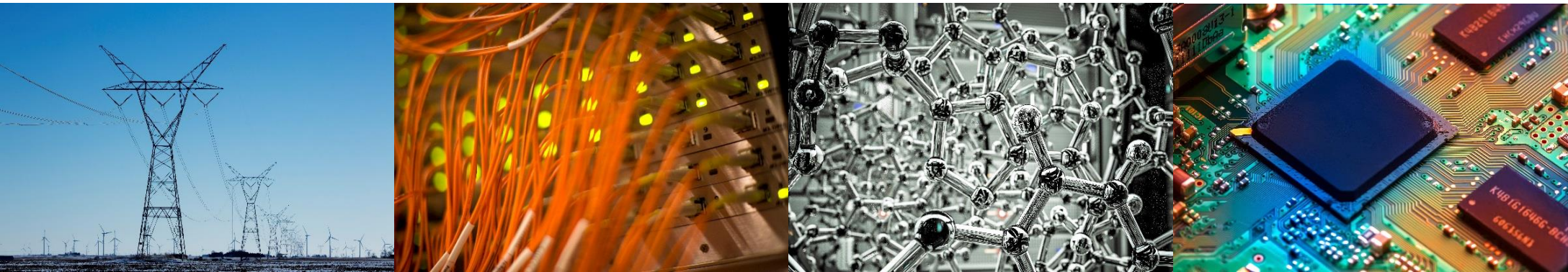


A PAM-4 behavioral model using Laguerre-Volterra feed forward neural network and its implementation in IBIS-AMI

Xinying Wang, Thong Nguyen, José E Schutt-Ainé



European Virtual IBIS Summit at Virtual SPI2021

May12, 2021

I ILLINOIS

Electrical & Computer Engineering

COLLEGE OF ENGINEERING

 UNIVERSITÄT
SIEGEN

SPI 2021

25th IEEE Workshop On Signal And Power Integrity

May 10-12 2021, Virtual Online Conference

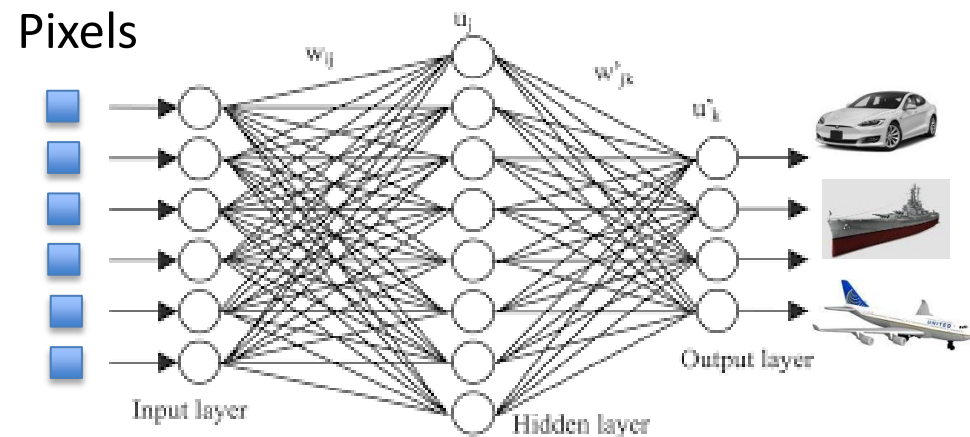
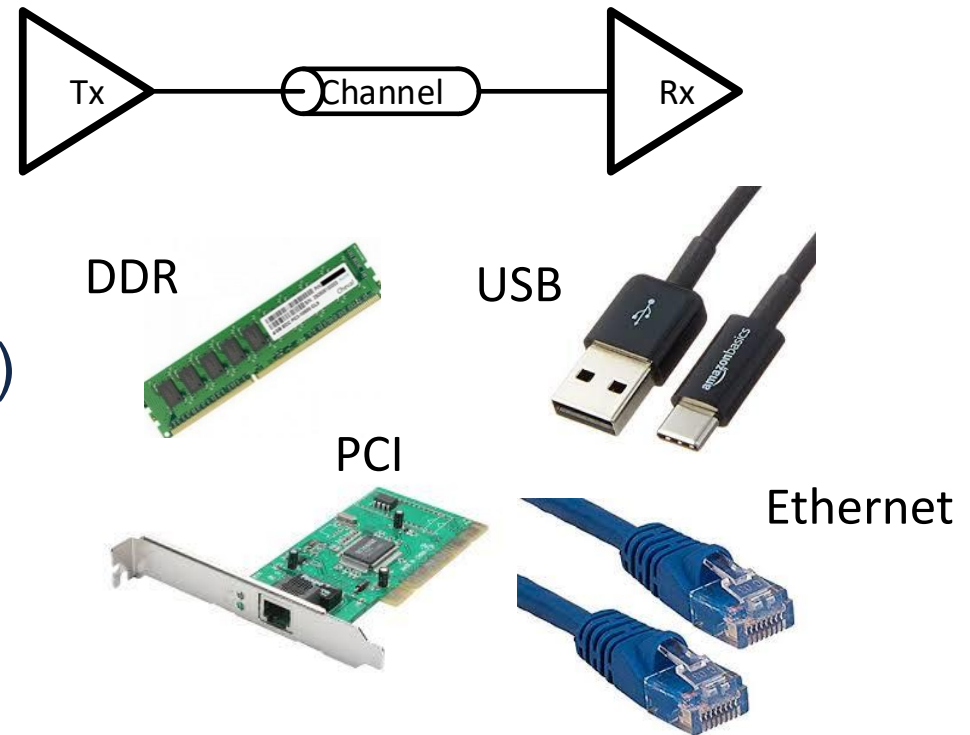
www.spi-conference.org  spi@uni-siegen.de

Outline

- Introduction and motivation
- Volterra series and Laguerre-Volterra feed forward neural network (LVFFN)
- PAM-4 behavior modeling using LVFFN
- IBIS-AMI model implementation and ezAMI software
- Summary

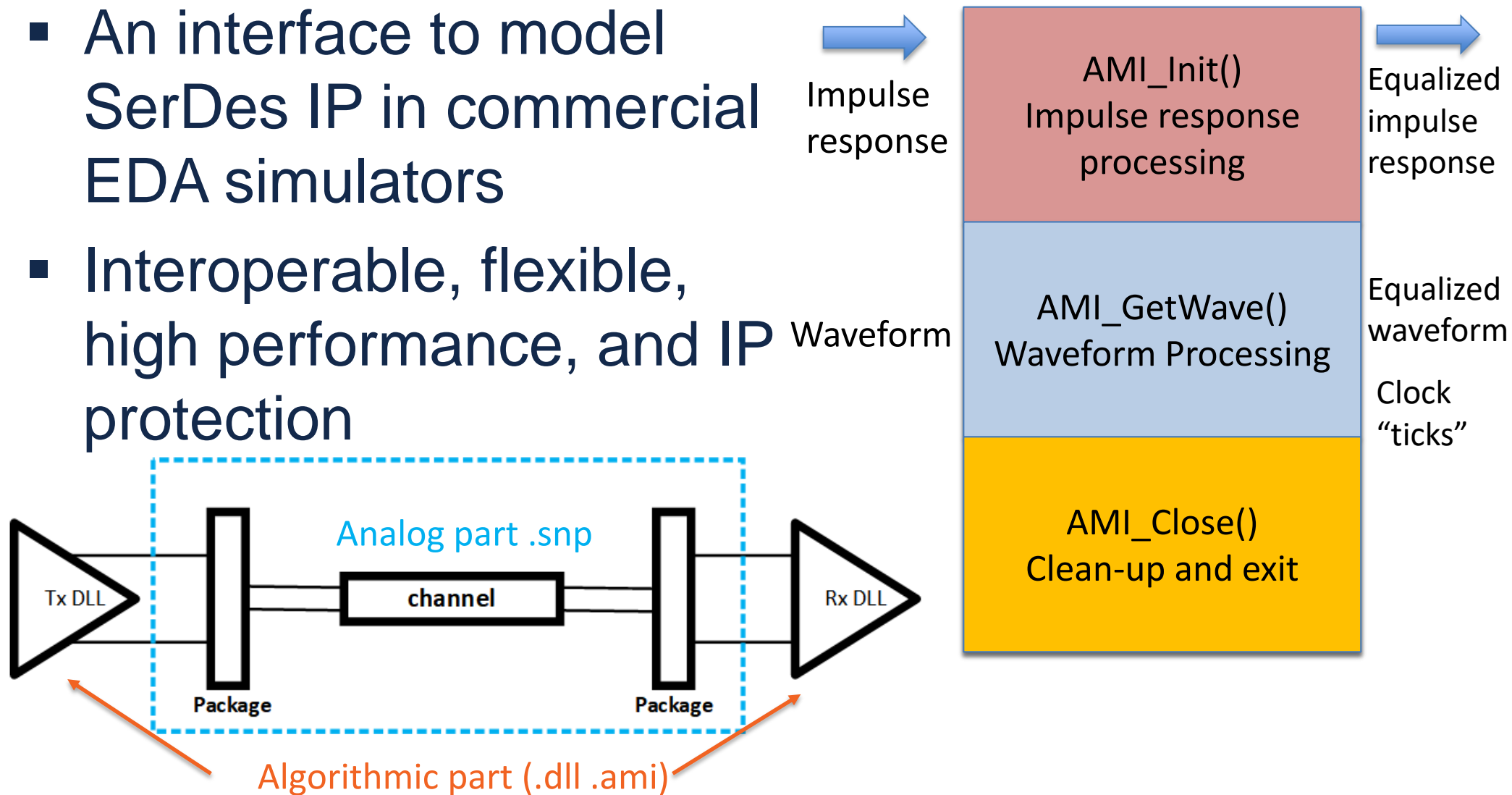
Introduction

- High speed link technology
 - Non-Return-to-Zero (NRZ)
 - Pulse Modulation 4-level (PAM-4)
- Behavioral model
 - A compact mathematical model
 - Fast and accurate
 - IP protection
- Artificial neural network
 - A mathematical mapping of input to output
 - FFN, CNN, RNN



IBIS-AMI Model Basics

- An interface to model SerDes IP in commercial EDA simulators
- Interoperable, flexible, high performance, and IP protection



Motivation and focus of this work

- High demand for rapid data transmission.
100Gb/s -> 400Gb/s
- Simulation and validation are desirable to cut development time and cost
 - A high-performance model is the key
- Behavioral model using machine learning is emerging
 - Large model size
 - Lack of interoperability and transportability
- High speed link behavioral model development
 - A machine learning behavioral model
 - Take nonlinearity into account
 - Fast and accurate
- IBIS-AMI model generation
 - Enhance model interoperability and transportability
 - Compatible with commercial circuit simulator
- IBIS-AMI model generation software

Volterra Series

- A versatile model for nonlinear systems with memory

$$y(t) = y_0 + \sum_{n=1}^{\infty} y_n(t)$$

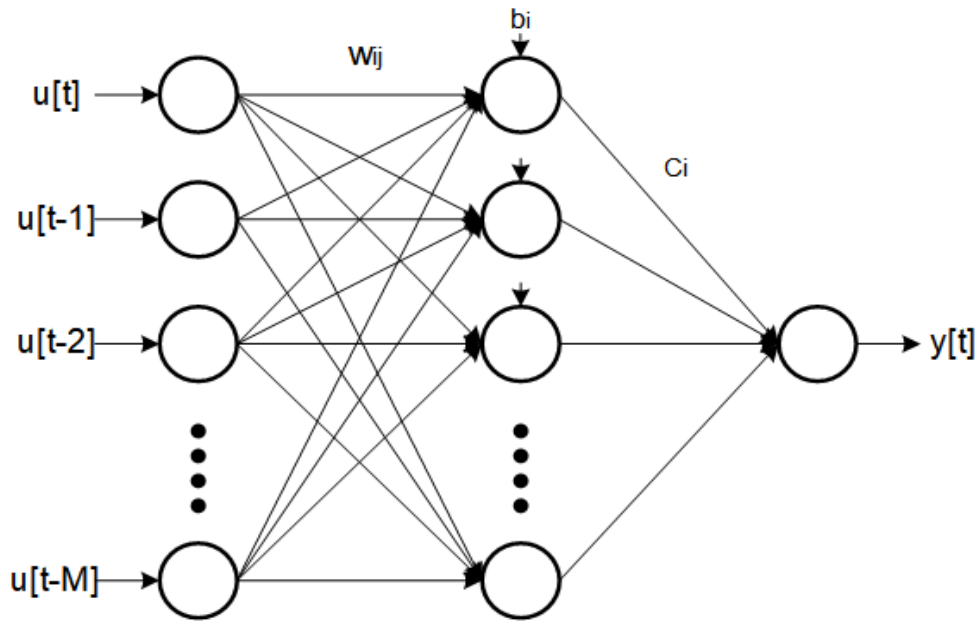
$$y_n(t) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) u(t - \tau_1) u(t - \tau_2) \dots u(t - \tau_n) d\tau_1 d\tau_2 \dots d\tau_n$$

In discrete time

$$y(t) = h_0 + \sum_{\tau_1=0}^{\infty} h_1(\tau_1) u(t - \tau_1) + \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} h_2(\tau_1, \tau_2) u(t - \tau_1) u(t - \tau_2) + \dots$$

$h_0, h_1, h_2, \dots, h_n$ are the Volterra kernels

Monomial Power Series Neural Network



$$y[t] = \sum_{i=1}^M c_i \sigma \left(b_i + \sum_{j=1}^M w_{ji} u[t-j] \right)$$



$$y(t) = h_0 + \sum_{\tau_1=0}^M h_1(\tau_1) u(t - \tau_1) +$$

$$\sum_{\tau_1=0}^M \sum_{\tau_2=0}^M h_2(\tau_1, \tau_2) u(t - \tau_1) u(t - \tau_2) + \dots$$



$$h_0 = \sum_{i=1}^M c_i a_{0i}$$

$$h_1(j) = \sum_{i=1}^M c_i a_{1i} w_{ji}$$

$$h_2(j, k) = \sum_{i=1}^M c_i a_{2i} w_{ji} w_{ki}$$

$$h_n(v_1, v_2, \dots, v_n) = \sum_{i=1}^M c_i a_{ni} w_{j_1 i} w_{j_2 i} \dots w_{j_n i}$$

X Wang, T Nguyen, J Schutt-Aine, EMC Sapporo & AP EMC 2019

T Nguyen, X Wang, J Schutt-Aine, ECTC 2019

Challenges with Volterra Series

- Large number of parameters when nonlinearity order goes high

Number of Volterra Kernels	1^{st} order	2^{nd} order	3^{rd} order	4^{th} order
$M = 10$	10	110	1,110	11,110
$M = 20$	20	420	8,420	168,420
$M = 30$	30	930	27,930	837,930
$M = 40$	40	1,640	641,640	3,201,640

M denotes memory length

Number of Laguerre Parameters	1^{st} order	2^{nd} order	3^{rd} order	4^{th} order
$R = 2$	2	6	14	30
$R = 3$	3	12	39	120
$R = 4$	4	20	84	340
$R = 5$	5	30	155	780

R denotes the number of Laguerre function for expansion

Laguerre-Volterra Feed Forward Neural Network (LVFFN)

- Reduce dimension through projecting Volterra kernels on finite number of Laguerre Functions

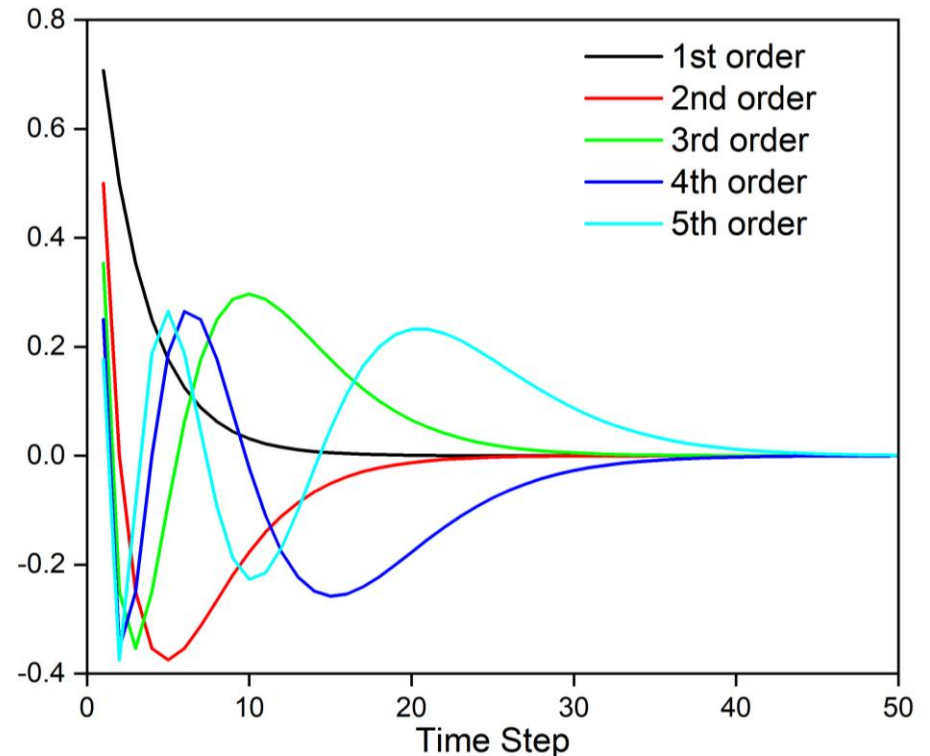
$$h(t) = \sum_{r=1}^R \theta_r \varphi_r(t)$$

- Typically, R is much smaller compared to the memory length M

Discrete Time Laguerre Functions

$$\varphi_r(t) = \alpha^{\frac{t-r}{2}} (1 - \alpha)^{\frac{1}{2}} \sum_{k=0}^r (-1)^k \binom{t}{k} \binom{r}{k} \alpha^{r-k} (1 - \alpha)^k$$

- Guaranteed to converge at infinity
- Functions are orthogonal basis
$$\langle \varphi_i, \varphi_j \rangle = \begin{cases} 0 & i \neq j \\ \sigma & i = j \end{cases}$$
- α is the decay factor which takes value (0,1)
- r is the function order



From Volterra Space to Laguerre Space

Volterra Series

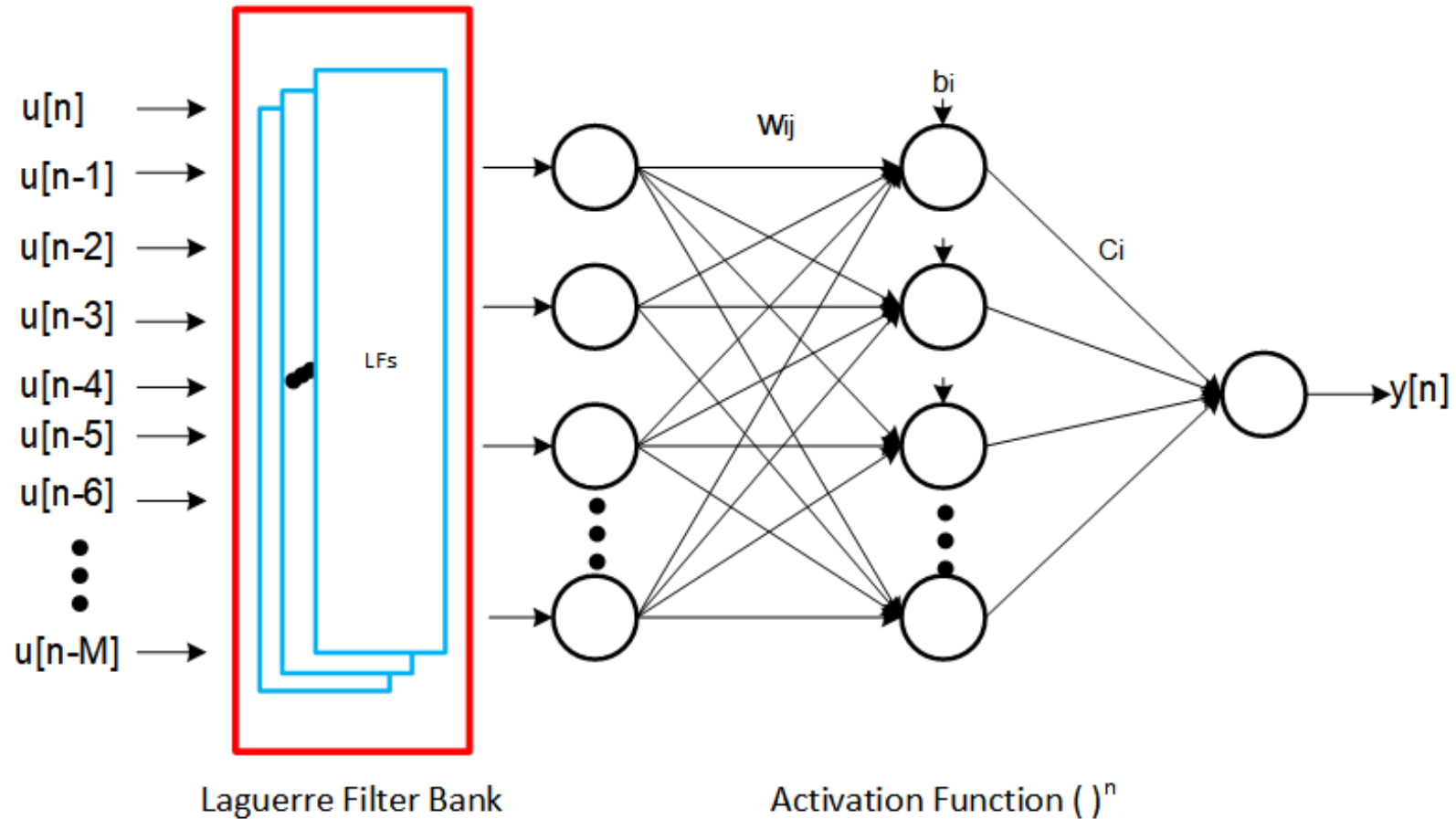
$$y(t) = h_0 + \sum_{\tau_1=0}^M h_1(\tau_1)u(t - \tau_1) + \sum_{\tau_1=0}^M \sum_{\tau_2=0}^M h_2(\tau_1, \tau_2) u(t - \tau_1)u(t - \tau_2) + \dots$$



Laguerre-Volterra model

$$y(t) = \theta_0 + \sum_{r_1=1}^R \theta_1(r_1)v_{r_1}(t) + \sum_{r_1=1}^R \sum_{r_2=1}^R \theta_2(r_1, r_2)v_{r_1}(t)v_{r_2}(t) + \dots$$

Identification of Laguerre Coefficients

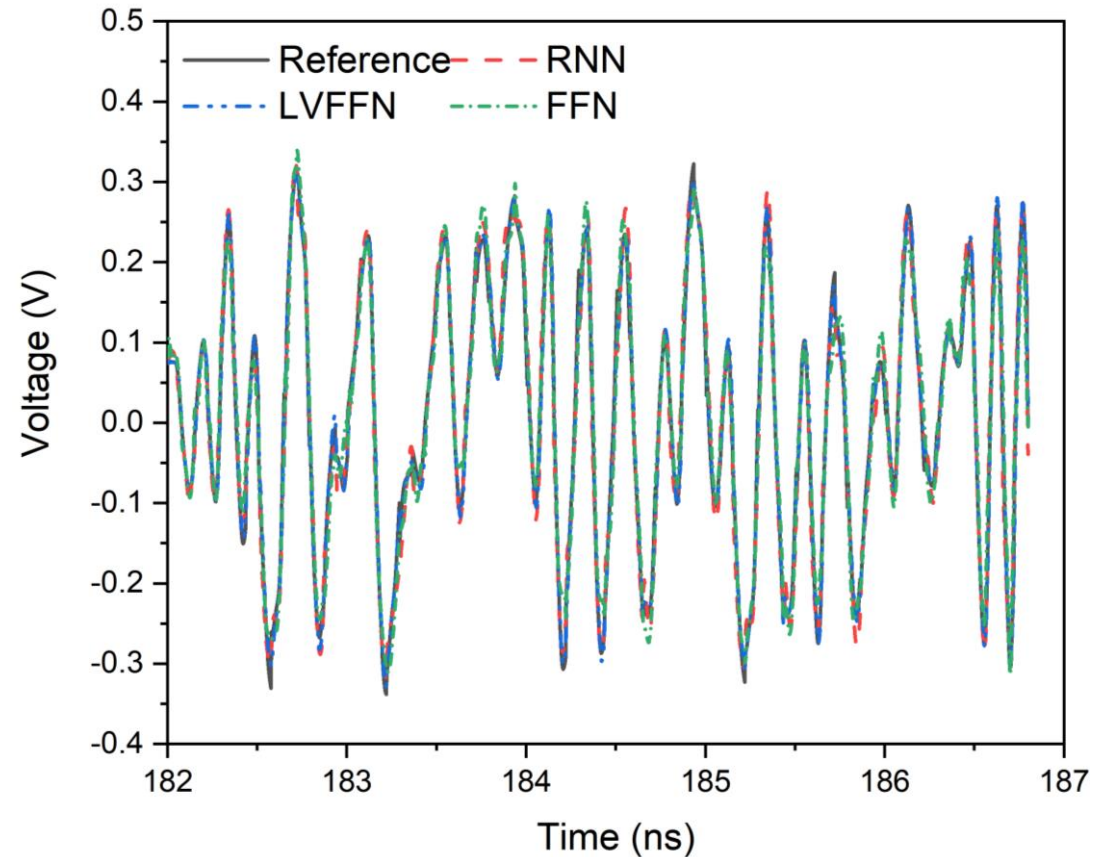
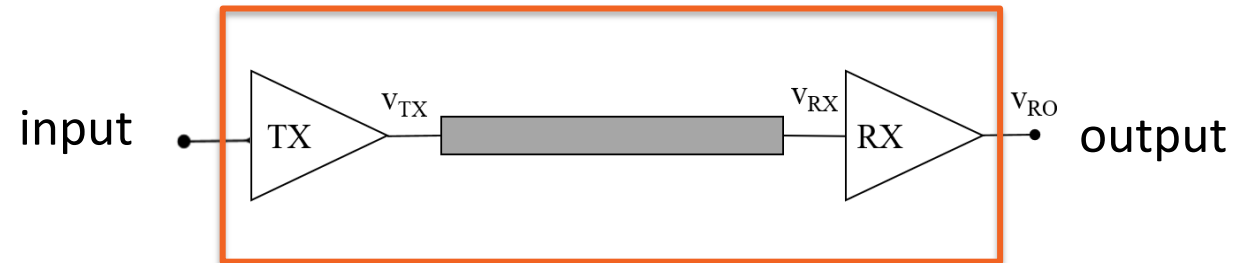


X Wang, T Nguyen, J Schutt-Aine, TCPMT 2020

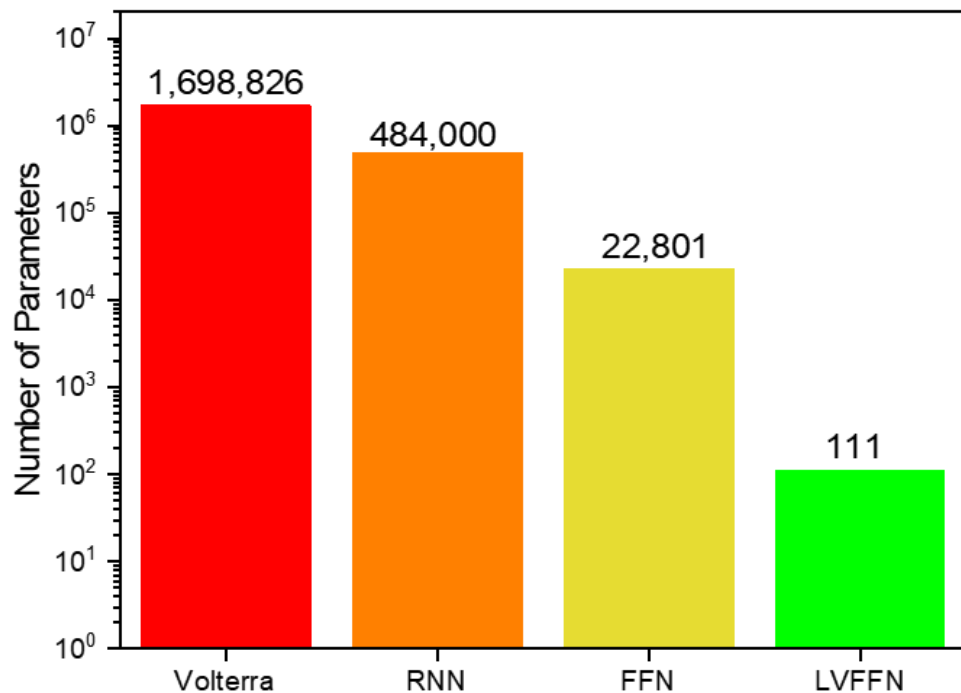
X Wang, T Nguyen, J Schutt-Aine, LASCAS 2020

Modeling PAM-4 system

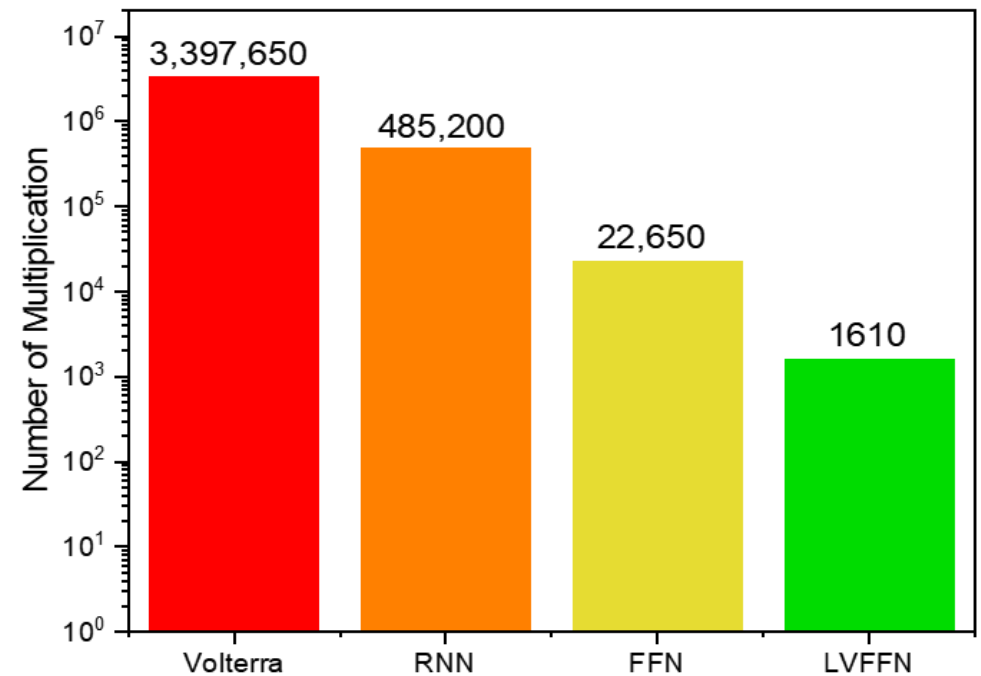
- LVFFN architecture: one hidden layer with 10 neurons
- FFN architecture: one hidden layer with 150 neurons
- RNN architecture:
 - 100 memory length
 - 6 stacked layers
 - 20 neurons for each layer



Model Size Reduction and Computation Efficiency



Model size comparison



Computation efficiency comparison

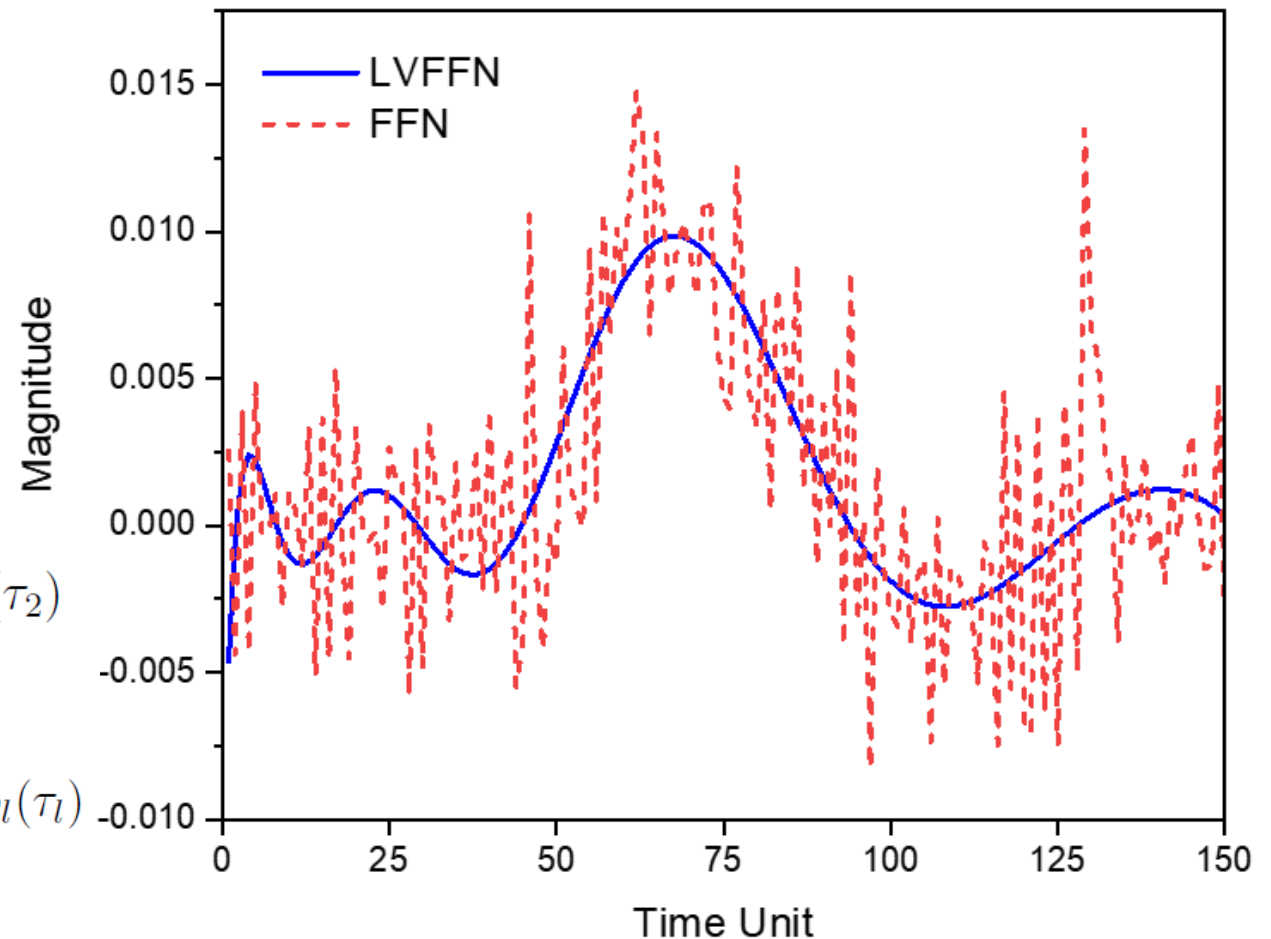
Volterra Kernel Extraction with LVFFN

$$h_0 = \theta_0$$

$$h_1(\tau) = \sum_{r=1}^{r=R} \theta_r \phi_r(\tau)$$

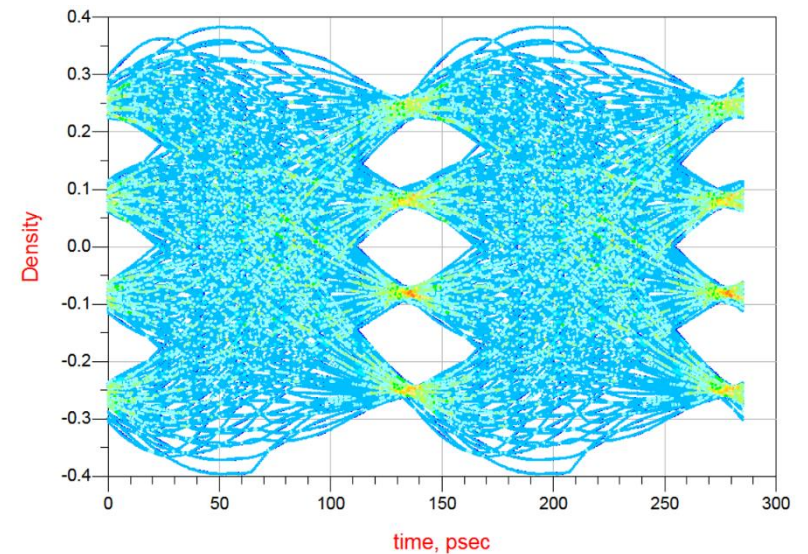
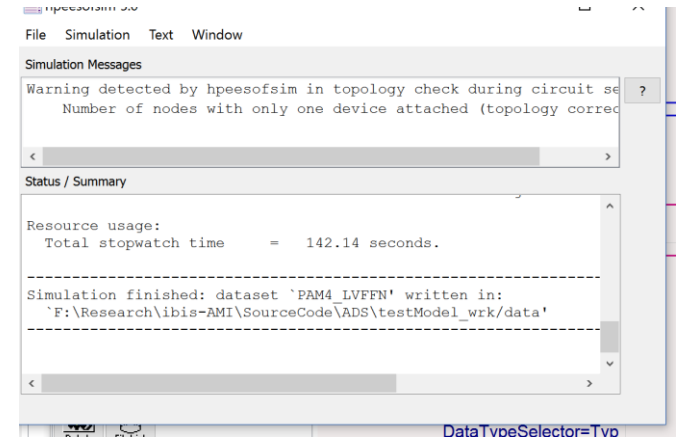
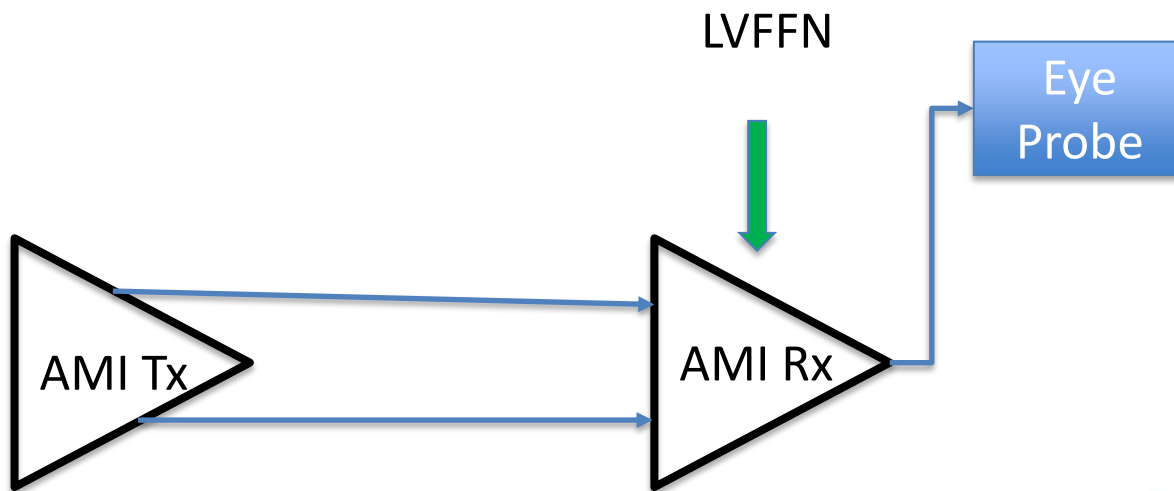
$$h_2(\tau_1, \tau_2) = \sum_{r_1=1}^{r_1=R} \sum_{r_2=1}^{r_2=R} \theta_{r_1, r_2} \phi_{r_1}(\tau_1) \phi_{r_2}(\tau_2)$$

$$h_n(\tau_1, \dots, \tau_n) = \sum_{r_1=1}^{r_1=R} \dots \sum_{r_n=1}^{r_n=R} \theta_{r_1, \dots, r_n} \prod_{l=1}^n \phi_l(\tau_l)$$



Implementation in IBIS-AMI

Simulating 1 million bits takes 142s!



IBIS-AMI Model Generation

- Model generation requires cross-disciplinary knowledge
- ezAMI software facilitates the model generation

Why AMI-model generation takes so long?

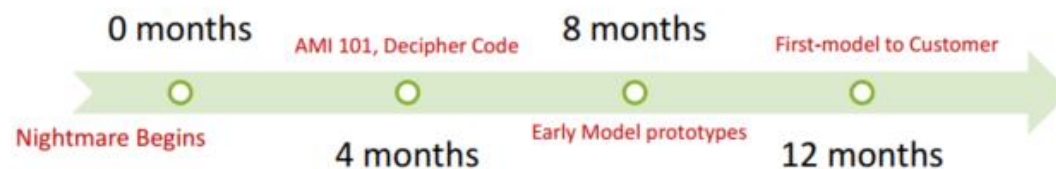


Typical Signal Integrity Engineers are NOT programmers



...they are having "Nightmares" in trying to develop AMI models

- Cryptic Matlab/C++ code passed from System-Architectures → AMI Modeler (if lucky)
- Challenge to Convert Algorithm design Code → AMI format



PSU SI Conference
2015

Page 9

<https://ibis.org/summits/jun10/pino.pdf>

ezAMI software

- IBIS-AMI model generator and simulator
- Supports model pre-generation verification
- Allows on-site development and debug
- Supports NRZ and PAM-4 simulation
- Project-based development
- User-friendly GUI system

Software interface

The screenshot displays the ezAMI1.0 software interface, which is divided into several sections:

- Menu Bar (A):** Located at the top, it includes 'File', 'Edit', 'Project', and 'Help' menus, along with icons for saving and running, and a toolbar with buttons for 'Save', 'Save All', 'Build', 'Run', and 'AMI Generation'.
- Project Browser (B):** A tree view on the left side showing the project structure. It includes a table with columns for 'Project' and 'Path':

Project	Path
LVFFN	F:/Research/ezAMI/AMI/LVFFN.ezproj
Source Code	
ami.cpp	F:/Research/ezAMI/AMI -backup/ami.cpp
ami.h	F:/Research/ezAMI/AMI -backup/ami.h
AMI_container.cpp	F:/Research/ezAMI/AMI -backup/AMI_container.cpp
AMIContainer.h	F:/Research/ezAMI/AMI -backup/AMIContainer.h
amiInterface.h	F:/Research/ezAMI/AMI -backup/amiInterface.h
Executable	
AMI Model	
Resource	
Igfilters.txt	F:/Research/ezAMI/AMI -backup/Igfilters.txt
weight.txt	F:/Research/ezAMI/AMI -backup/weight.txt
- Block Diagram (C):** A schematic diagram showing a signal flow from an input waveform through a block labeled 'AMI' to an output waveform.
- Code Editor (D):** A window showing the source code for 'ami.cpp'. The code includes a header file and defines three functions: 'AMI_Init', 'AMI_GetWave', and 'AMI_Close'.

```
//This is the ami interface file.
#include "ami.h"

long AMI_Init(double *impulse_matrix,
              long row_size,
              long aggressors,
              double sample_interval,
              double bit_time,
              char **AMI_parameters_in,
              char **AMI_parameters_out,
              void **AMI_memory_handle,
              char **msg) {
    return 1;
}

/*****
long AMI_GetWave(double *wave,
                 long wave_size,
                 long aggressors,
                 double *clock_times,
                 char **AMI_parameters_out,
                 void *AMI_memory) {
    return 1;
}

/*****
long AMI_Close(void *AMI_memory) {
    return 1;
}
```

Software download links

- <https://gitlab.engr.illinois.edu/xinying/ezami>
- <https://github.com/WXY163/ezAMI>
- Installer:
<https://github.com/WXY163/ezAMI/tree/master/installer>
- Tutorial :
<https://github.com/WXY163/ezAMI/blob/master/documents/totutorial.pdf>

Quick demo...

Summary

- Proposed a Laguerre-Volterra feed forward neural network (LVFFN) which can significantly reduce the model size and enhance the computation efficiency for modeling PAM-4 systems
- Implemented the PAM-4 LVFFN model in IBIS-AMI and simulated it in industrial EDA tools
- Developed an IBIS-AMI model generation software ezAMI which can help developing non-traditional SERDES IBIS-AMI model.

Thank you!

