified maximum number of generations. There are only three parameters to control: the population size, the minimum amount of improvement of each generation, and the maximum number of generations.

GA algorithms usually terminate at a good fitness but not necessarily at the best. This solution is further tweaked using the Steepest Descent method (Climbing the Hill). The Steepest Descent method calculates the fitness function for the GA "best solution," and the fitness function for each gene is incremented by one and decremented by one. The best one of these cases is the new "best solution". This method is repeated until none of the incremented/decremented chromosomes give a better fitness, or a maximum stall generation is met.

3. Benchmark and optimization algorithm validation

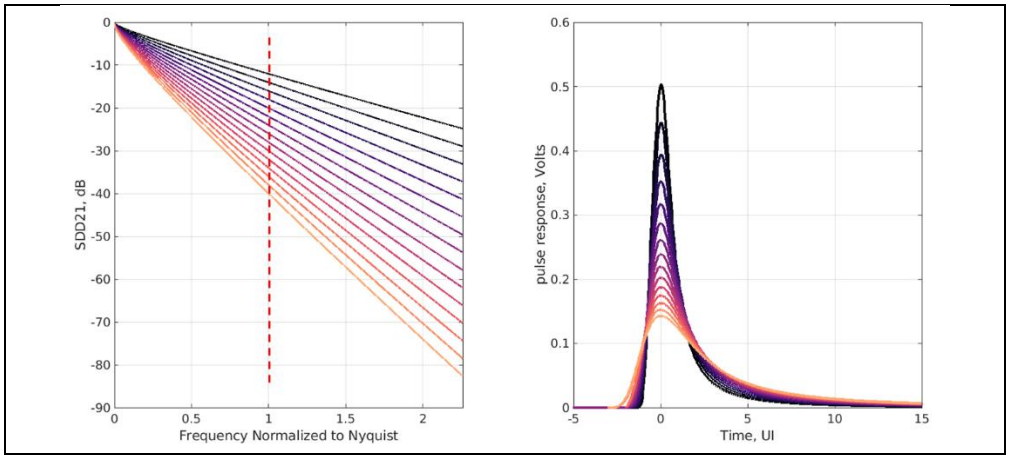
Before integrating the genetic algorithm described above inside the IBIS-AMI model, benchmarks were conducted using MATLAB testbenches to compare and validate the convergence and runtime benefit of the proposed algorithm against the original Brute Force method.

3.1. Setup

The testbench used for the comparison takes a channel impulse response as an input and returns the converged set of equalizer coefficients and the corresponding fitness metric (as described in section 1.3.1). The equalizer characteristics and fitness function used to drive the optimization are kept identical for each optimization method.

The testbenches were run over two sets of equalizers to check the robustness of the optimization algorithm under test, one representing a 100Gb/s DSP transceiver and the second for a 200Gb/s DSP transceiver. Table 1 summarizes the equalizer complexity in terms of the number of permutations. The adaptation algorithm and its capability to achieve or come close to the true "optimal" settings are validated over an idealized channel represented as a lossy transmission line (see annex 93A in [3]) where the Nyquist loss was swept from low reach (~12dB to extra-long reach 40dB). The transfer function of the transmission lines used are shown in Figure 9 in both frequency and time-domain.

Figure 9: Transmission line models



In addition to the convergence characteristics, the total optimization runtime was recorded for each case as well. All simulations were run on the same Linux computer with identical computing resources.

3.1.1. Search space

Since the benchmark is done against a brute force approach, which evaluates the fitness over the entire EQ combination range, it was decided to limit the full search space for the 100Gb/s and 200Gb/s equalizer as described in Table 1, by applying the following reductions:

- RX FFE floating taps turned off during adaptation (only adjusted after optimization is complete).
- TX FIR training is reduced to a single knob, and only the first pre-cursor is optimized:

$$\text{TX FIR } taps = [c_{m1} \;\; 1 - c_{m1}]$$

With this reduction, the new problem size utilized for the benchmark is highlighted in Table 2.

Table 2: Brute force permutations

100G Equalizer permutations	200G Equalizer permutations
199,680	8,028,160

3.1.2. Hybrid search

For the hybrid search, the main search loop is divided into smaller sections, using a coarse search for the FIR taps first, and then a finer search in the second iteration. The normalized FIR tap step size used is 0.08 for the coarse search and 0.02 for the fine search. This configuration yields an equalizer search range that is roughly reduced to a quarter of the brute force approach, and the corresponding total permutation size is shown in Table 3.

Table 3: Hybrid search permutations

100G Equalizer permutations	200G Equalizer permutations
59,904	2,408,448

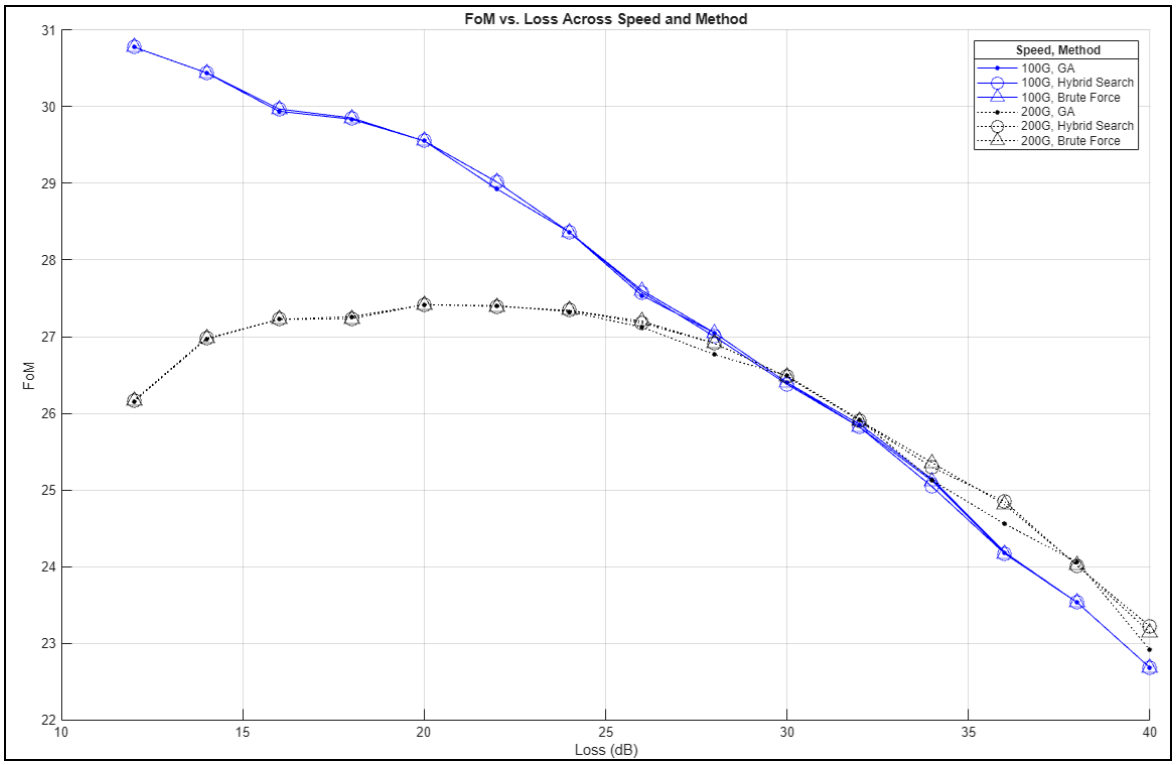
3.1.3. Genetic algorithms

The genetic algorithm used for the benchmark consists of populations of 60 equalizer combinations and a maximum stopping criterion of 20 generations. Before stopping, the algorithm allows for a maximum of 5 stall generations (generations that have not improved fitness over the previous population). The GA will be applied to the total available permutations described in Table 2.

3.2. Results

The summarized results of the benchmark are shown in Figure 10, where the Y-axis represents the converged value of the fitness metric used during the optimization, and the X-axis represents the channel bump-bump loss measured at the corresponding Nyquist Frequency for each 100Gb/s and 200Gb/s signaling and equalizer configuration. With the 100Gb/s equalizer, we can see that both the hybrid search and GA yield the same optimal equalizer prediction as the brute force search, represented by a nearly identical final FoM.

Figure 10: Optimization algorithm benchmark



The 200G equalizer optimization also shows good convergence with the GA optimization, with only a handful of points coming within a fraction of the optimal brute force fitness metric.

The next goal was to weigh each optimization method's simulation time penalty. The runtime analysis across different methods reveals distinct performance characteristics. GA demonstrates notable efficiency, maintaining relatively low run times with 100G and 200G equalizers. Since the number of fitness cost evaluations in the GA loop is bounded by the population size and a maximum number of generations allowed, the GA does not scale with the equalizer complexity, and therefore, the runtime remains relatively constant.

In contrast, Hybrid Search exhibits a substantial increase in run time as the number of permutations grows from ~60 thousand at 100G to ~2.5 million at 200G (Table 3), indicating that while it may be feasible with fewer configurations, it becomes significantly more resource-intensive and less scalable with larger permutation sets. On the other hand, Brute Force has the highest run time for 100G, making it the least efficient option for smaller permutation sets. At 200G, Brute Force takes approximately five days to complete, highlighting its impracticality for larger permutation sets due to its exhaustive nature and substantial computational demands.

Table 4: Optimization techniques runtime benchmark

Speed	Method	Mean Run Time		Fitness Function Evaluations
100	GA	3.4	min	900
100	Hybrid Search	45.3	min	59,904
100	Brute Force	2.1	hours	199,680
200	GA	5.5	min	900
200	Hybrid Search	1.4	days	2,408,448
200	Brute Force	4.9	days	8,028,160

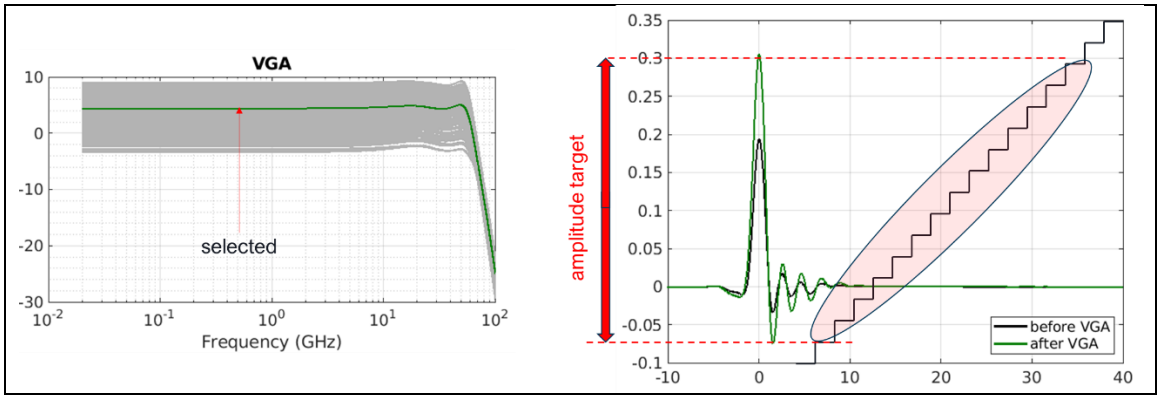
Factoring in that this benchmark was done over a reduced EQ search range, it becomes clear that the GA emerges as the only viable solution among the above three methods to support the optimization for the 200Gb/s equalizer. Although further work could have been pursued in tuning the hybrid search method for runtime, we recognized that ultimately, the hybrid search will always need adjustments according to the signaling rate and equalizer underlying data, and therefore, require a systematic understanding of the gradients for the optimization problem. This makes the Genetic Algorithm the preferable choice due to its efficiency, scalability, and ability to work with arbitrary knobs and constraints, particularly in scenarios requiring fast processing across varying levels of complexity.

4. Modeling Refinement

Following the clear benefit highlighted above with the usage of more efficient optimization techniques to speed up and take control of the model throughput, we decided to revisit the current fitness metric calculation and use some of the computational gain arising with the GA to close current modeling gaps, notably the simplified equalizer modeling employed during the adaptation phase and described in Section 1.3.1 and Figure 5.

Another gap and key aspect of the pulse-based fitness metric is that the amplitude tracking of the signal throughout the data path cannot be properly reflected when operating in non-linear regions of an amplifier stage. A good example is the Variable Gain Amplifier (VGA) register selection. Unlike a Nyquist filtering block like the transmit FIR, CTLE, or receive FFE, the purpose of the VGA is to provide a broadband gain and maximize the signal amplitude to fit within the limit (dynamic range) of the ADC. A graphical view of the VGA register selection process with the pulse-response based processing is shown in Figure 11, where the stairs depicted in the right plot represent the ADC quantization range and the red area the adaptation target.

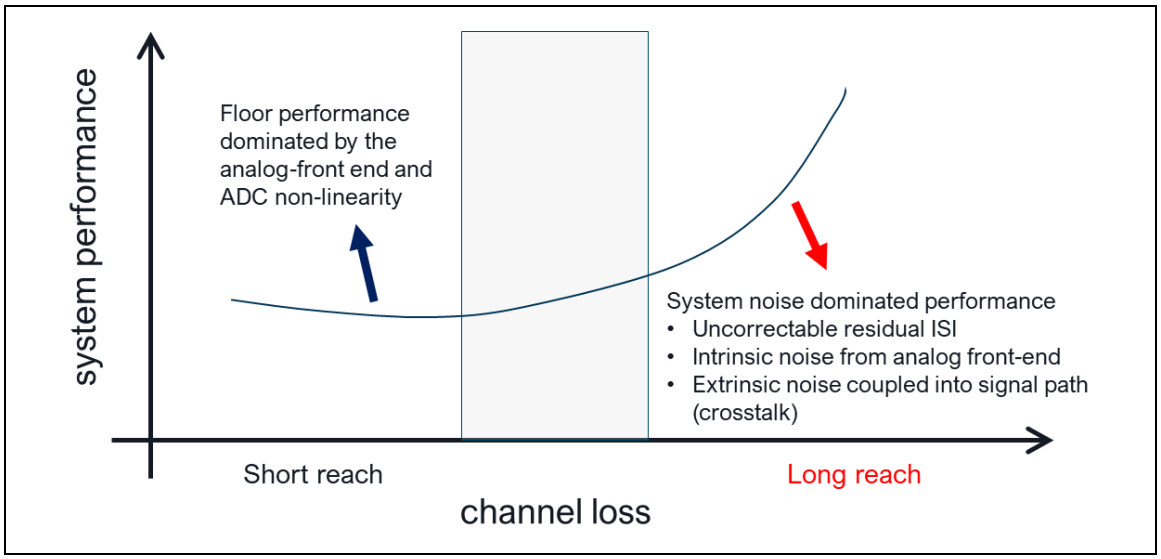
Figure 11: Pulse-based VGA adaptation



This selection is based on a fixed swing target rooted in the ADC design and not the actual non-linearity performance of the block. Not only the VGA but also other analog amplifiers are affected similarly. When a large signal propagates through these analog blocks, amplitude suffers linear compression, negatively impacting the overall performance of the time-domain signal, which is inherently not captured by the linear pulse-response driving the optimization.

To further illustrate the current limitations of the legacy fitness metrics employed in IBIS-AMI models thus far, an empirical plot in Figure 12 depicts a typical SerDes device performance curve as it would be measured with an electrical validation platform in a lab setup across varying reach. The two main parts of the channel reach range are exhibiting different trends, and a distinct mechanism limits the achievable system performance in both cases.

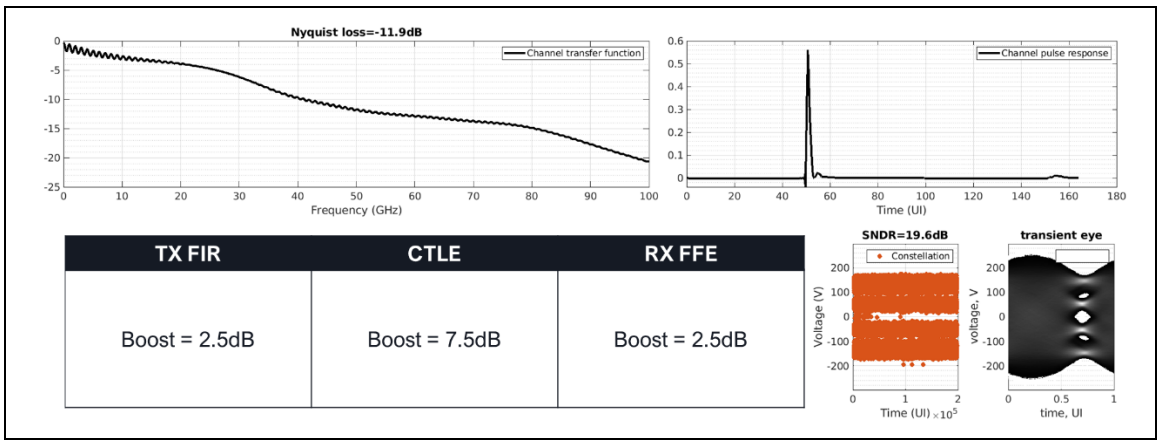
Figure 12 : BER vs loss trend long versus short reach



The focus and main discussion driving SerDes architecture development for a given SerDes are based on the right-hand side of the performance curve-where the challenge from an equalizer perspective is to provide adequate Nyquist boost with the lowest possible noise signature (intrinsic and amplification). This is a well-understood and implemented use case in the various simulation tools made available by the standardization electrical working group. The linear pulse response-based metrics, where the proper noise propagation is calculated, can very well predict the performance curve for these channel reach. On the other side of the loss spectrum, when channel reach is well below the transceiver's ISI correction capability, performance will typically run flat across a wide loss range, showing a performance floor, which is determined by the system's baseline noise, the ADC range (maximum achievable linear signal), and the amplifier's overall linearity across the entire signal path.

Effectively, with short-reach channels, because the only constraint the pulse-based fitness approach has is to concurrently maximize signal amplitude and minimize propagated noise, the optimization will typically navigate to an equalizer recipe that will drive the signal beyond the linear limits of the analog data path due to its incapacity at incorporating proper signal amplitude distortion using a single unit symbol response. Figure 13 graphically shows the above-mentioned issue using a 12 dB channel (from bump to bump) operated at 200Gb/s being impaired by non-linear compression because improper settings were chosen using the current pulse-based fitness. The output of the transient equalizer shows that both the orange scatter plot and the full transient eye-diagram exhibit un-even Signal-To-Noise-Ratio performance for the different symbol levels, resulting in "compressed" or close top and bottom part of the PAM4 eye, which degrades the overall performance.

Figure 13: Short channel adapted EQ using pulse-based fitness metric

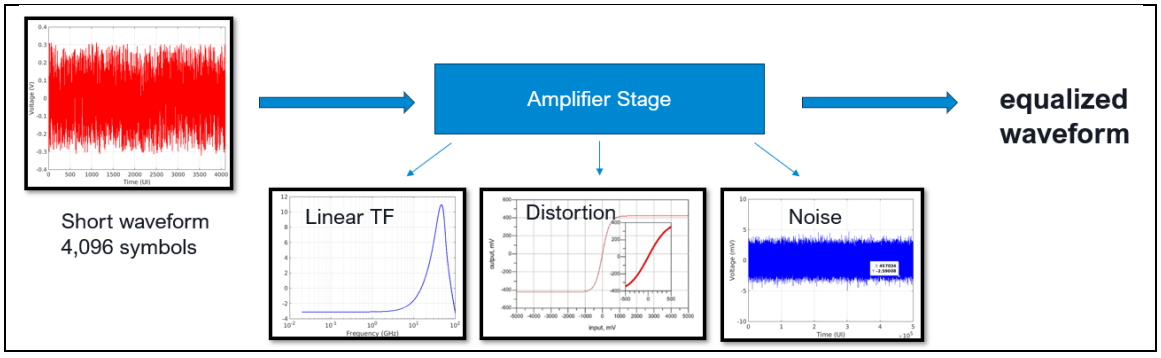


4.1. Proposed modeling improvement

To address the above-mentioned limitations yielding the sub-optimal adaptation results, we decided to revisit the entire fitness metric calculation, and a reasonable solution was to use the same equalizer modeling used in transient simulation. This means the adaptation loop must operate on an actual waveform rather than post-processing pulse responses. Some limitation of the waveform size to be used during the adaptation process is needed as we cannot test each equalizer combination on a full-size symbol stream (typically on the order of one hundred thousand to one million symbols) even with the reduced number of necessary fitness function evaluation in the genetic algorithm.

The proposed approach described in Figure 14 (representing a unit slice amplifier block) is to use a PRBS13 waveform block consisting of 4,096 symbols that are generated out of the channel fingerprint at the beginning of the adaptation loop and re-used during optimization to calculate the waveform propagation across the channel-conditioning chain.

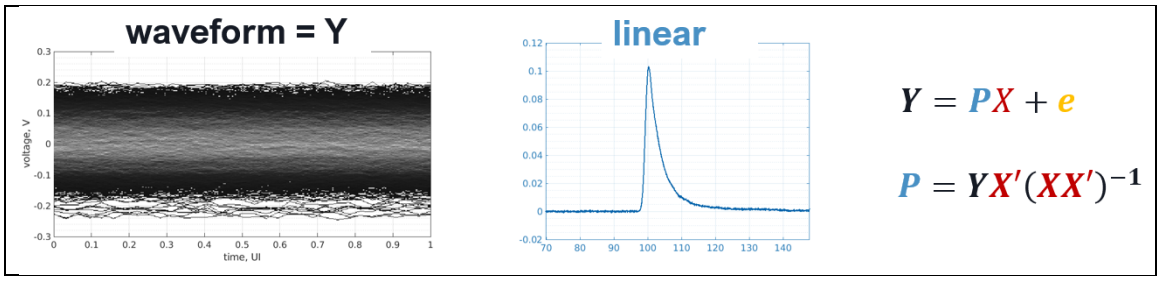
Figure 14: Single stage amplifier transient modeling



The multiple blocks making the entire analog amplifier portion of the receiver are cascaded up until the ADC, resulting in a partially equalized waveform, including

Nyquist filtering and non-linear compression, at the input of the DSP and providing the proper signal amplitude tracking. An additional step is needed at that point to retrieve the linear fitted pulse response, as the RX FFE needs to be configured to remove the remaining residual ISI in the signal. Given the partially equalized waveform "**Y**" at the ADC input, the least-square estimate of the pulse response (see Figure 15) is used where given the known PRBS13 symbol pattern "**X,**" the linear term "**P**" is retrieved and provides the necessary information to adjust the required FFE filter coefficient. The error term "**e**" proxies the additional non-linear impairments that were introduced with the new waveform-based modeling for each EQ block and contains the entire analog data path noise and distortion effects.

Figure 15: Lest-square estimate retrieval of partially equalized pulse response

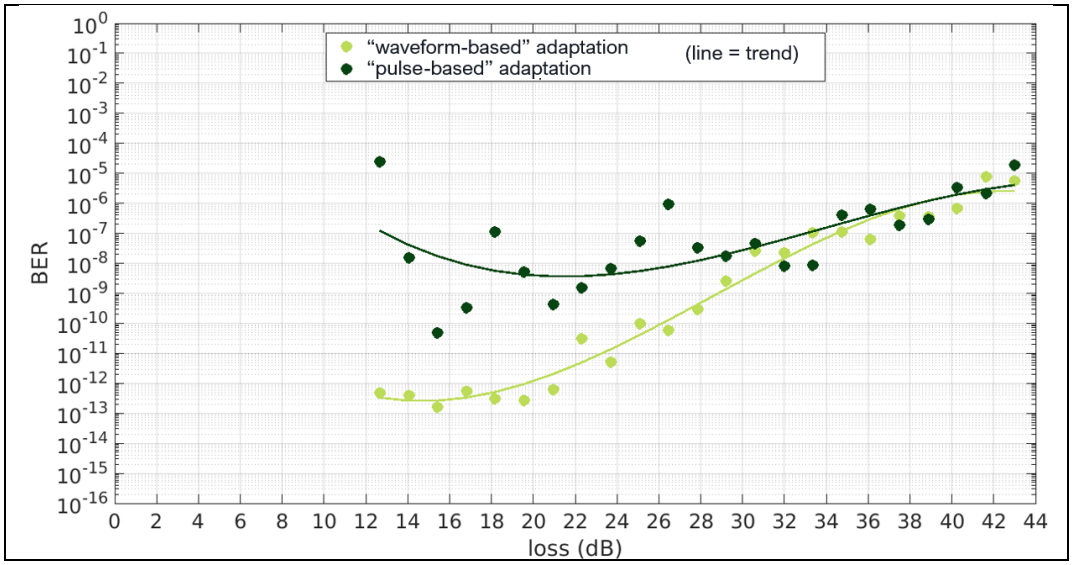


The filtering from the obtained linear FFE taps further processes the waveform until residual ISI is fully removed, at which point the corresponding constellation can be retrieved (sampled) and a fitness metric (Signal to Noise and Distortion Ratio, SNDR) extracted. This effectively drives the optimization algorithm with non-linearity constraint awareness.

4.2. Validation

Figure 14 compares the legacy pulse-based cost function with the proposed enhancement, using the same genetic algorithm that was used to derive the optimal equalizer solution shown in Figure 16. The channels used are identical to those used during the optimization benchmark study in Section 3 and represent a full range of potential real-life system implementations operating at 200Gb/s. We can clearly observe the pulse-driven solution holding well and providing an optimal EQ formula when the channel under test is dominated by Nyquist boost and bandwidth limitations (long reach). However, a strong inflection around 25B and the pulse-based approach clearly demonstrates mis-adaptation and high variation/instability as the channel reach gets shorter.

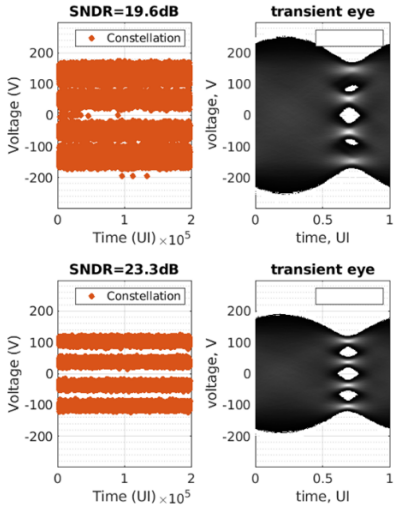
Figure 16: Adaptation adjustment



Further examining the adapted data for the same 12dB channels as shown previously in Figure 13, we can now see that the "waveform-based" fitness metrics depicted in the second row of Figure 17 properly avoided the non-optimal high-distortion settings and provided a more appropriate EQ combination in terms of equalized overall signal performance.

Figure 17: Adaptation improvements short reach

	TX FIR	CTLE	RX FFE
Pulse-based	Boost = 2.5dB	Boost = 7.5dB	Boost = 2.5dB
Waveform-based	Boost = 5.5dB	Boost = 2.5dB	Boost = 4.5dB



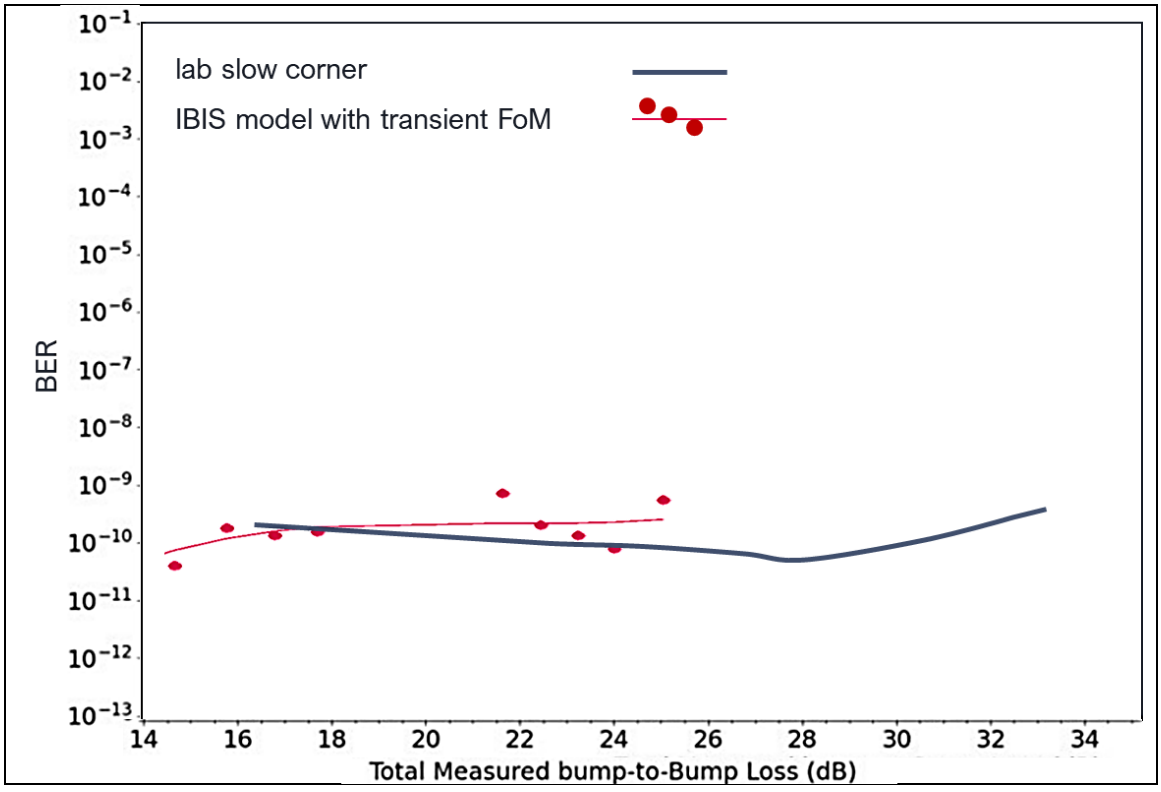
Additionally, a runtime benchmark comparing the cost to obtain the pulse-based fitness compared to the upgraded "transient-based" FoM suggests a mild ~10%-15% overhead of compute time exists for each cost-function evaluation. This does not materially offset

the optimization gain provided by the genetic algorithm and is deemed reasonable given the reliability benefits of the new adaptation results.

4.3. Lab comparison

Further validation of the proposed IBIS-AMI model's optimization cost-function changes were performed on a 100Gb/s SerDes platform. The goal was again to confirm the simulation mode's adaptation behavior against the physical device in the short reach region where the linear pulse adaptation was shown to be pessimistic because of sub-optimal adaptation results. Figure 18 compares a loopback capture of a 100Gb/s SerDes test chip over a validation platform with variable ISI stress and the results of the corresponding IBIS-AMI model incorporating the cost-function adjustment previously described. The model reproduces the flat behavior across the short-reach region where performance is dominated by the linear performance of the front-end.

Figure 18 – Low loss system performance benchmark



5. Conclusion

The goal of this paper was to revisit current methodologies adopted with our SerDes IBIS-AMI modeling for 100Gb/s generation of SerDes and identify enablers to solve key limitations and challenges in terms of accuracy and throughput to provide a robust path in reliable performance prediction using IBIS-AMI for 200Gb/s generation of copper links.

The adaptation of the SerDes model was seen as the key limiter in terms of model computational efficiency. A thorough investigation into alternative optimization techniques led to the successful benchmark and integration of a genetic-algorithm-based routine in the transceiver IBIS-AMI adaptation loop to provide a faster response time.

Given the computational benefit of the genetic algorithm optimization and the throughput predictability, further attention was paid to the overall accuracy of the adaptation routine linked to its fitness metric. The suggested path was to close the equalizer modeling gap between the "simplified" pulse-based equalizer routine that is traditionally used and the actual time-domain (non-linearity aware) equalizer that is a closer representation of the physical behavior of the SerDes. Integrating the changes into the adaptation fitness yielded prediction improvement and robustness, especially for shorter reach channels, with negligible impact on the overall model throughput. These results were further validated against the 100Gb/s platforms lab observed trends.

These steps will set a solid base for deploying and using robust IBIS-AMI models amongst system engineers and platform integrators when dealing with the AI-fueled incoming wave of copper link development.

References

[1] MathWorks, "ADC IBIS-AMI Model Based on COM," accessed February 17, 2022, https://www.mathworks.com/help/serdes/ug/adc-ibis-ami-model-based-on-com.html

[2] A. Tsychenko et Al, "DesignCon 2022 IBIS-AMI Modeling and Correlation Methodology for ADC-Based SerDes Beyond 100 Gb/s," DesignCon, 2022,

[3] IEEE Standards for Ethernet: IEEE Std 802.3-2022.

[4] IEEE P802.3df & IEEE P802.3dj Ethernet Task Force 2024, "Tools and Channels" accessed August, 2024, https://www.ieee802.org/3/dj/public/tools/index.html

[5] IBIS (I/O Buffer Information Specification) Version 7.2, January 2023.

[6] MathWorks, "ADC IBIS-AMI Model Based on COM with Genetic Algorithm Optimization" accessed October 9, 2024, https://www.mathworks.com/help/serdes/ug/adc-ibis-ami-model-based-com-with-genetic-algorithm.html