The Optimization of IBIS-AMI Model Parameters with Machine Learning Algorithms

Asia IBIS Summit
Shanghai, China
October 25, 2024

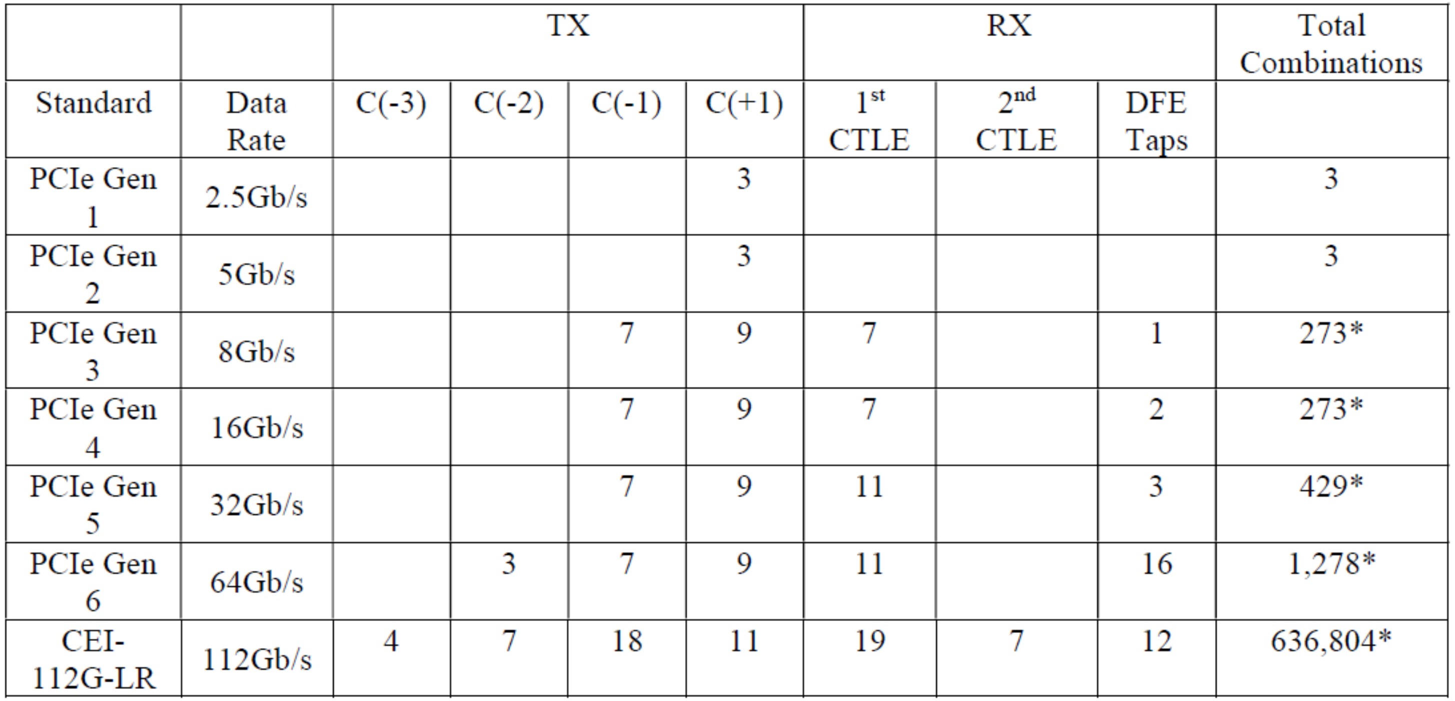
Jianping Kong, Principal Application Engineer
Multiphysics Analysis

Agenda

- Introduction
- Machine Learning Optimization
- IBIS-AMI Model Generation
- Simulation Setup
- Simulation Results
- Conclusions

Introduction

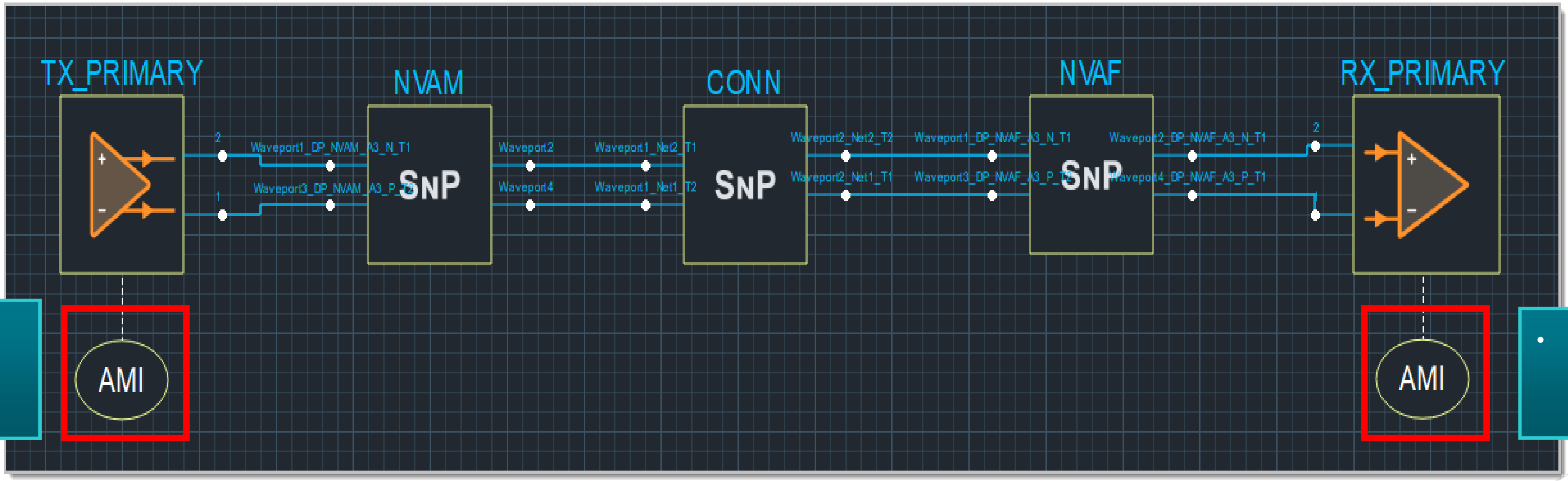
- Increased data rates
  - From 2003 to 2022, PCIe data rates have increased by 25x.
- Increased design complexity
  - Growth in the number of RX and TX equalization stages.
  - Increased number of TX cursor and ranges.
  - Increased number of RX CTLE stages and DFE taps.
- Increased channel validation time.
  - Traditional exhaustive simulations become impractical, even with parallel processing.



		TX				RX			Total Combinations
Standard	Data Rate	C(-3)	C(-2)	C(-1)	C(+1)	1st CTLE	2nd CTLE	DFE Taps	
PCIe Gen 1	2.5Gb/s				3				3
PCIe Gen 2	5Gb/s				3				3
PCIe Gen 3	8Gb/s			7	9	7		1	273*
PCIe Gen 4	16Gb/s			7	9	7		2	273*
PCIe Gen 5	32Gb/s			7	9	11		3	429*
PCIe Gen 6	64Gb/s		3	7	9	11		16	1,278*
CEI-112G-LR	112Gb/s	4	7	18	11	19	7	12	636,804*

Introduction (continued)

- A method is needed that will find the best equalization parameters for a given channel in a reasonable amount of time
- Optimization through machine learning will be integrated within the channel simulation environment


TX_PRIMARY
+
−
2
1
Waveport1_DP_NVAM_A3_N_T1
Waveport3_DP_NVAM_A3_P_T2
NVAM
SNP
Waveport2
Waveport4
Waveport1_Net2_T1
Waveport1_Net1_T2
CONN
SNP
Waveport2_Net2_T2
Waveport2_Net1_T1
Waveport1_DP_NVAF_A3_N_T1
Waveport3_DP_NVAF_A3_P_T2
NVAF
SNP
Waveport2_DP_NVAF_A3_N_T1
Waveport4_DP_NVAF_A3_P_T1
2
RX_PRIMARY
+
−
AMI
AMI

- Transmitter Settings

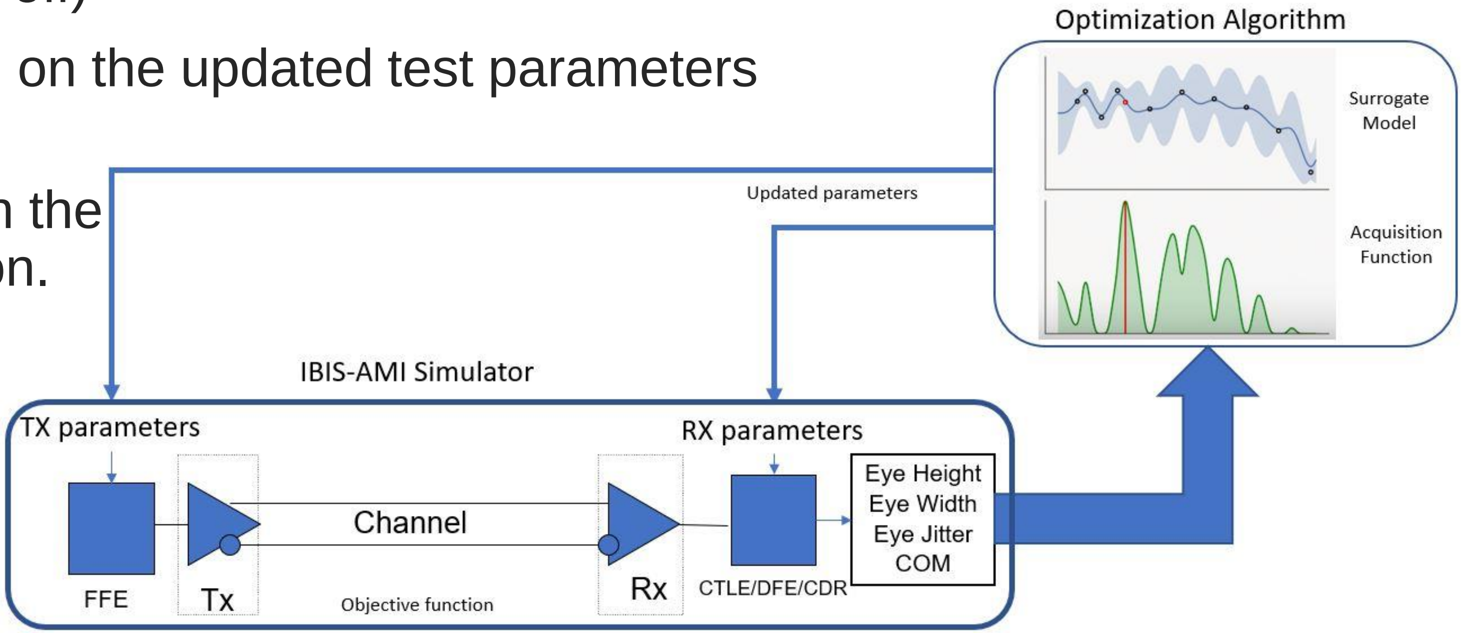
- Receiver Settings

Machine Learning for Optimization

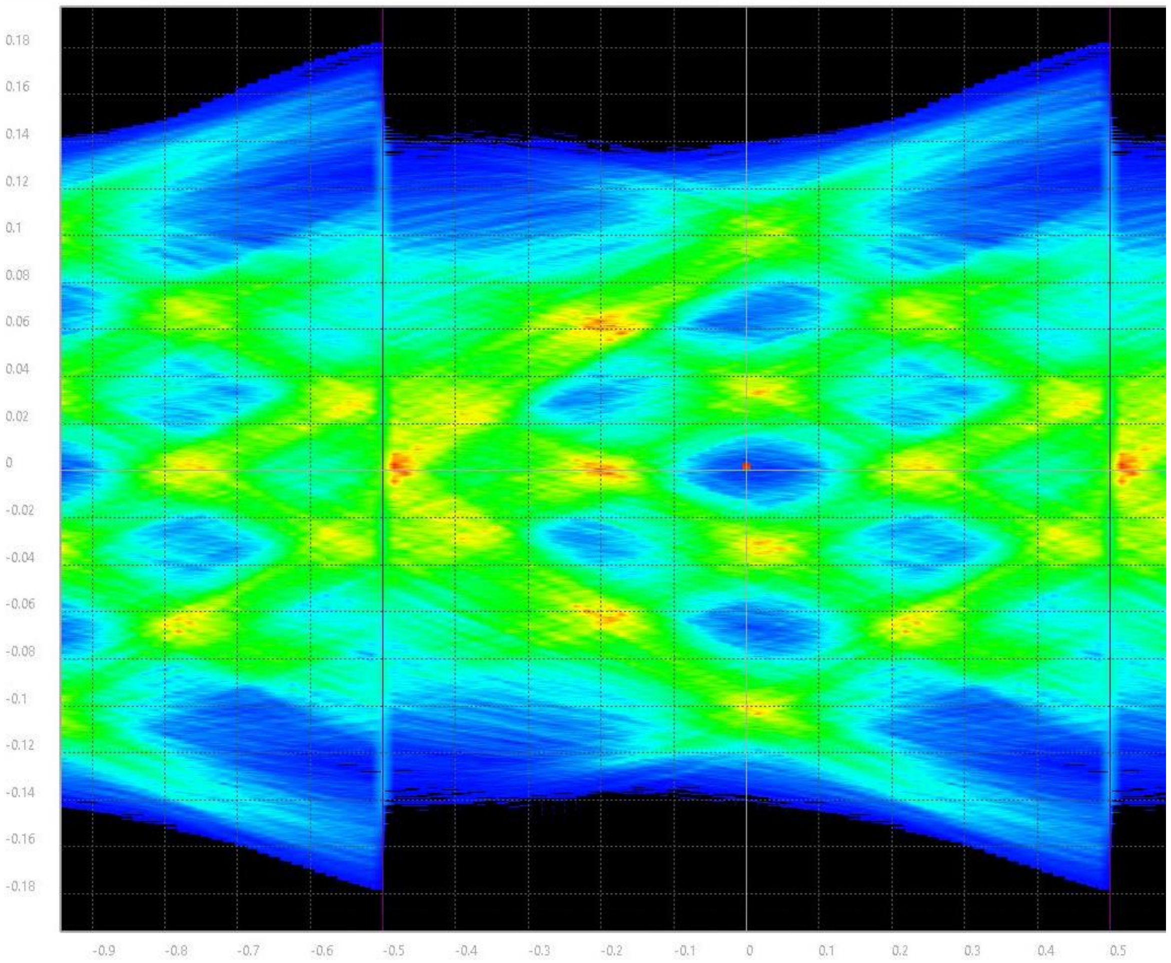
- Machine Learning:
  - The use and development of computer systems that can learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data.
- Optimization:
  - An act, process, or methodology of making something (such as a design, system, or decision) as fully perfect, functional, or effective as possible.
- Goal: Apply a machine learning optimization algorithm that will optimize the equalization parameters to maximize/minimize an output from the simulation results.
- These class of ML algorithms are useful when the objective function is computationally expensive or time-consuming to evaluate.

Machine Learning Optimization Process

1. An initial random sampling is collected by applying random parameter values to the objective functions.
2. The surrogate model is trained on the initial random sampling.
3. Create updated sets of test parameters.
   (Exploration vs. Exploitation trade off)
4. Evaluate the objective function based on the updated test parameters from the surrogate model.
5. Update the surrogate model based on the latest samples of the objective function.
6. Repeat steps 3-5 until
   a stopping criterion has been met.



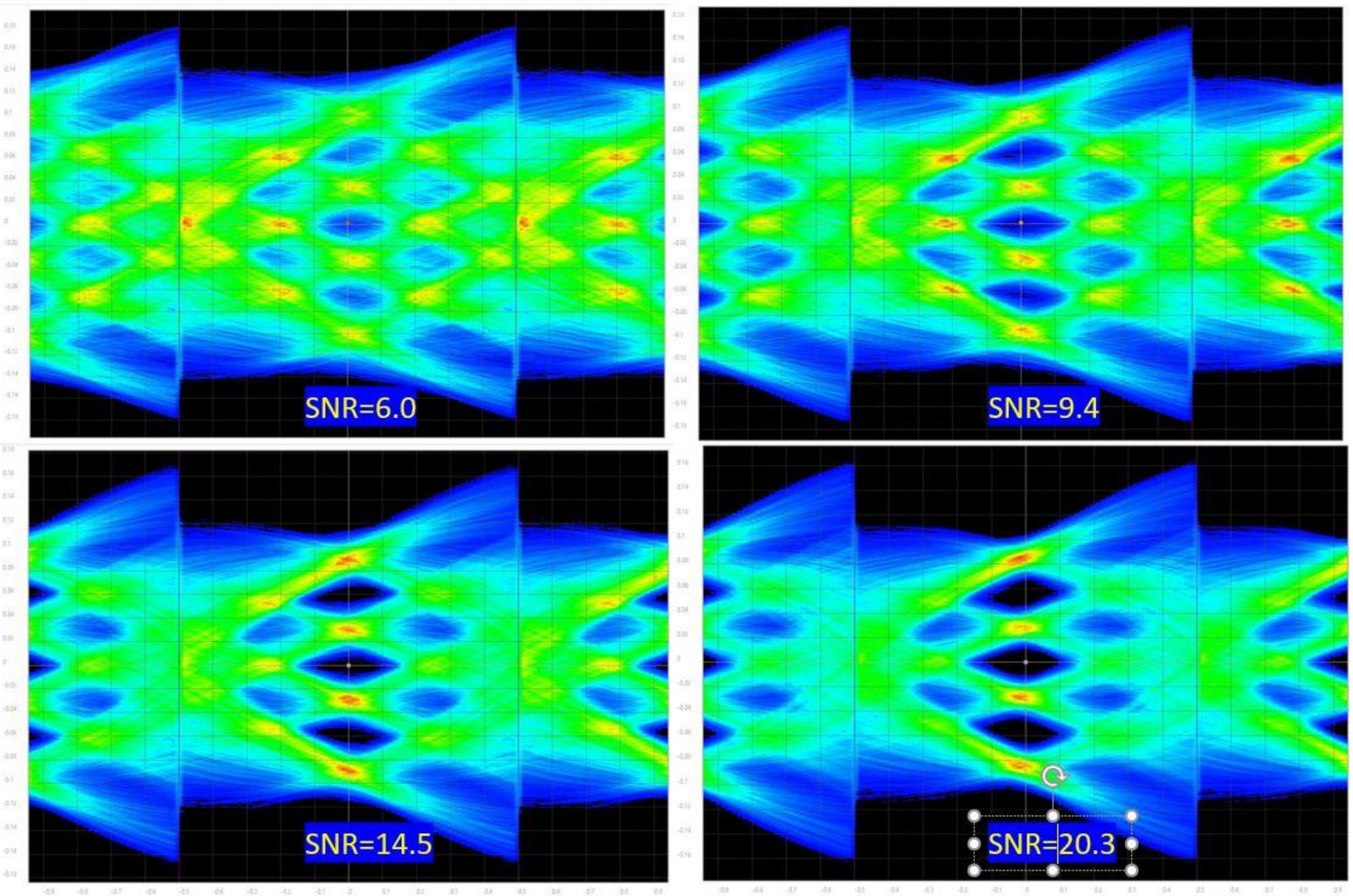
Optimization Needs



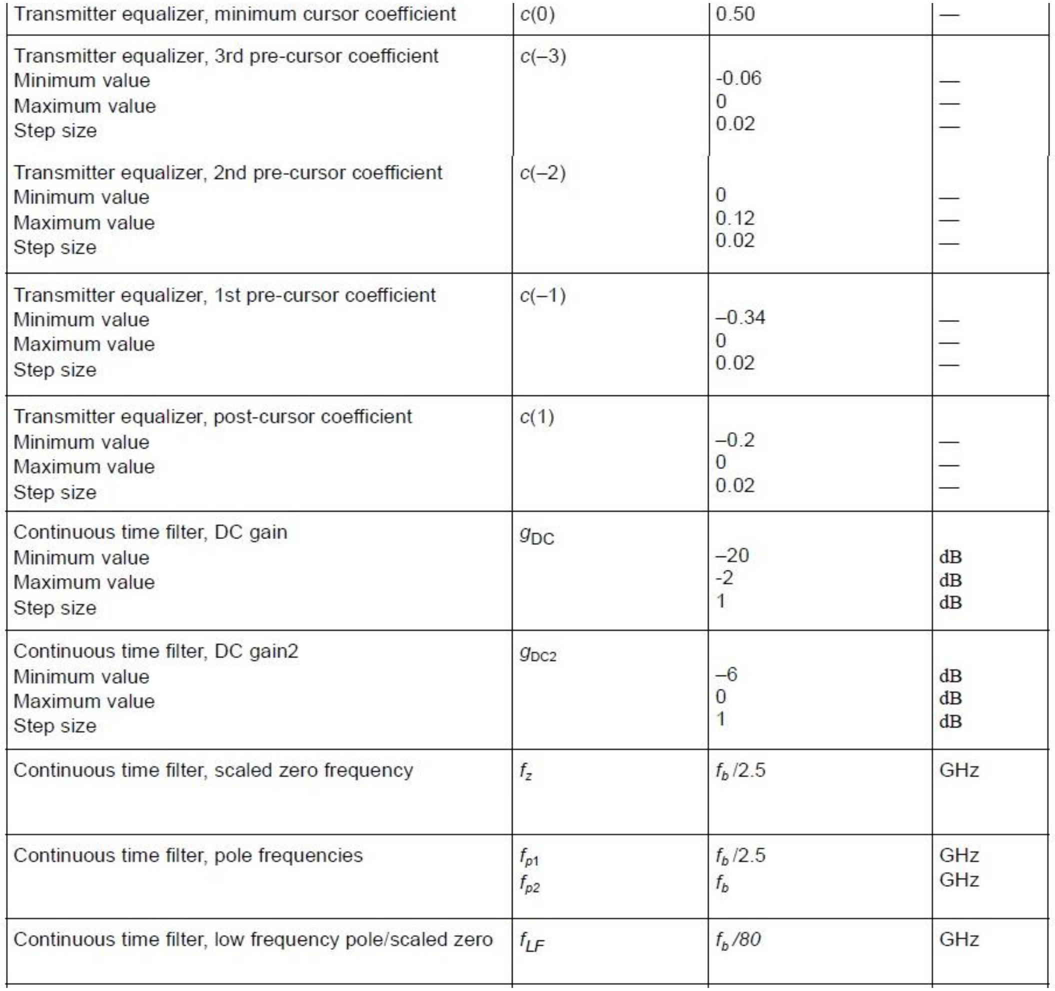
- Typical serial link simulations measurements:
  - Eye height.
  - Eye width.
  - Eye jitter
  - Channel Operating Margin (COM).
- These measurements (or some combination) can be used as an output that the ML optimization algorithm can optimize on.
- A disadvantage is that these measurements are zero if the eye is closed.
  - When the outputs are mostly zero, the algorithm has no gradient to optimize on.
  - This is not an issue if most/all the simulation results will likely have an open eye (i.e NRZ signaling).
  - The issue is that PAMx simulations eye are closed for most simulation results.
- **A measurement is needed that has a non-zero result when the simulated eye is closed.**

SNR as Objective Function

SNR examples



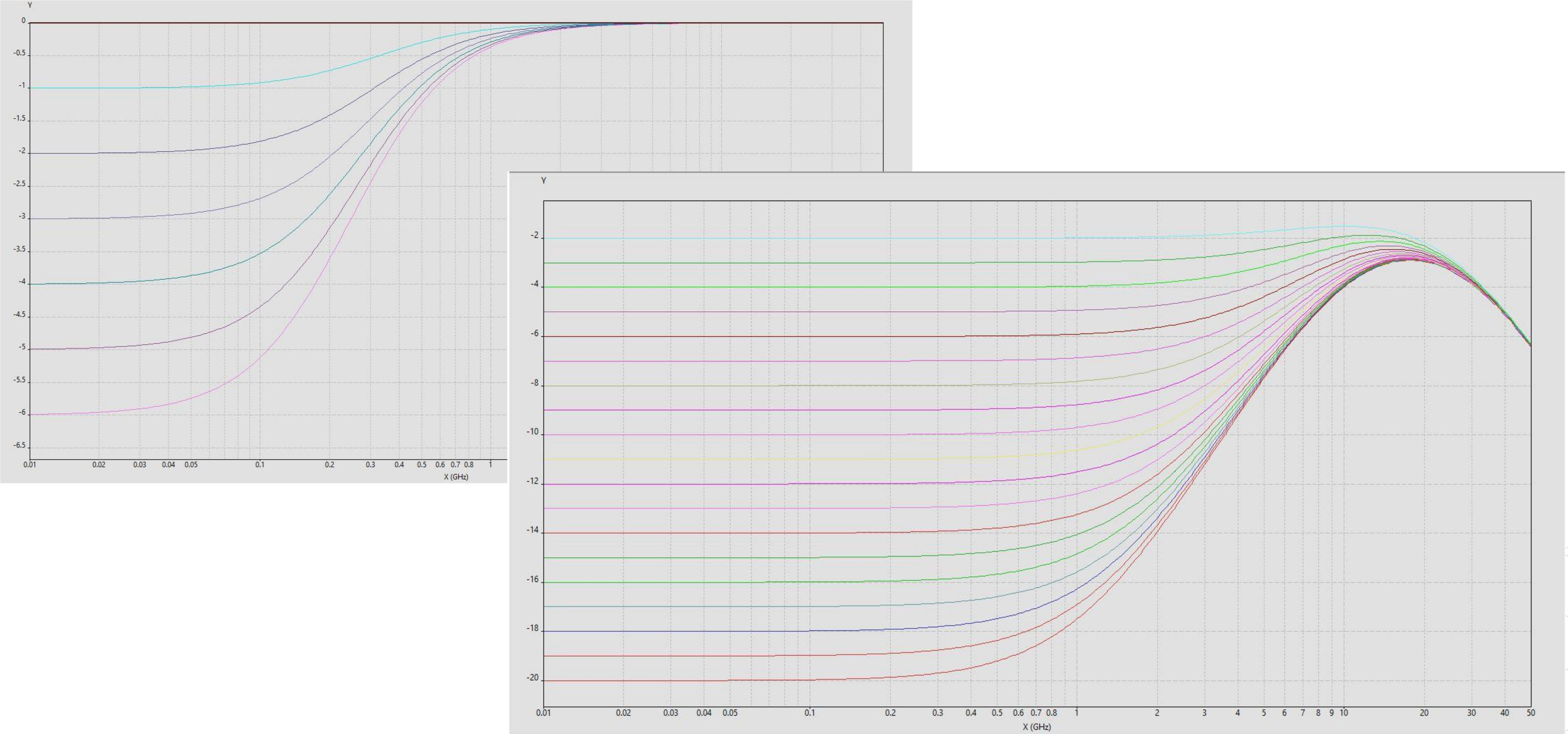
IBIS-AMI Model



Transmitter equalizer, minimum cursor coefficient	$c(0)$	0.50	—
Transmitter equalizer, 3rd pre-cursor coefficient Minimum value Maximum value Step size	$c(-3)$	-0.06 0 0.02	— — —
Transmitter equalizer, 2nd pre-cursor coefficient Minimum value Maximum value Step size	$c(-2)$	0 0.12 0.02	— — —
Transmitter equalizer, 1st pre-cursor coefficient Minimum value Maximum value Step size	$c(-1)$	–0.34 0 0.02	— — —
Transmitter equalizer, post-cursor coefficient Minimum value Maximum value Step size	$c(1)$	–0.2 0 0.02	— — —
Continuous time filter, DC gain Minimum value Maximum value Step size	g_{DC}	–20 -2 1	dB dB dB
Continuous time filter, DC gain2 Minimum value Maximum value Step size	g_{DC2}	–6 0 1	dB dB dB
Continuous time filter, scaled zero frequency	f_z	$f_b/2.5$	GHz
Continuous time filter, pole frequencies	f_{p1} f_{p2}	$f_b/2.5$ f_b	GHz GHz
Continuous time filter, low frequency pole/scaled zero	f_{LF}	$f_b/80$	GHz

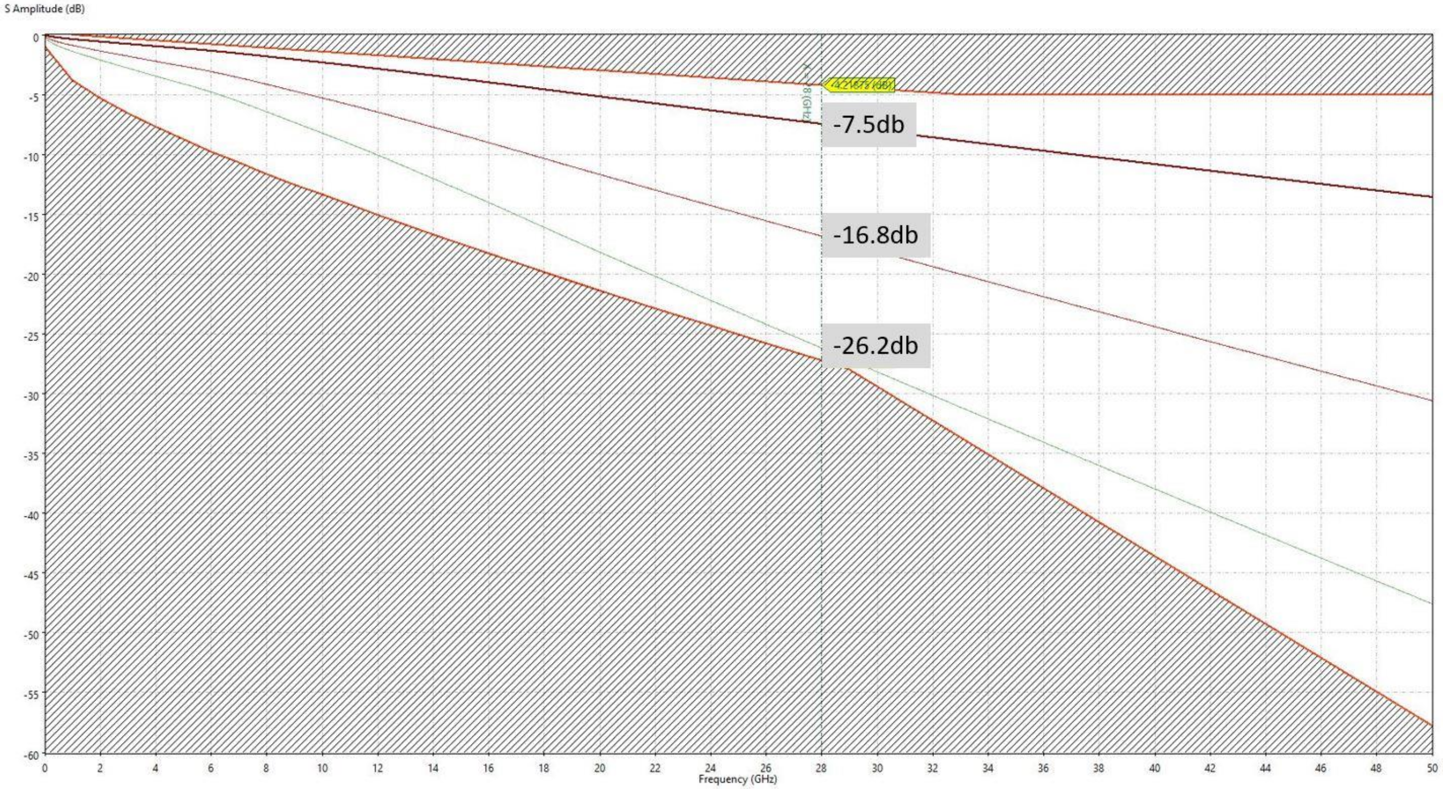
- An IBIS-AMI model was created based on the OIF-CEI 112G-LR reference model
- TX:
  - Three taps of pre cursor
  - One tap of post cursor
- RX:
  - Two CTLE stages
  - 12 tap DFE (adaptive)

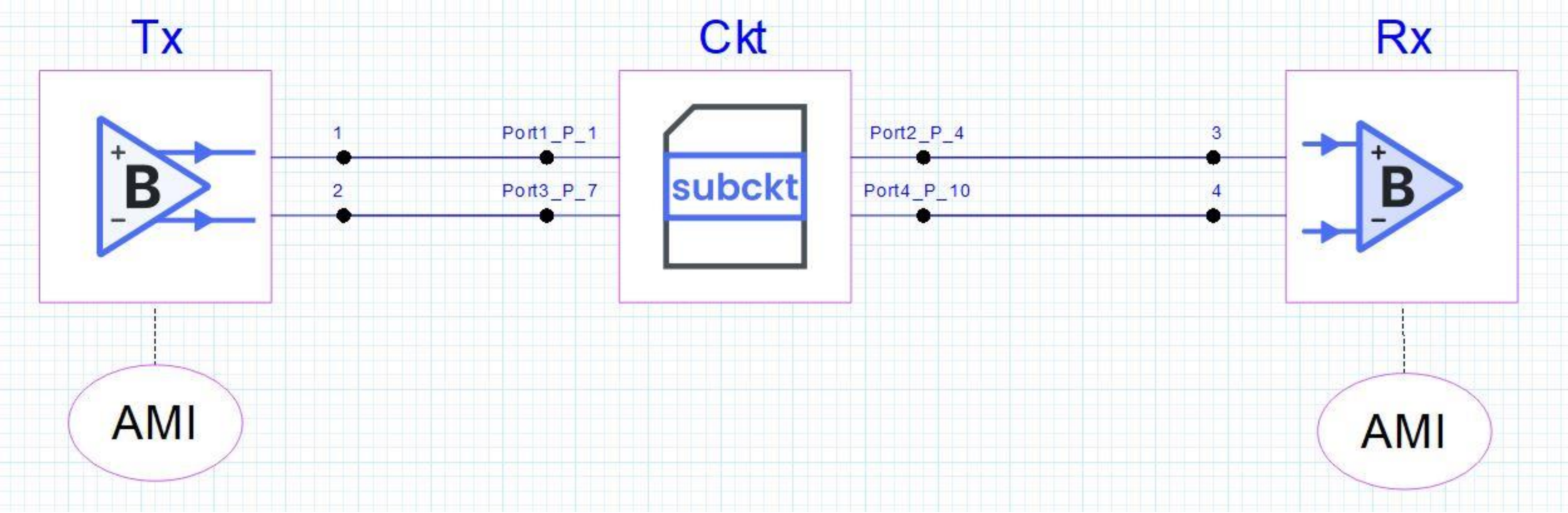
IBIS-AMI model, CTLE stages



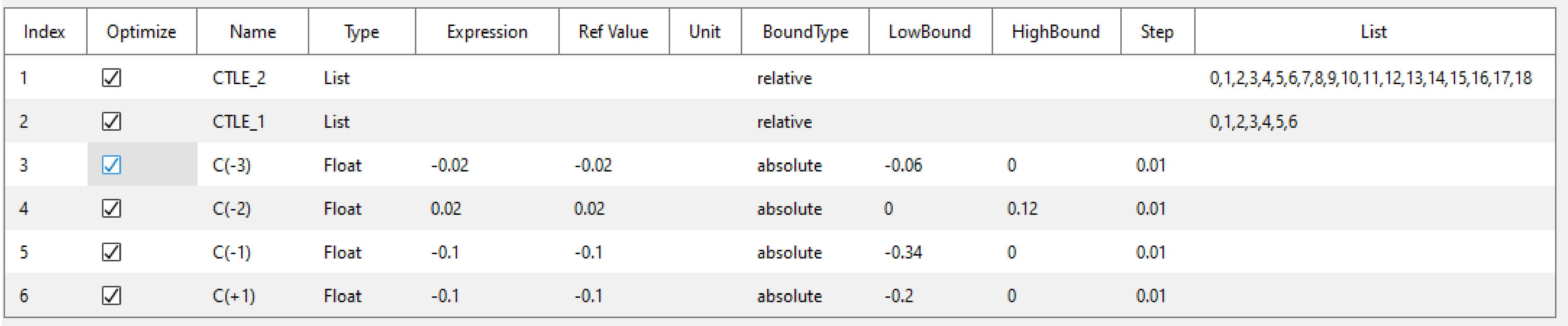
Simulation Setup

- Three different channels used:
  - Short (-7.5dB @ 28GHz)
  - Medium (-16.8dB @ 28GHz)
  - Long (-26.2dB @ 28GHz)





Equalization Parameter Setup



Index	Optimize	Name	Type	Expression	Ref Value	Unit	BoundType	LowBound	HighBound	Step	List
1	☑	CTLE_2	List				relative				0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18
2	☑	CTLE_1	List				relative				0,1,2,3,4,5,6
3	☑	C(-3)	Float	-0.02	-0.02		absolute	-0.06	0	0.01	
4	☑	C(-2)	Float	0.02	0.02		absolute	0	0.12	0.01	
5	☑	C(-1)	Float	-0.1	-0.1		absolute	-0.34	0	0.01	
6	☑	C(+1)	Float	-0.1	-0.1		absolute	-0.2	0	0.01	

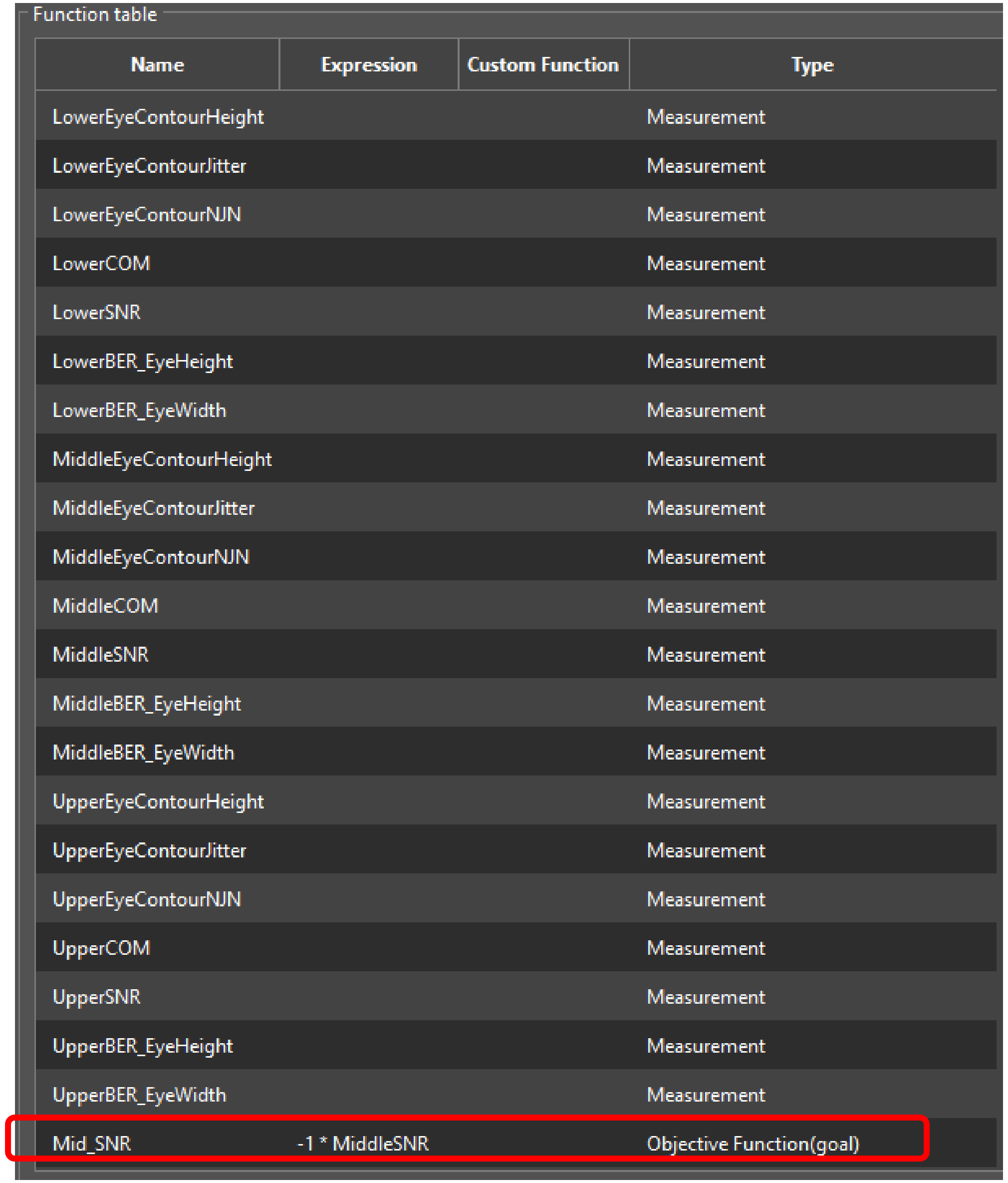
- TX FFE Taps:
  - Ranged float values
  - Step size = 0.01
- RX CTLE settings:
  - List of integer values
  - Peaking increases with increased setting

- The absolute sum of the TX coefficients must be less than/equal to 0.5
  - ABS(C(-3)) + ABS(C(-2))+ABS(C(-1))+ABS(C(+1))<=0.5
  - 57,785 valid TX coefficient combinations
  - For invalid TX coefficient combinations, all TX cursors values were set to 0.01
  - This provides low SNR results to the optimization algorithm, preventing it from adapting towards invalid equalization settings

- **Total Equalization Combinations:**

  7,685,405

Objective Function Setup



Function table

Name	Expression	Custom Function	Type
LowerEyeContourHeight			Measurement
LowerEyeContourJitter			Measurement
LowerEyeContourNJN			Measurement
LowerCOM			Measurement
LowerSNR			Measurement
LowerBER_EyeHeight			Measurement
LowerBER_EyeWidth			Measurement
MiddleEyeContourHeight			Measurement
MiddleEyeContourJitter			Measurement
MiddleEyeContourNJN			Measurement
MiddleCOM			Measurement
MiddleSNR			Measurement
MiddleBER_EyeHeight			Measurement
MiddleBER_EyeWidth			Measurement
UpperEyeContourHeight			Measurement
UpperEyeContourJitter			Measurement
UpperEyeContourNJN			Measurement
UpperCOM			Measurement
UpperSNR			Measurement
UpperBER_EyeHeight			Measurement
UpperBER_EyeWidth			Measurement
Mid_SNR	-1 * MiddleSNR		Objective Function(goal)

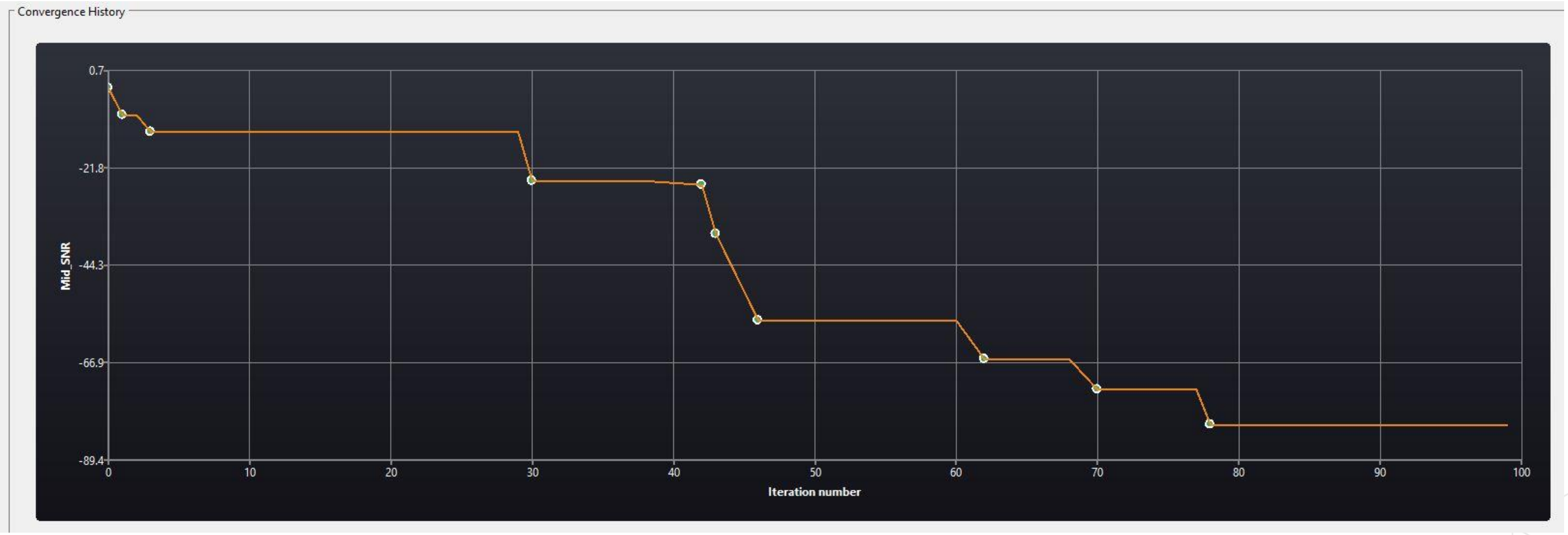
- List of available measurements from the serial link simulation results
- From these measurements, an objective function is created
- This could be a single measurement, or a mathematical combination of the available measurements
- The negative of the middle SNR eye was chosen to be the Objective Goal Function
  - The optimization algorithm is setup to find the minimal value of the objection function

Simulation Flow

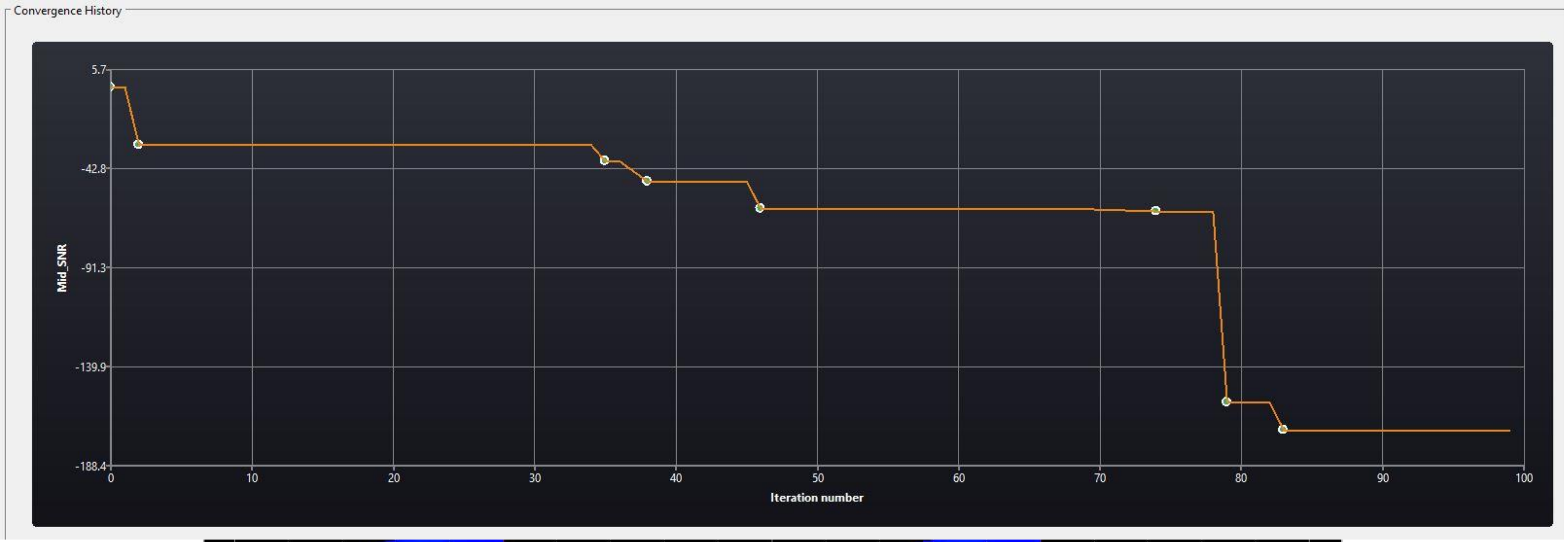
1. Based on the number of parameters to be optimized (6), 30 initial simulations were run with random parameter settings. The SNR of each simulation is recorded.
2. The surrogate model is trained on the SNR output from the initial set of simulations.
3. The surrogate model is then prepared for the next set of simulations.
   - Duplicate simulations are skipped to allow for more efficient simulations.
   - Exploration vs. Exploitation trade off
4. The results from the next set of simulation are used to update the surrogate model.
5. Steps 3 and 4 are repeated until no parameter sets are available, or if the total number of simulations has been reached (100 simulations).

Simulation Result Example

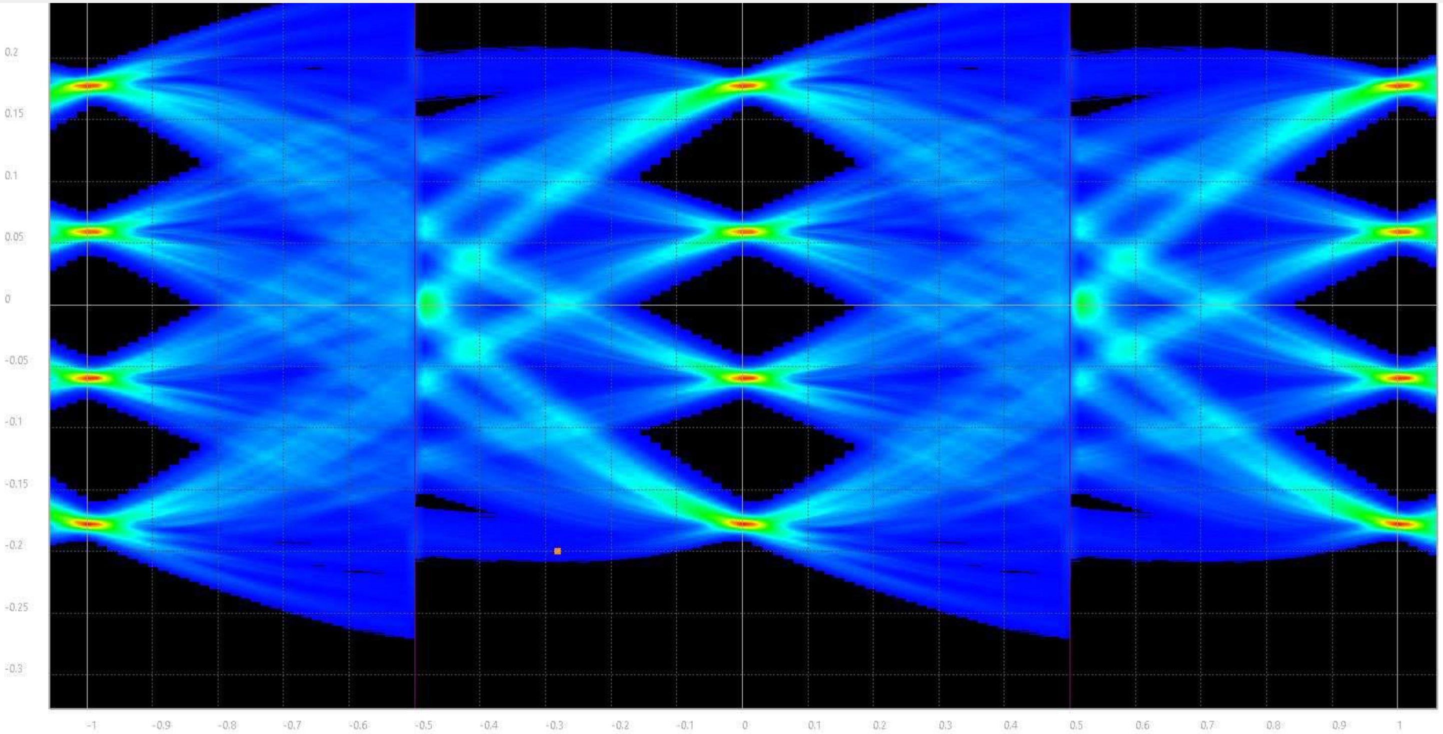
- A graph of the objective function (negative SNR) is plotted to show the convergence of the parameters. Lower negative values are better.
- Most optimization runs showed minimal improvement through the first 30 simulations.
- After this, the algorithm slowly trades off more exploitation simulations for fewer exploration simulations.



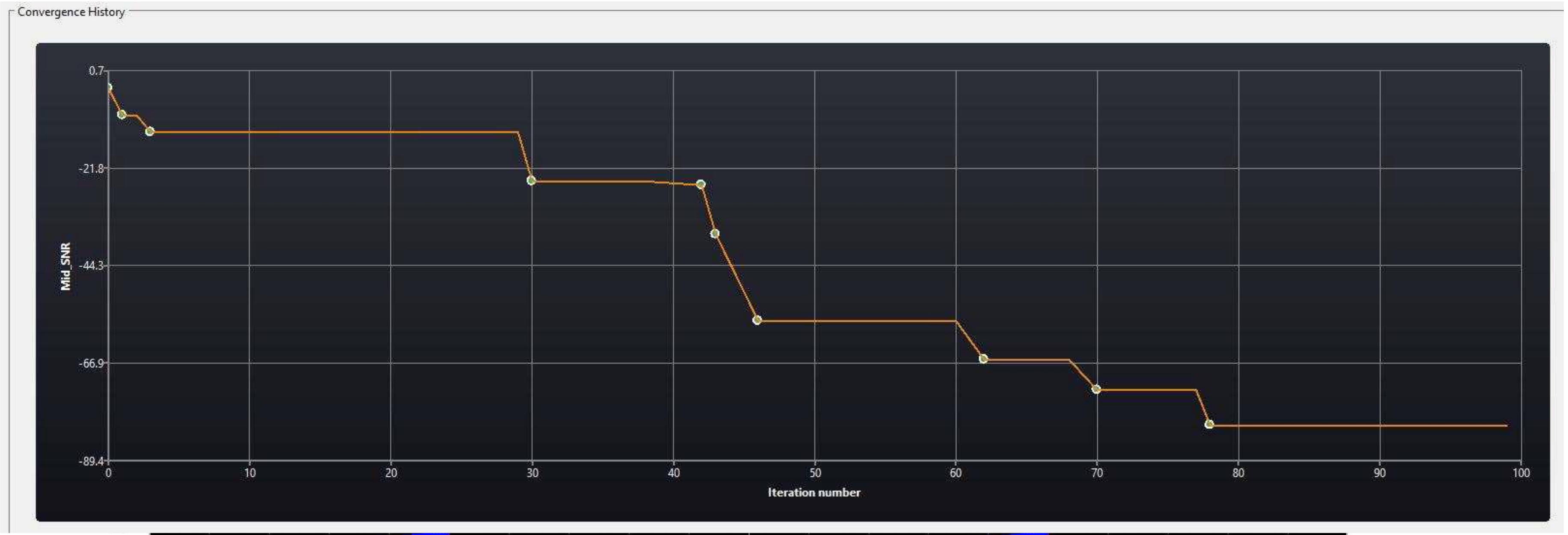
Results, Short



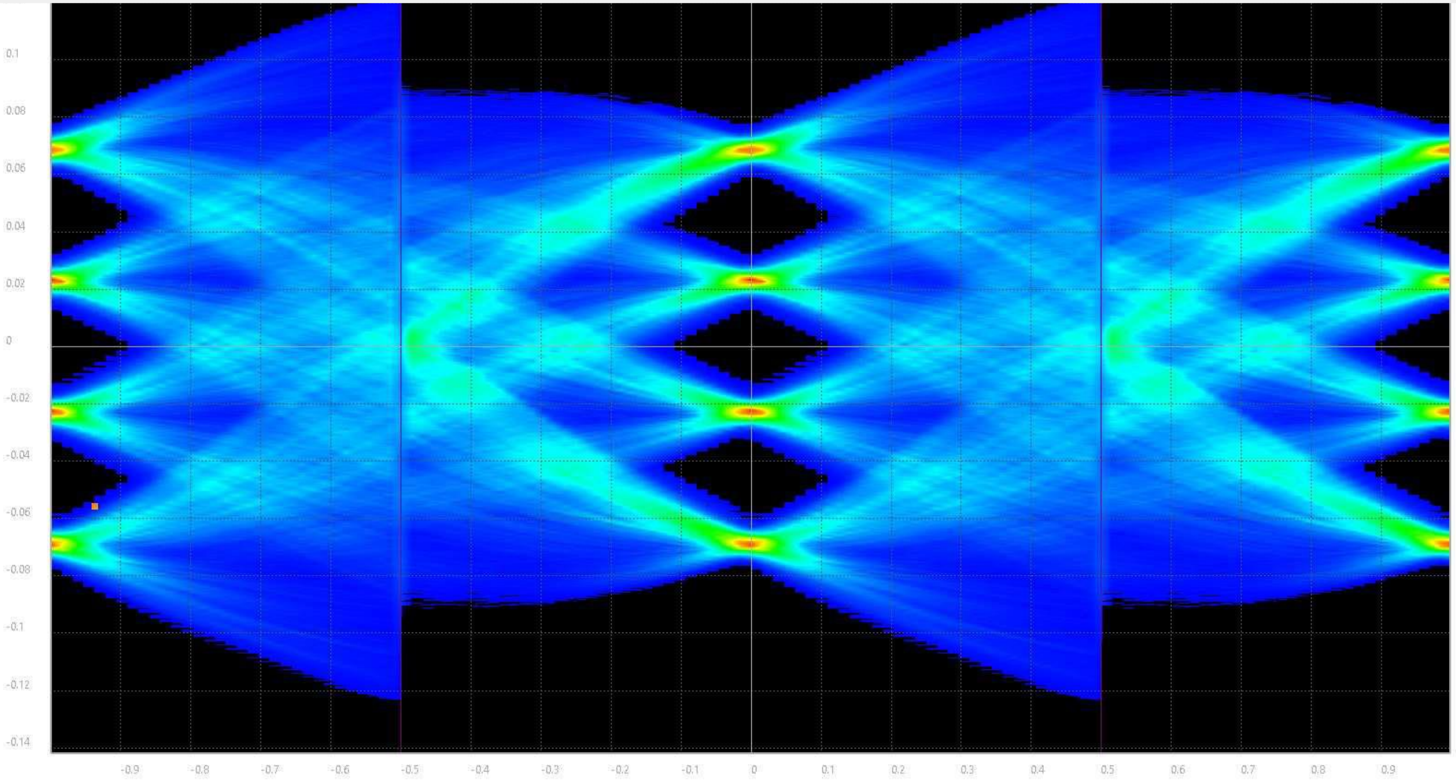
Mid_SNR = -171dB
CTLE_1 = 1
CTLE_2 =2
C(-3) =0.00
C(-2)=0.02
C(-1) =-0.13
C(+1) =-0.03



Results, Medium

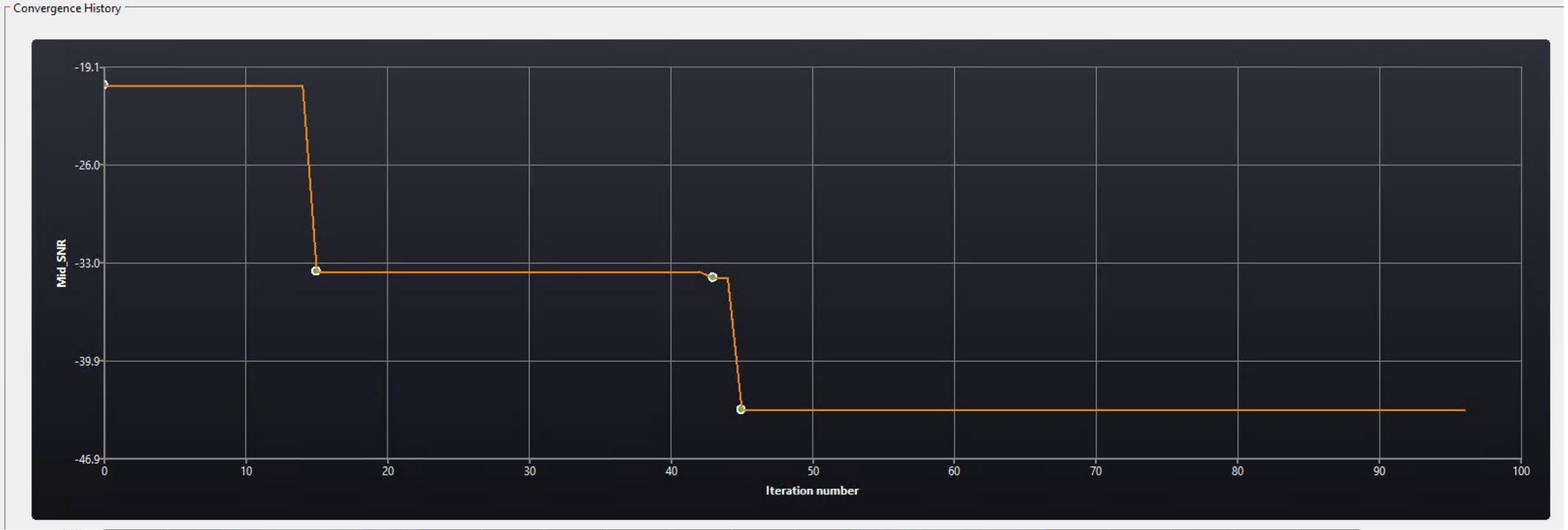


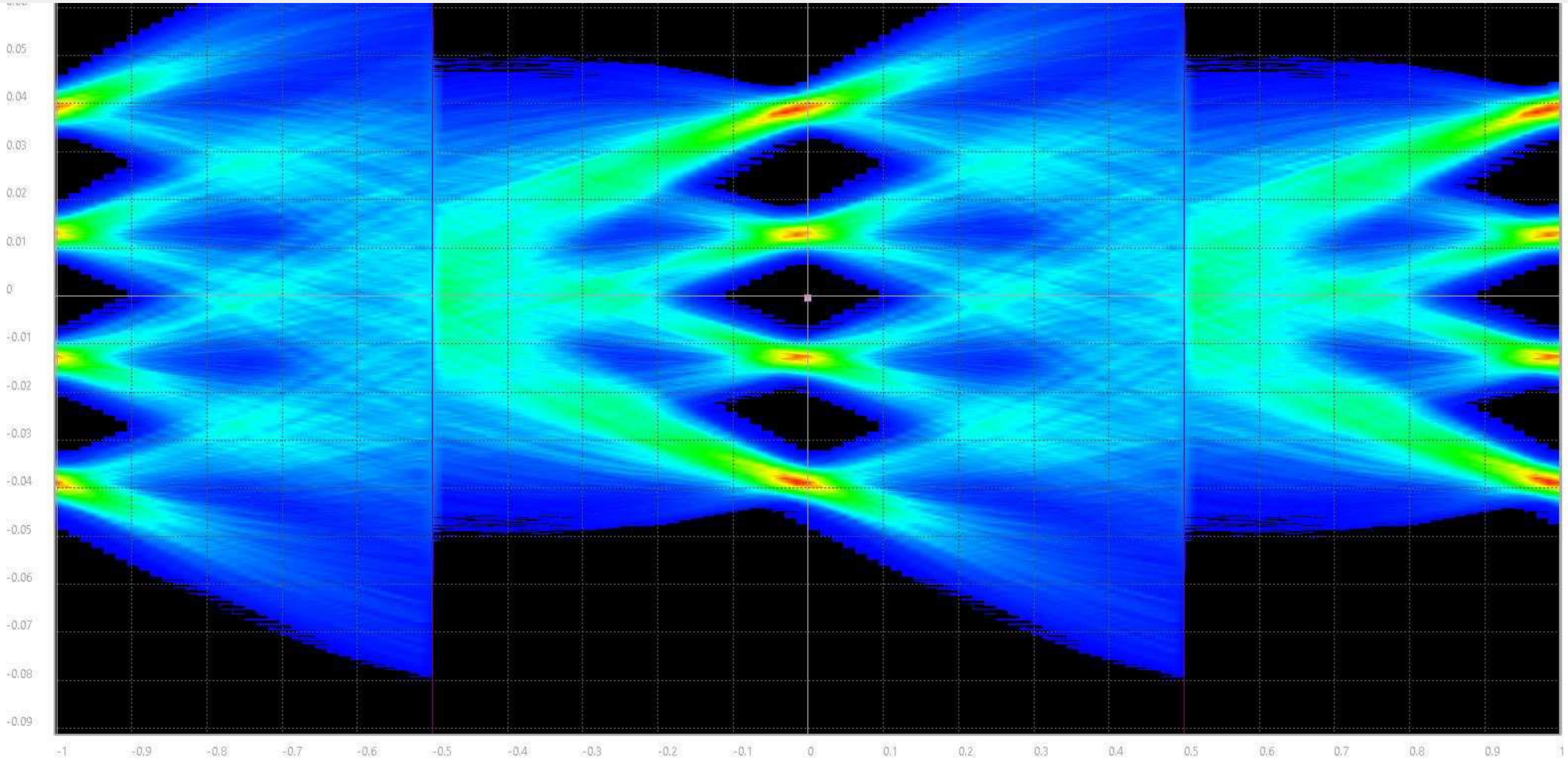
Mid_SNR = -81dB
CTLE_1 = 1
CTLE_2 =8
C(-3) =-0.02
C(-2)=0.06
C(-1) =-0.23
C(+1) =-0.03



Results, Long

Mid_SNR = -43dB
CTLE_1 = 6
CTLE_2 =0
C(-3) =-0.00
C(-2)=0.05
C(-1) =-0.25
C(+1) =-0.15





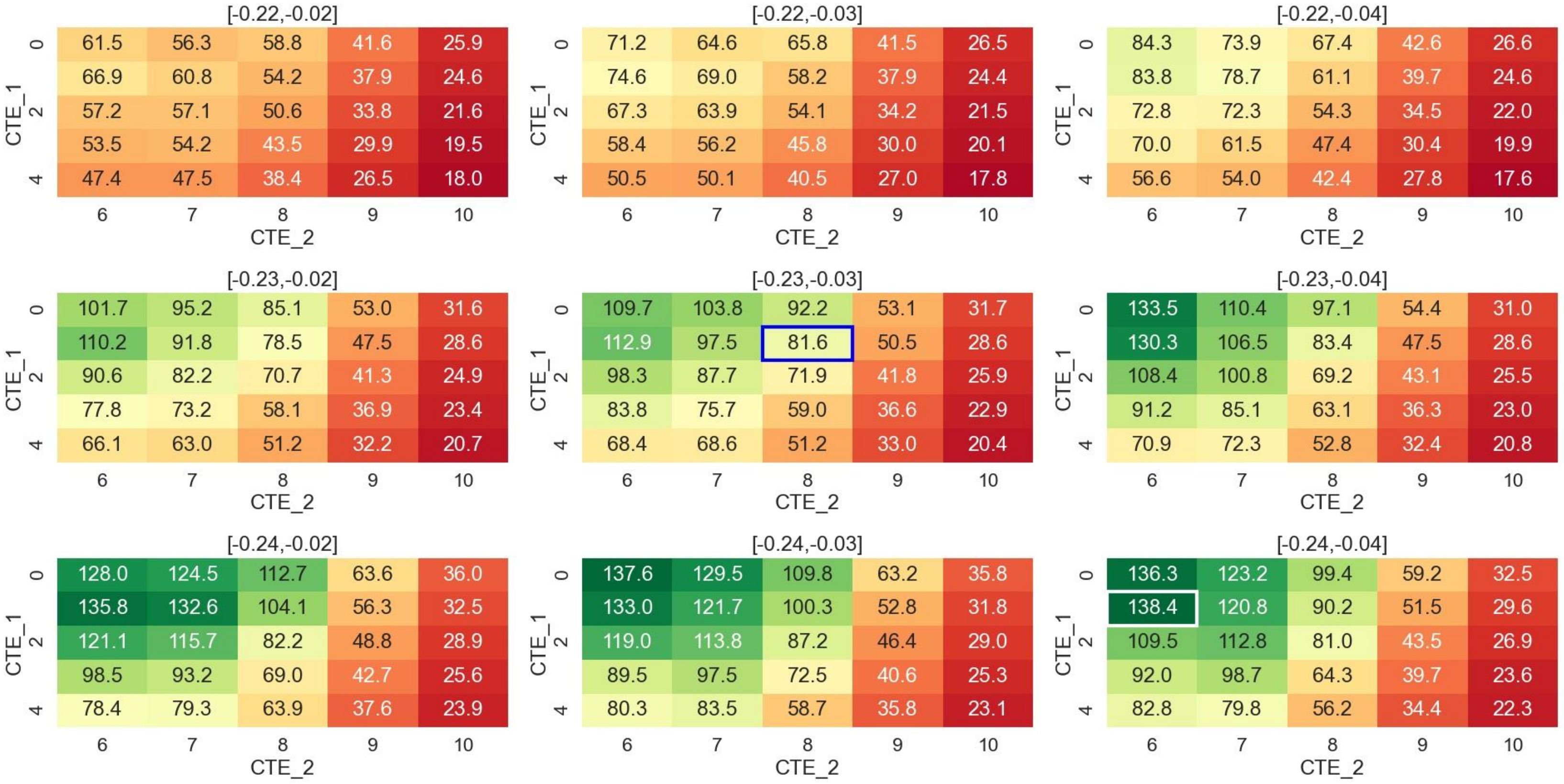
Validation of Optimization Results

- The optimized parameters need to be validated that they are the best, or close to the best.
- An exhaustive search to verify is not practical.
- Assuming that the optimized parameters are close to the best, a local sweep of the parameters could be used to validate the results.
- The parameters from the medium channel test case was local swept to verify the optimized parameters. Bold value are the values from optimization algorithm.
  - C(-1) (-0.22, **-0.23**, -0.24)
  - C(+1) (-0.02, **-0.03**, -0.04)
  - CTLE stage 1 (0,**1**,2,3,4)
  - CTLE stage 2 (6,7,**8**,9,10)
  - C(-2), C(-3) were not swept.

Medium Channel Validation

- Blue box is the optimized results.
- White box is the highest SNR from the local sweep results.
- The algorithm results were very close to the local sweep results.

Optimized	Local Sweep
SNR = 81dB	SNR = 138dB
CTLE_1 = 1	CTLE_1 = 1
CTLE_2 =8	CTLE_2 =6
C(-3) =-0.02	C(-3) =-0.02
C(-2)=0.06	C(-2)=0.06
C(-1) =-0.23	C(-1) =-0.24
C(+1) =-0.03	C(+1) =-0.04



[-0.22,-0.02]

CTE_1	6	7	8	9	10
0	61.5	56.3	58.8	41.6	25.9
	66.9	60.8	54.2	37.9	24.6
2	57.2	57.1	50.6	33.8	21.6
	53.5	54.2	43.5	29.9	19.5
4	47.4	47.5	38.4	26.5	18.0

CTE_2

[-0.22,-0.03]

CTE_1	6	7	8	9	10
0	71.2	64.6	65.8	41.5	26.5
	74.6	69.0	58.2	37.9	24.4
2	67.3	63.9	54.1	34.2	21.5
	58.4	56.2	45.8	30.0	20.1
4	50.5	50.1	40.5	27.0	17.8

CTE_2

[-0.22,-0.04]

CTE_1	6	7	8	9	10
0	84.3	73.9	67.4	42.6	26.6
	83.8	78.7	61.1	39.7	24.6
2	72.8	72.3	54.3	34.5	22.0
	70.0	61.5	47.4	30.4	19.9
4	56.6	54.0	42.4	27.8	17.6

CTE_2

[-0.23,-0.02]

CTE_1	6	7	8	9	10
0	101.7	95.2	85.1	53.0	31.6
	110.2	91.8	78.5	47.5	28.6
2	90.6	82.2	70.7	41.3	24.9
	77.8	73.2	58.1	36.9	23.4
4	66.1	63.0	51.2	32.2	20.7

CTE_2

[-0.23,-0.03]

CTE_1	6	7	8	9	10
0	109.7	103.8	92.2	53.1	31.7
	112.9	97.5	81.6	50.5	28.6
2	98.3	87.7	71.9	41.8	25.9
	83.8	75.7	59.0	36.6	22.9
4	68.4	68.6	51.2	33.0	20.4

CTE_2

[-0.23,-0.04]

CTE_1	6	7	8	9	10
0	133.5	110.4	97.1	54.4	31.0
	130.3	106.5	83.4	47.5	28.6
2	108.4	100.8	69.2	43.1	25.5
	91.2	85.1	63.1	36.3	23.0
4	70.9	72.3	52.8	32.4	20.8

CTE_2

[-0.24,-0.02]

CTE_1	6	7	8	9	10
0	128.0	124.5	112.7	63.6	36.0
	135.8	132.6	104.1	56.3	32.5
2	121.1	115.7	82.2	48.8	28.9
	98.5	93.2	69.0	42.7	25.6
4	78.4	79.3	63.9	37.6	23.9

CTE_2

[-0.24,-0.03]

CTE_1	6	7	8	9	10
0	137.6	129.5	109.8	63.2	35.8
	133.0	121.7	100.3	52.8	31.8
2	119.0	113.8	87.2	46.4	29.0
	89.5	97.5	72.5	40.6	25.3
4	80.3	83.5	58.7	35.8	23.1

CTE_2

[-0.24,-0.04]

CTE_1	6	7	8	9	10
0	136.3	123.2	99.4	59.2	32.5
	138.4	120.8	90.2	51.5	29.6
2	109.5	112.8	81.0	43.5	26.9
	92.0	98.7	64.3	39.7	23.6
4	82.8	79.8	56.2	34.4	22.3

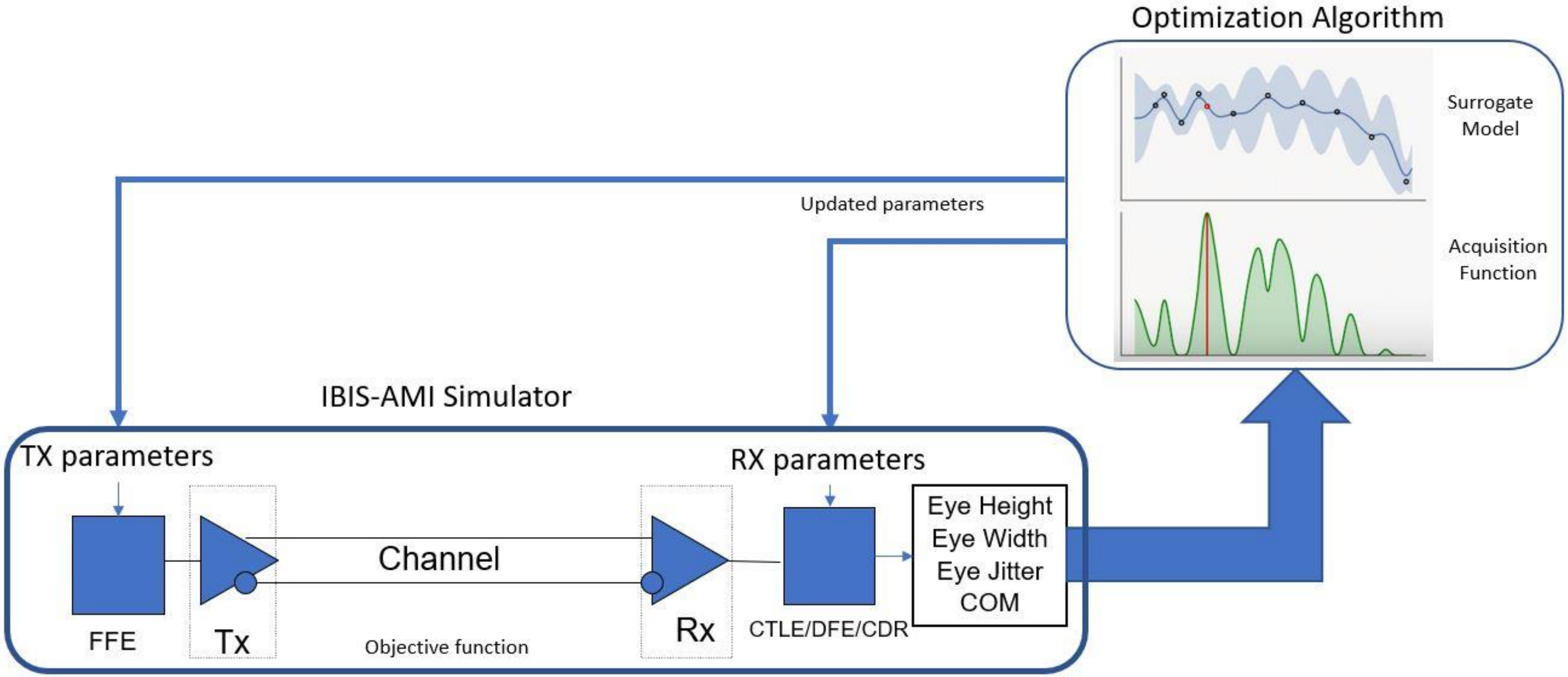
CTE_2

Conclusions

- A machine learning algorithm was applied to the optimization of AMI parameters in a serial link simulation.
- The results show that the algorithm was able to find good results for three different channels, indicating the robustness of the algorithm.
- This method was able to find a good set of parameters in fewer simulations than if an exhaustive method had been deployed, saving the use of limited compute resources.
- In most test cases, it was found that only 100 simulations were needed to find the best set of parameters. Compare this to the over 7 million simulations for an exhaustive search.

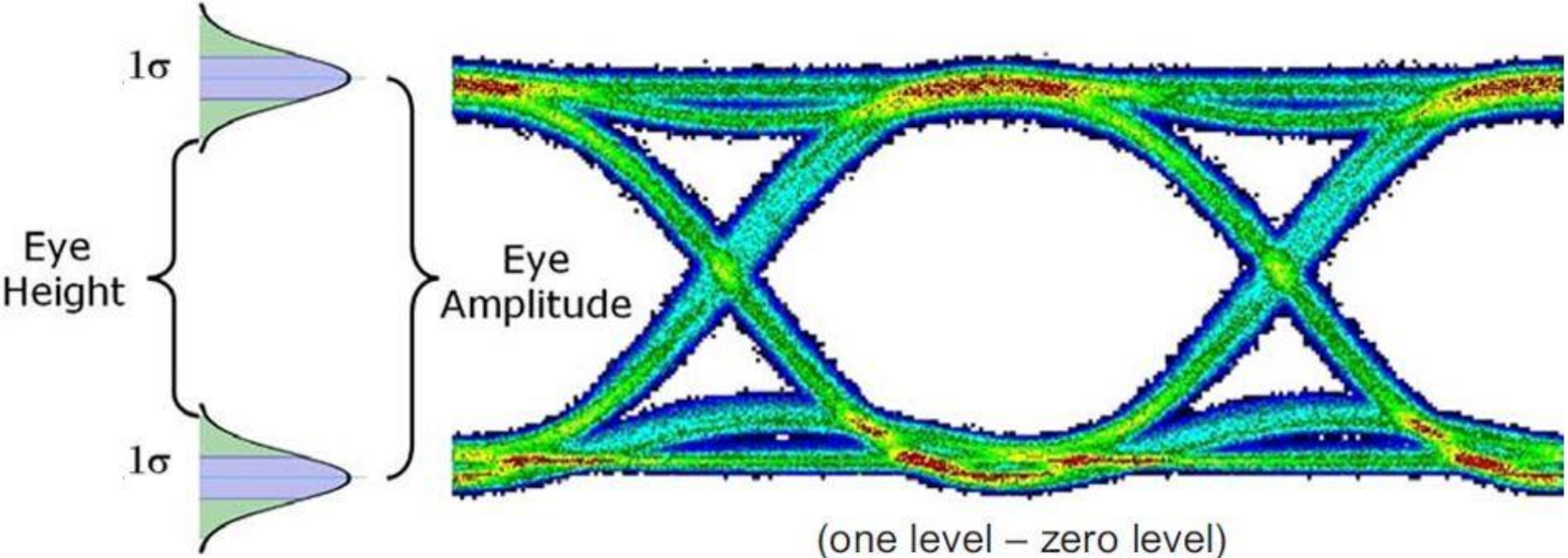
More Information

- Jared James (jjames@cadence.com)
- Ambrish Varma(ambrishv@cadence.com)



cādence®

Signal-to-Noise Ratio (SNR) Measurement



$$\text{SNR} = \frac{(\text{one level} - \text{zero level})}{(1\sigma\,[\text{one level}] + 1\sigma\,[\text{zero level}])}$$