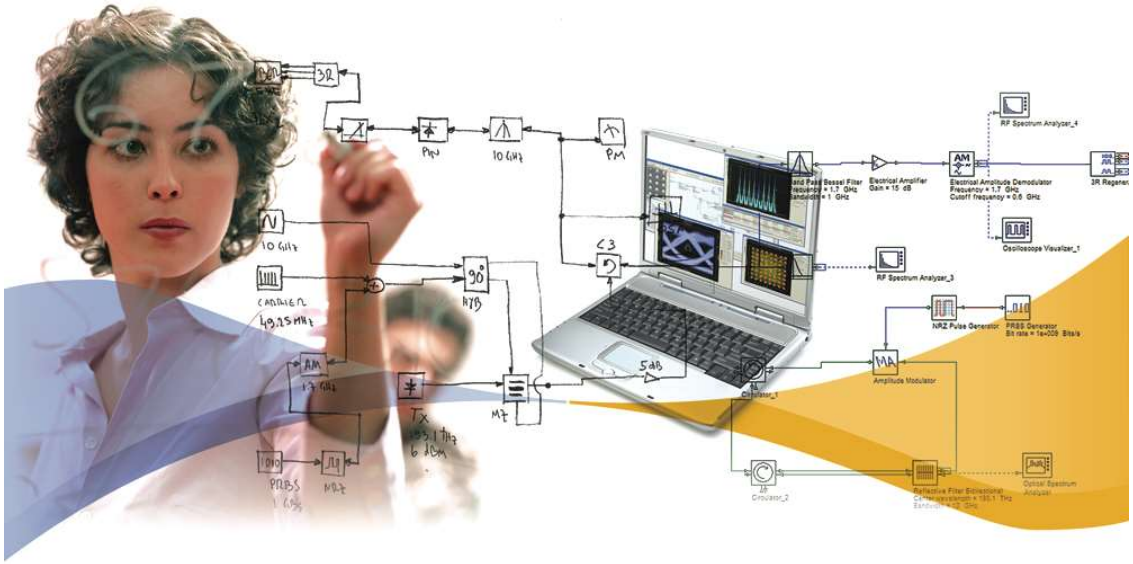


OptiSystem-MATLAB data interchange model and features

7 Capella Court
Nepean, ON, Canada
K2E 7X1

+1 (613) 224-4700
www.optiwave.com

Optical signal data format (1)

Signal type	Data elements	Comments
Sampled	***InputPort1.Sampled.Signal***	Represents the complex envelope of the optical signal (real/imag) – *1xn complex double* If there are two polarization states, two rows will be created (*2xn complex double*)
	InputPort1.Sampled.Time ***Inputport1.Sampled.Frequency***	The time/frequency sampling points for the sampled optical signal (s or Hz) • If the parameter **Sampled signal domain** = "Time", use ***InputPort1.Sampled.Time*** • If the parameter **Sampled signal domain** = "Frequency", use ***InputPort1.Sampled.Frequency***
	InputPort1.Sampled.CentralFrequency	The center frequency (Hz) of the optical signal
Sampled (Channels)	***InputPort1.Channels***	List of wavelength channels entering specified port • To access data for a sampled signal (Channel A), use ***InputPort1.Sampled(A).Signal***, etc. • To access data for a parameterized signal (Channel A), use ***InputPort1.Parameterized.Power(A)***, etc.
Sampled (Spatial)	***InputPort1.Sampled.Spatial.ModeX.Amplitude***	Real or complex amplitude of spatial mode(s) – *nxn array* Note 1: To access Y polarization data, use ***ModeY*** in lieu of ***ModeX*** Note 2: If more then one mode is present, separate sampled signals will be created for each mode and can be accessed as follows (for Mode A): ***InputPort1.Sampled(A).Spatial.ModeX.Amplitude***
	InputPort1.Sampled.Spatial.ModeX.Properties	String value (describes mode type and index)
	InputPort1.Sampled.Spatial.ModeX.DeltaSpaceX ***InputPort1.Sampled.Spatial.ModeX.DeltaFrequencyX***	X-polarization: Discretization in space (m) or discretization in frequency (1/m)
	InputPort1.Sampled.Spatial.ModeX.DeltaSpaceY ***InputPort1.Sampled.Spatial.ModeX.DeltaFrequencyY***	Y-polarization: Discretization in space (m) or discretization in frequency (1/m)
Parameterized	***InputPort1.Parameterized.Power***	Average power of parameterized optical signal (W)
	InputPort1.Parameterized.Frequency	Central frequency of parameterized optical signal
	InputPort1.Parameterized.SplittingRatio	Polarization splitting ratio of parameterized optical signal
	InputPort1.Parameterized.Phase	Phase of parameterized optical signal

Optical signal data format (2)

Signal type	Data elements	Comments
Noise	***InputPort1.Noise.Power***	Average power of each noise bin (W)
	InputPort1.Noise.LowerFrequency;	Lower frequency range of each noise bin (Hz)
	InputPort1.Noise.UpperFrequency;	Upper frequency range of each noise bin (Hz)
	InputPort1.Noise.Phase	Phase of each noise bin (Hz)
Individual sample	***InputPort1.IndividualSample***	Represents the complex amplitude of the optical signal for a single sampling point

Electrical & Binary/M-ary data formats

Electrical

Signal type	Data elements	Comments
Sampled	***InputPort1.Sampled.Signal***	Represents the electrical signal sampled waveform (real/imag) – *1xn complex double*
	InputPort1.Sampled.Time ***Inputport1.Sampled.Frequency***	The time/frequency sampling points for the sampled electrical signal (s or Hz) • If the parameter **Sampled signal domain** = "Time", use ***InputPort1.Sampled.Time*** • If the parameter **Sampled signal domain** = "Frequency", use ***InputPort1.Sampled.Frequency***
Noise	***InputPort1.Noise.Signal***	Represents the electrical noise sampled waveform (real/imag) – *1xn complex double* Note: If the noise is combined with the sampled signal (before the MATLAB Component) these arrays will be empty (zero values)
	InputPort1.Noise.Time ***Inputport1.Noise.Frequency***	The time/frequency sampling points for the sampled electrical noise (s or Hz) • If the parameter **Sampled signal domain** = "Time", use ***InputPort1.Noise.Time*** • If the parameter **Sampled signal domain** = "Frequency", use ***InputPort1.Noise.Frequency***
Individual sample	***InputPort1.IndividualSample***	Represents the amplitude of the electrical and noise signal for a single sampling point

Binary & M-ary

Signal type	Data elements	Comments
Binary	***InputPort1.Sequence***	Represents the sequence of binary bits (0's and 1's)
	InputPort1.BitRate	Bit rate of binary sequence (1/s)
M-ary	***InputPort1.Sequence***	Represents the sequence of M-ary symbols – *1xn double*
	InputPort1.BitRate	Sample rate of M-ary sequence (1/s)

MATLAB m-file example (OpticalData.m)

The data structure of any input port can be equated to any output port as long as they are the same type (optical, electrical, m-ary, binary)

Ports are accessed using the nomenclature InputPort1, InputPort2, etc.

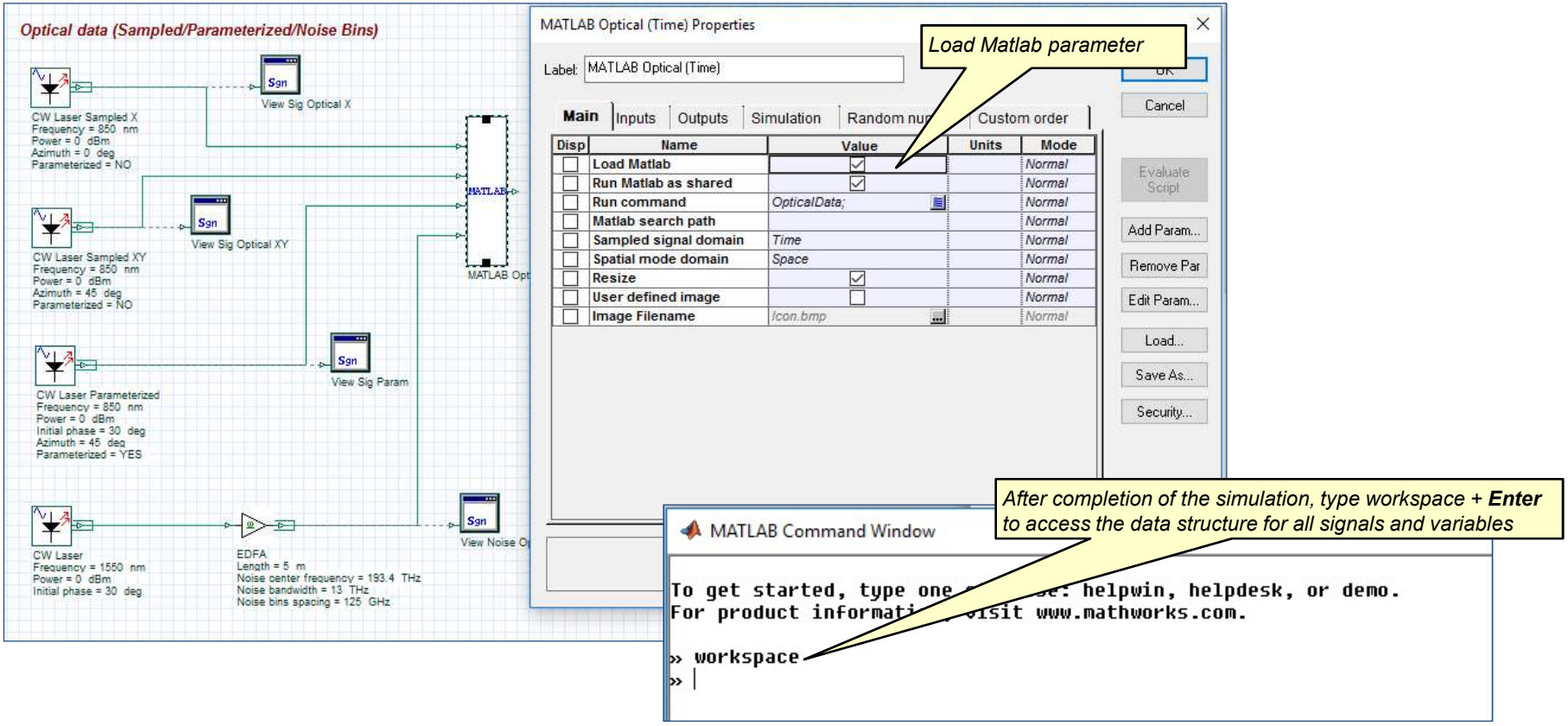
Variables on left are local to the MATLAB workspace.

OpticalData.m

```
OutputPort1 = InputPort1;

%OPTICAL SAMPLED SIGNAL (InputPort1.Sampled; InputPort1.Sampled)
%--------------------------------------------------------------------------

%Signal with 1 polarization state (default) - InputPort1
OpticalSignal_Envelope = InputPort1.Sampled.Signal; %Complex envelope of the optical signal (real/imag amplitudes)
OpticalSignal_Time = InputPort1.Sampled.Time; %Sampled time array (s)
OpticalSignal_CtrFreq_S = InputPort1.Sampled.CentralFrequency; %Central frequency of the optical signal (Hz)

%Signal with 2 polarization states - InputPort2

OpticalSignal_Envelope_XY = InputPort2.Sampled.Signal; %Complex envelope of the optical signal (X data in 1st row ;
OpticalSignal_Time_XY = InputPort2.Sampled.Time; %Sampled time array
OpticalSignal_CtrFreq_XY = InputPort2.Sampled.CentralFrequency; %Central frequency of the optical signal

%OPTICAL PARAMETERIZED SIGNAL (InputPort3.Parameterized)
%--------------------------------------------------------------------------

%Parameterized signal
OpticalSignal_Pwr = InputPort3.Parameterized.Power; %Average power of parameterized optical signal (W)
OpticalSignal_CtrFreq_P = InputPort3.Parameterized.Frequency; %Central frequency of parameterized optical signal
OpticalSignal_SR = InputPort3.Parameterized.SplittingRatio; %Polarization splitting ratio of parameterized optical s
OpticalSignal_Ph = InputPort3.Parameterized.Phase; %Absolute phase of parameterized optical signal

%NOISE BINS (InputPort4.Noise)
%--------------------------------------------------------------------------

NoiseArray_Pwr = InputPort4.Noise.Power; %Average power of each noise bin (W)
NoiseArrayLowFreq = InputPort4.Noise.LowerFrequency; %Lower frequency range of each noise bin (Hz)
NoiseArrayHighFreq = InputPort4.Noise.UpperFrequency; %Upper frequency range of each noise bin (Hz)
NoiseArrayPhase = InputPort4.Noise.Phase; %Phase of each noise bin (rad)
```

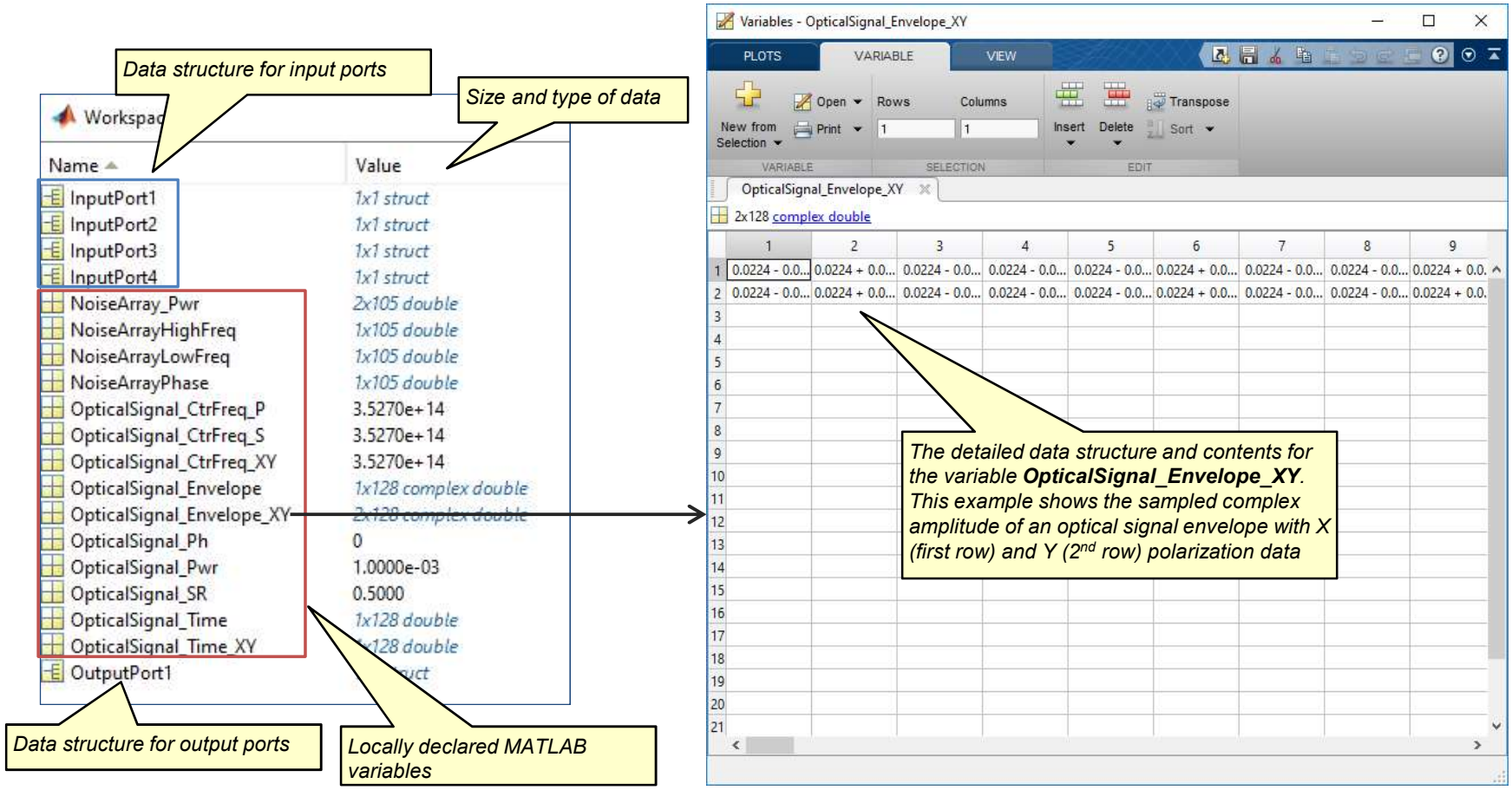
Accessing the MATLAB workspace

1. The data structure for all input and output ports and all variables declared within the MATLAB m-file can be viewed from the MATLAB workspace
2. To access the workspace, first select **Load MATLAB** from the ***MATLAB Component*** and select OK. This action pre-loads MATLAB (it will stay open unless it is manually closed)
3. After running a simulation, open the MATLAB Command Window and type “workspace”.



Example of workspace for *Optical_Data.osd*

To view further details on a data structure, double left click on any variable to open up the Variables window



How to configure output ports

1. The data structure for all input ports is automatically created during the simulation, however output port data structures need to be configured from the MATLAB m-file
2. Two methods can be used:
   1. Set an output port to be equal to an input port
   2. Manually set the attributes of the data structure through declarations within the workspace

Method 1: Set output port equal to an input port

```
% Creating an output structure similar to the input
OutputPort1 = InputPort1;
OutputPort2 = InputPort2;
```

*When using this method the data type for the two linked ports **must be the same***
*NOTE: Once the data structure is in place its contents can be modified within the MATLAB m-file (for example by performing math operations on the amplitude data). Also the size of the arrays can be changed but the size of the time/frequency and sampled signal arrays associated with the output port **must match!**.*

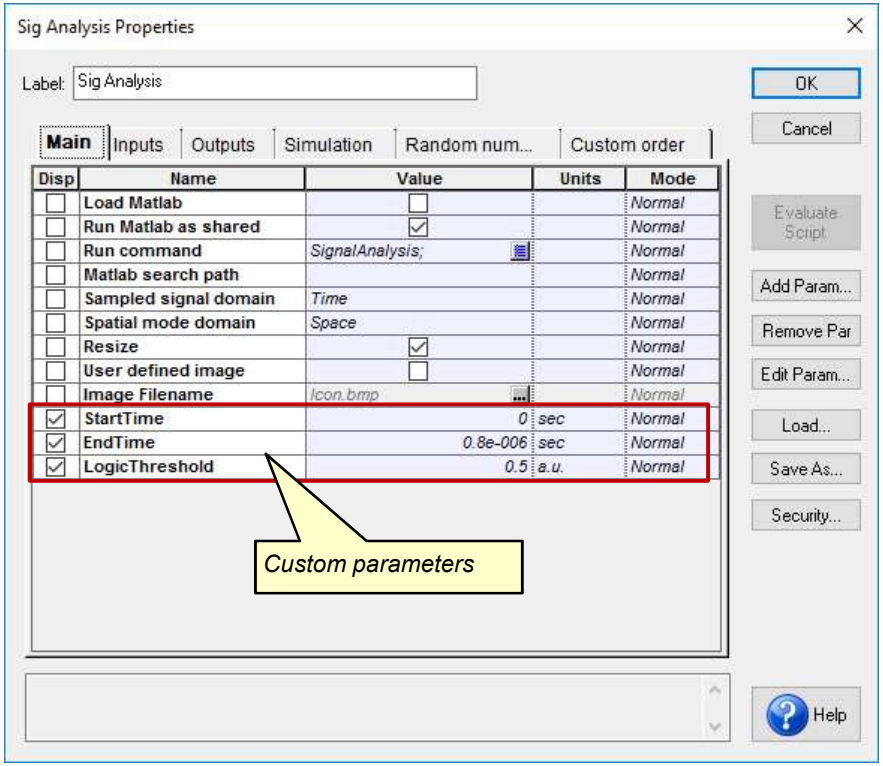
Method 2: Set output port(s) explicitly

```
OutputPort1.TypeSignal           = 'Electrical';
OutputPort1.Noise                = [];
OutputPort1.IndividualSample     = [];
OutputPort1.Sampled.Signal       = SignalI;
OutputPort1.Sampled.Time         = TimeOS;

OutputPort2.TypeSignal           = 'Electrical';
OutputPort2.Noise                = [];
OutputPort2.IndividualSample     = [];
OutputPort2.Sampled.Signal       = SignalQ;
OutputPort2.Sampled.Time         = TimeOS;
```

In this example, Output Port 1 and Output Port 2 are fully defined from within the MATLAB workspace. Make sure that the defined signal type matches the port setting for the component

Linking parameters between OptiSystem & MATLAB

Custom parameters can be added to the MATLAB component and accessed within the MATLAB workspace during a simulation (the example project folder can be found under “MATLAB Signal Analysis”)



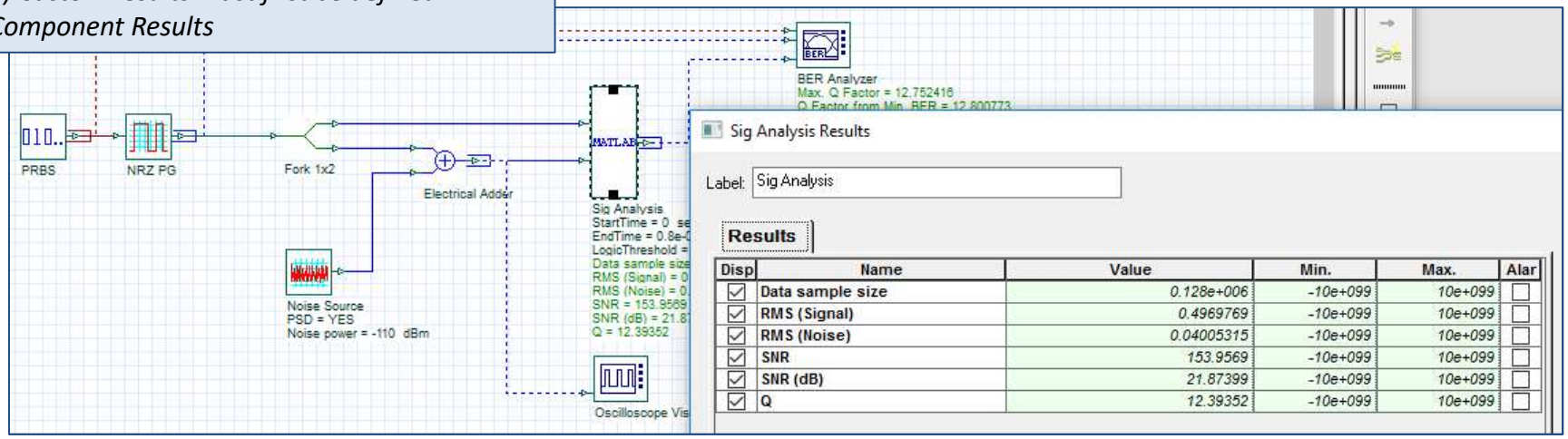
MATLAB m-file. The left variable is local to the MATLAB workspace whereas the right variable must match the OptiSystem parameter

```
% Creating an out      ucture similar to the input
OutputPort1 = In      ort2;

% Set time wi    ow for reading in data
TStart = StartTime;
TEnd = EndTime;
LogicTh = LogicThreshold;

TotalNumSamples = length(InputPort1.Sampled.Signal);
```

Linking MATLAB variables to Results (1)

Calculations performed in MATLAB can be displayed as a Component Result (the example project folder can be found under “MATLAB Signal Analysis”)

1) Custom results must first be defined in Component Results



2) Calculations performed in MATLAB are exported as a text file (to same folder where OptiSystem project is located)

```
% PRINT RESULT TO TEXT FILE
fid=fopen('C:\Users\mverreault\Desktop\MATLAB Signal Analysis\SignalAnalysis.txt','w');
fprintf(fid,'%E \r\n', DataSampleSize);
fprintf(fid,'%E \r\n', RMS_Signal_Amp);
fprintf(fid,'%E \r\n', RMS_Noise_Amp);
fprintf(fid,'%E \r\n', SNR);
fprintf(fid,'%E \r\n', SNR_dB);
fprintf(fid,'%E \r\n', Q);
fclose(fid);
```

Linking MATLAB variables to Results (2)

3) The OptiSystem Component script feature is then used to access and display the results

```
Option Explicit

Dim file, input, Data1, Data2, Data3, Data4, Data5, Data6
CONST ForReading = 1

'Create a File System Object
Set file = CreateObject("Scripting.FileSystemObject")

'Open the text file
Set input = file.OpenTextFile ("SignalAnalysis.txt", ForReading)

'Step through the lines
Data1 = input.ReadLine
Data2 = input.ReadLine
Data3 = input.ReadLine
Data4 = input.ReadLine
Data5 = input.ReadLine
Data6 = input.ReadLine

Dim ThisComponent
Set ThisComponent = GetThisComponent()

ThisComponent.SetResultValue "Data sample size" , Cdbl(Data1)
ThisComponent.SetResultValue "RMS (Signal)" , Cdbl(Data2)
ThisComponent.SetResultValue "RMS (Noise)" , Cdbl(Data3)
ThisComponent.SetResultValue "SNR" , Cdbl(Data4)
ThisComponent.SetResultValue "SNR (dB)" , Cdbl(Data5)
ThisComponent.SetResultValue "Q" , Cdbl(Data6)
```