Senior Design Project Plan---High Speed Network Connection

**Introduction**

This senior design project is about network based analysis for high speed electronics. We will be analyzing networks such as serial link. The vision of this project is to improve the current algorithms we have on network analysis technology to be more accurate and less prone to error.

**Development Process and Deliverables**

Concept -> Derivation -> Declaration -> Definition -> Integration -> Test and Validation -> Documentation -> Report -> Release

Concept and Derivation already completed;

Declaration will be done in MATLAB using a set of functions to achieve the new algorithm;

Definition will be done in MATLAB to generate the program;

Integration to include other sources, objects: FFT, network parameter analysis routines;

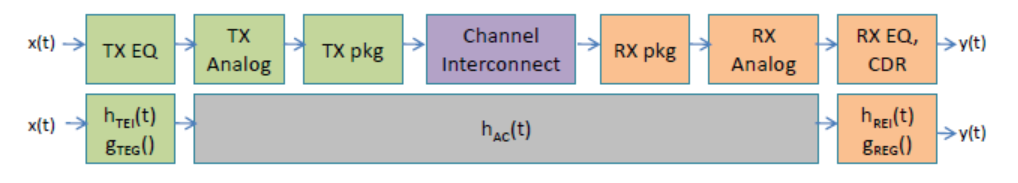
Test and Validation: to test program using examples, generate test report;

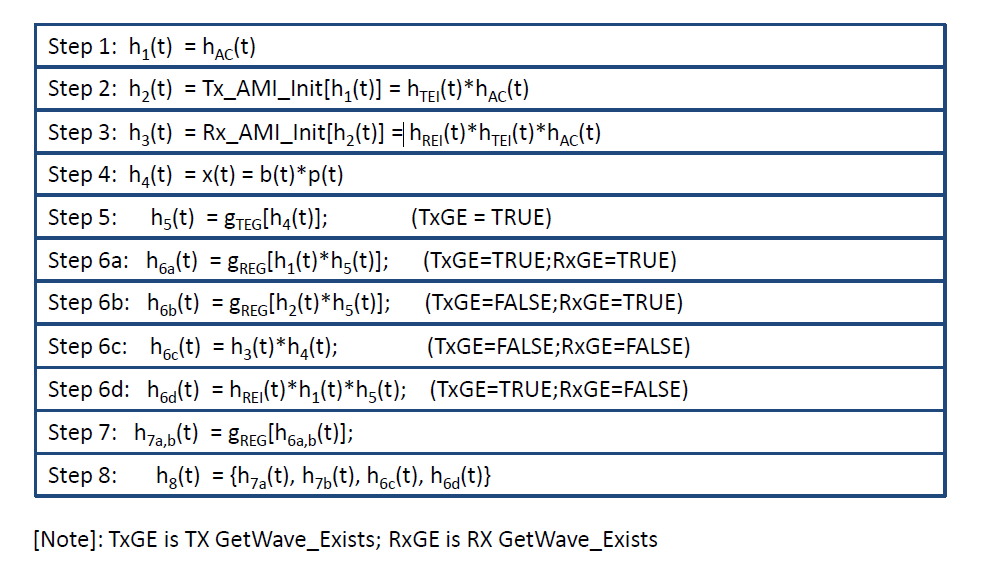
Documentation: on classes, methods, generate reference guide and/or user guide;

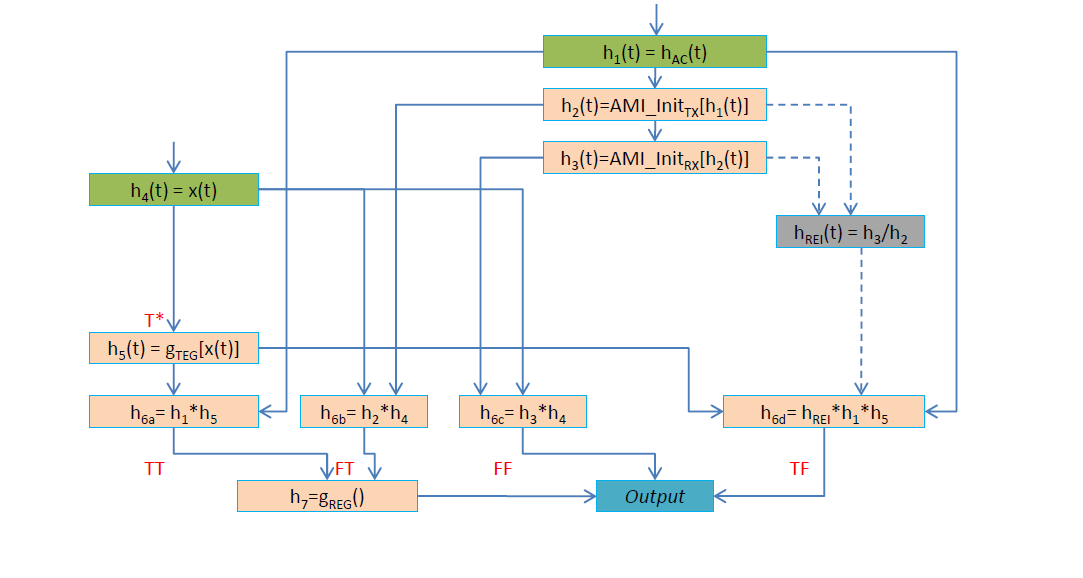
Report: senior design report;

Release: the entire package including documents, coding archives, executable, reports, notes;

**Process and Plan**

The reference flow of the project is as follows:





The flow chart in the top is analysis of an end-to-end serial link network while the one in the below is the branch of 4 different situations.

First, we will concatenate n 4-port (is three 4-port networks in our project) networks to an impulse response hAC(t). The RX block will work to restore the input signal x(t) = b(t) \* p(t) after it passing through the network. There have 4 situations to get the output:

1. TX GetWave\_Exists=TRUE; RX GetWave\_Exists=TRUE. Then we will use TX\_GetWave function, to get the output of TX (t) or (t) = gTEG[x(t)]. After that will convolve (t) with hAC(t) to get the time domain response (t). Last, because the response (t) has serious distortion, we will use RX\_GetWave function to get the final output y(t) = gREG((t)).
2. TX GetWave\_Exists=FALSE; RX GetWave\_Exists=TRUE. Using the method in the above chart, we have: (t) = x(t), (t) = (t) \* hTEI(t) \* hAC(t) (mention that, AMI\_InitTX[hAC(t)] = hTEI(t) \* hAC(t) ), y(t) = gREG((t)).
3. TX GetWave\_Exists=FALSE; RX GetWave\_Exists=FALSE. We have: (t) = x(t), (t) = (t) \* hTEI(t) \* hAC(t), y(t) = hREI(t) \* (t).
4. TX GetWave\_Exists=TRUE; RX GetWave\_Exists=FALSE. We have: (t)= gTEG[x(t)], (t) = (t) \* hAC(t), y(t) = hREI(t) \* (t).

AMI\_Init() functions are for LTI systems and AMI\_GetWave() functions are for NLTV systems .

We can obtain hAC(t) by steps as follows.

1. Using concatenate code to get the Z-parameters.
2. Using the algorithm of differential only in serdes1211.pdf to compute the transfer function (jw).
3. Then we will convert (jw) to time domain by using IFT and obtain hAC(t).

I’m now working on the getting hAC(t) section, and I think I can complete it by 2/12/2015. The next step, getting the time domain response (t) including 4 different situations will be completed by 2/19/2015. The last step of getting the final output y(t) will be coded by 2/26/2015. Then I will debug the program and complete this section by 3/12/2015. From mid-March to mid-May, I will run some examples and compare the results, write the report, put all the theory derivations together.

Yuan Yao