Creating Graphical Results from Simulink Models

The previous tutorial required you to create three separate Simulink models for the BPSK, QPSK, and QAM modulation schemes. Make sure you have them as they are needed to do this tutorial.

The BERTool (Bit Error Rate Tool) provides a graphical user interface (GUI) for viewing and analysing the BER in communication systems.

By the end of this tutorial you should be able to:

- Generate BER data for your Simulink models
- Plot BER data for multiple Simulink models on a single graph
- Fit a curve for the set of simulation data
- Compare actual results with theoretical results

Getting Started

Open one of your Simulink models that you created from the previous tutorial. For demonstration purposes BPSK is used here, but any one will be fine.

Defining variables

We now need to define three variables using the Matlab command line:

- EbNo (Signal-to-noise ratio)
- MaxErrs (Maximum errors i.e. when the simulation will stop)
- MaxBits (Maximum number of bits)

In the Matlab command window, enter the following:

EbNo = 0; MaxErrs = 100; MaxBits = 1e8

Configuring Simulink Model

We now need to make a few changes to the Simulink model in order to link it to the above variables and configure it to send data to the Matlab workspace rather than the display block we have been using so far, as shown in Figure 1.

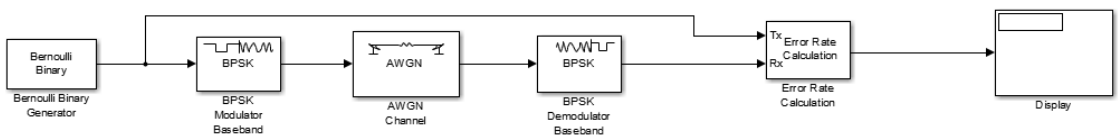

Figure 1 - Viewing Results with Display Block

We need to make changes to the following blocks:

- Error Calculation
- AWGN Channel

Additionally, we need to insert an additional block called **Signal to Workspace**

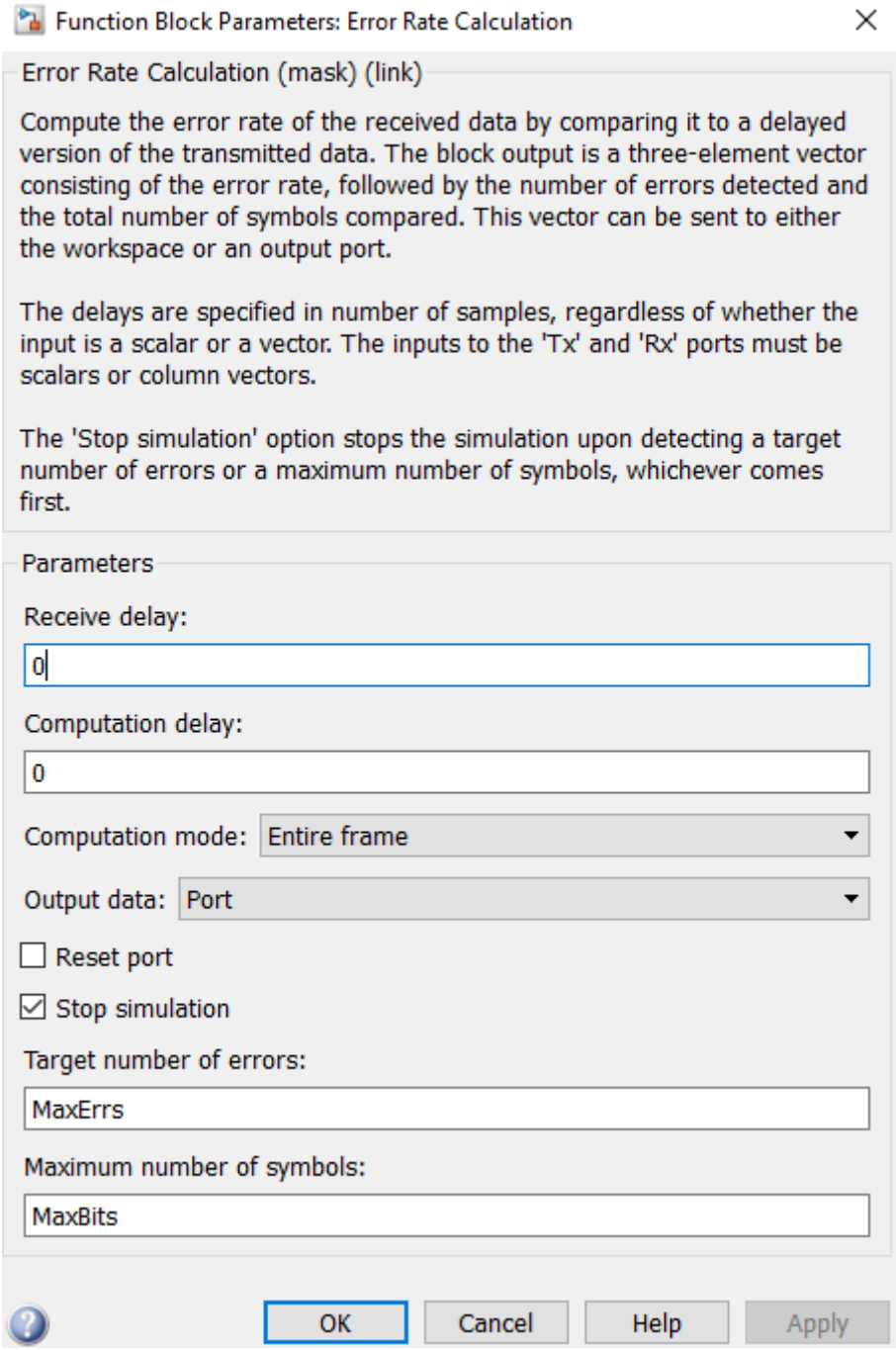

Figure 2 - Error Calculation Configuration

As shown in Figure 2, there are two changes we need to make to the error rate calculation block:

- *Target number of errors* – set this to **MaxErrs**, which we defined earlier.
- *Maximum number of symbols* – set this to **MaxBits**, which we defined earlier.

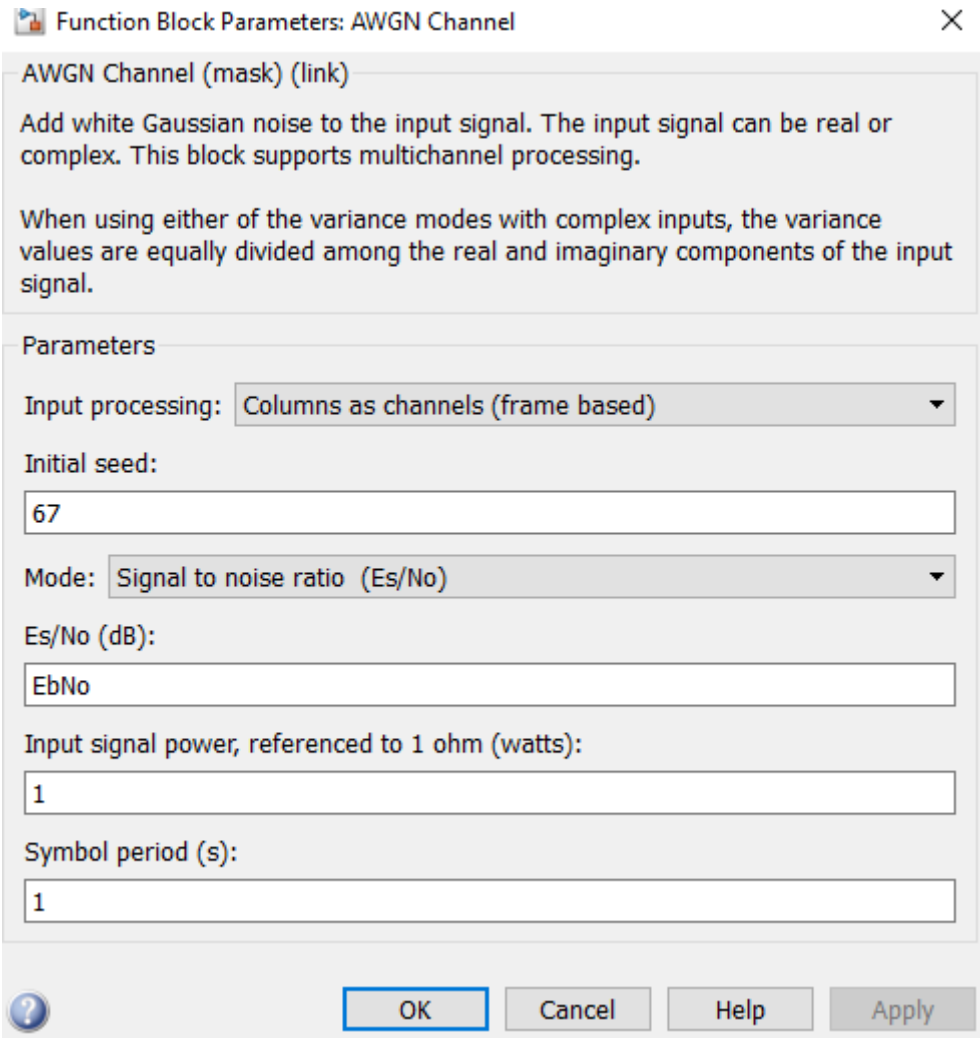

Figure 3 - AWGN Block Configuration

In the AWGN block, shown in Figure 3, make the following changes:

- Set the Es/No (db) value to **EbNo**, the variable we defined earlier.

Finally, we need to insert an additional block to link the Simulink model to the Matlab workspace. For this, we need the **Signal to Workspace** block which is found in the DSP System Toolbox/Sinks library.

Please note: the name of this block may be different depending on the Matlab version you use. In the latest version (2015b) it is just called "To Workspace".

This block needs to be placed in between the error rate calculation and the display block, as shown in Figure 4. This enables us to see numerical results in Simulink as well as generating them graphically.

Once you have added the block linking the Simulink model to the workspace, it should look similar to Figure 4:

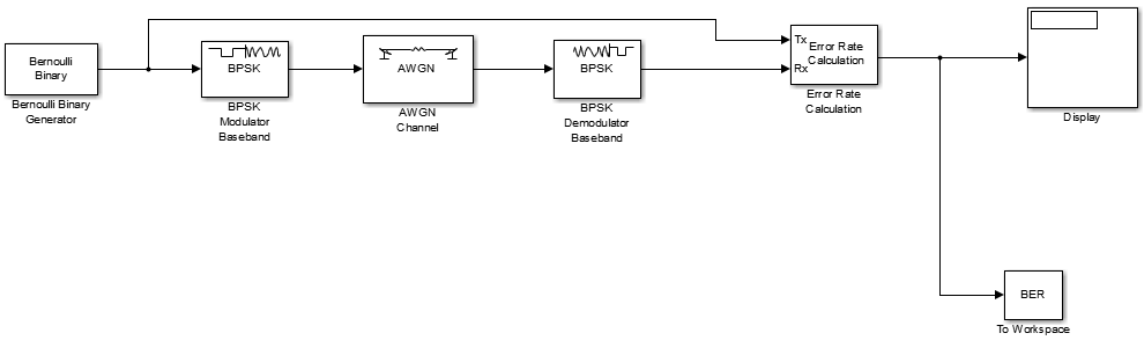

Figure 4 - Simulink to Workspace

Finally, double click on the block and set the following parameters:

- *Variable name* – this is what will link the Simulink model to the BERTool. You can set this to whatever you want but remember it because it is needed later. In this case, **BER** is used.
- *Limit data points to last* – set this to a value of **1**.

Save your model as this point.

Now, from the Matlab command line, type **bertool**. This will bring up the window shown in Figure 5.

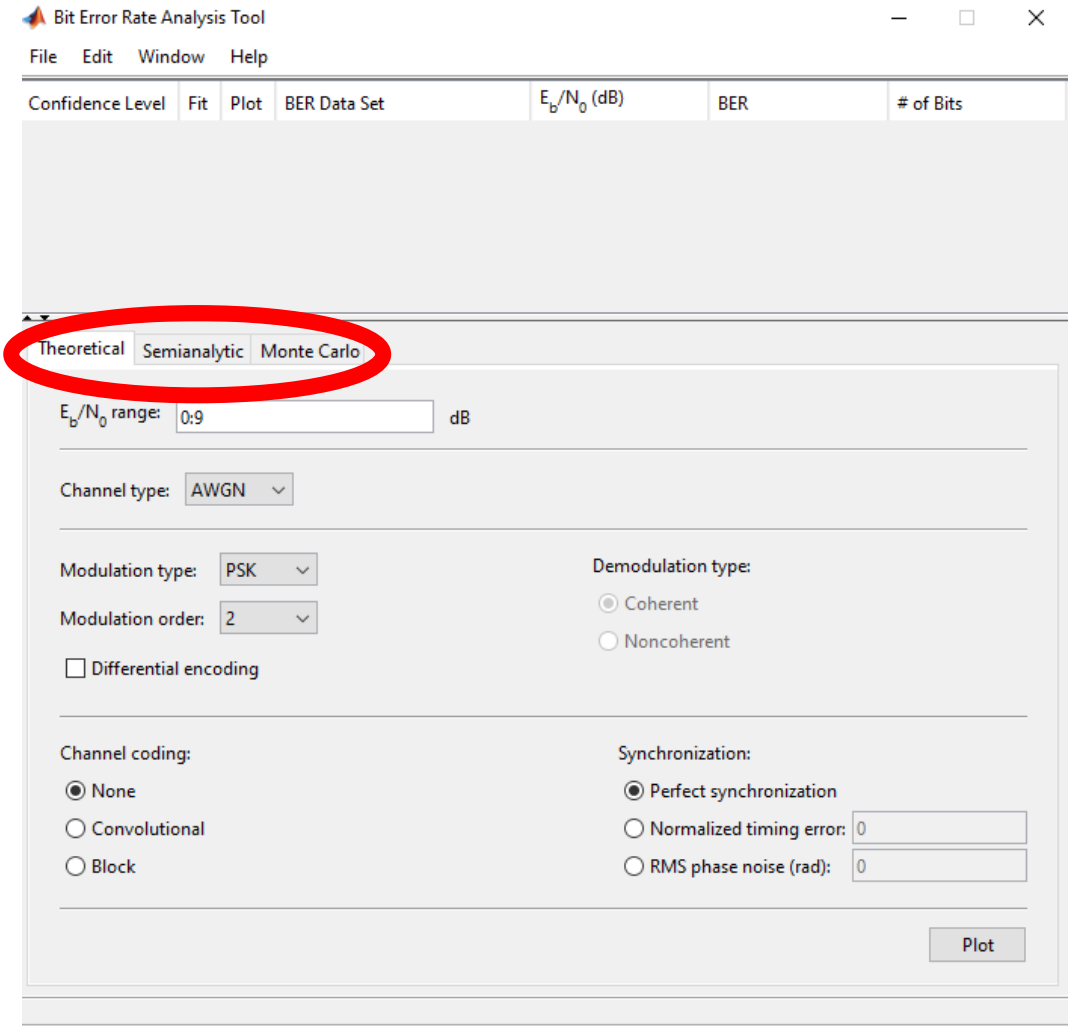

Figure 5 - Bit Error Rate Tool

The three tabs circled in Figure 5 correspond to the different methods the BERTool can use to generate data. We will primarily focus on the **Monte Carlo** tab, which enables us to link it to Simulink models. However, we will also look at theoretical results which we can use for comparative purposes.

Click on the **Monte Carlo** tab, as shown in Figure 6.

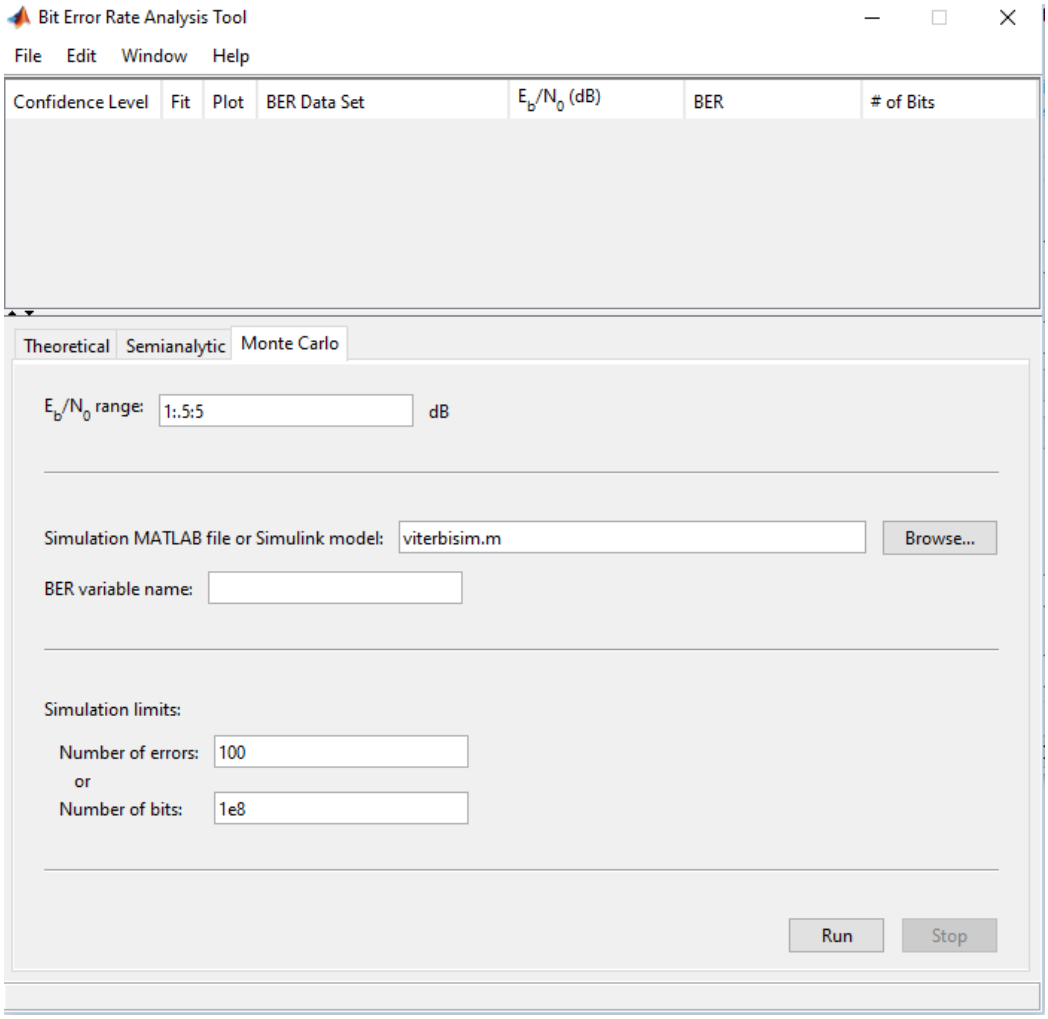

Figure 6 - Monte Carlo

Configure the following:

- The Eb/No range specifies the signal to noise ratio and, if necessary, the interval at which the noise should increase. It should be entered in the format **Signal:Interval:Noise**. So for example, the default value shown in Figure 6, **1:.5:5**, would be a signal value of **1** and the noise level would be raised to **5** in intervals of **0.5**.

  In this case, we will start by setting a simple value of **0:9** without worrying about the interval. By default, it will increase in intervals of 1 db.
- You will then need to select the Simulink model you wish to use. Do this by clicking the **Browse** button and locating the file as normal.
- Enter the *BER variable name* as the one that you configured in the signal to workspace block in your Simulink model. In this case, it was **BER**.
- Leave *Simulation limits* as the default for now.

Once you have done the previous steps your BERTool settings should reflect Figure 7

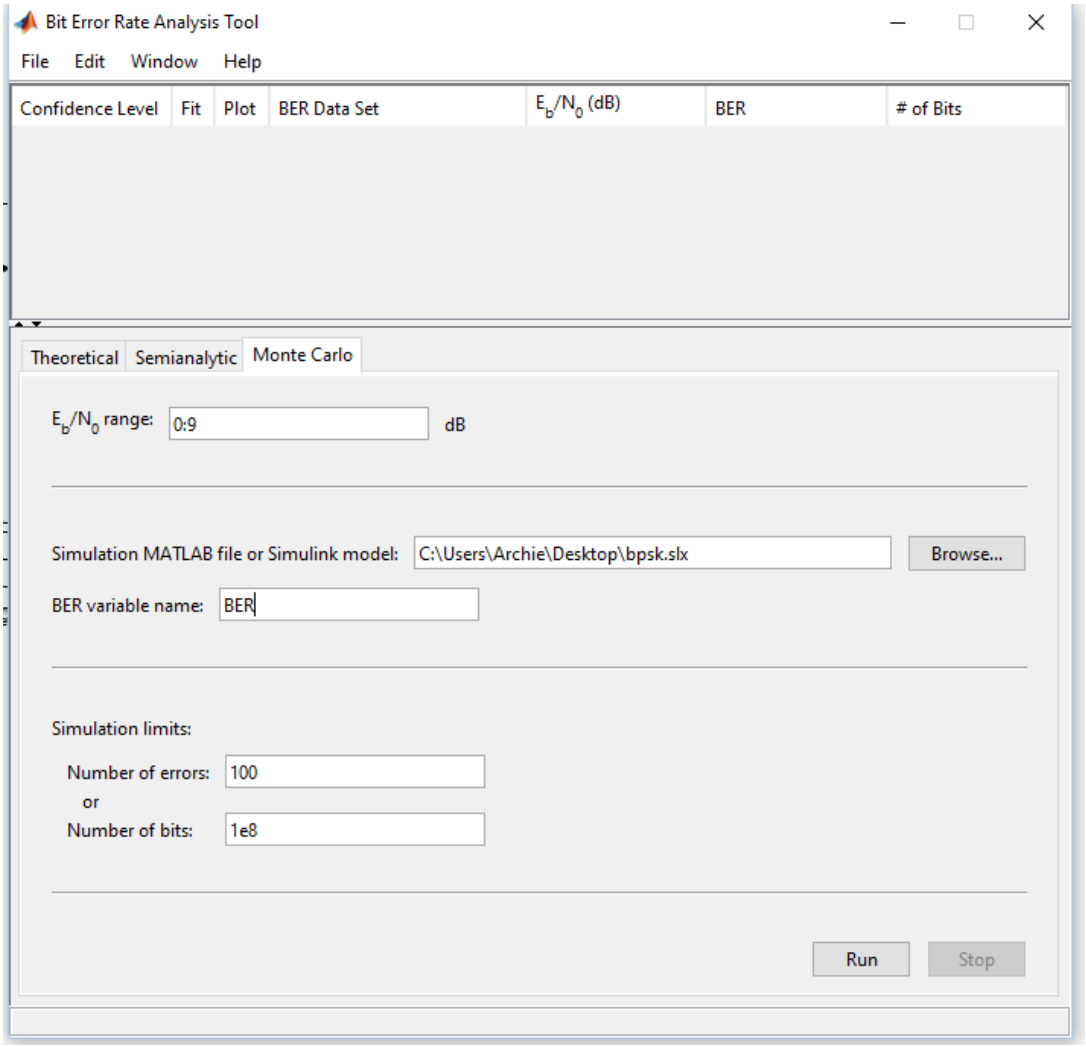

Figure 7 - BERTool - Ready to Run

Click **Run**.

This will bring up a *BER Figure* window that will plot the results for you as the simulation is being run. If you observe the lower left corner of the BERTool window you will be able to see the noise levels increasing. In this case, it will go up to 9 db as we specified.

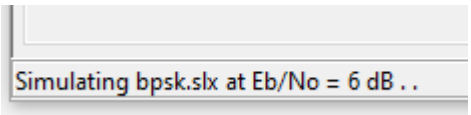

Figure 8 - Noise Levels

The simulation will continue until the specified number of errors are reached, in this case **100**. Note that this might take a while.

Once the simulation is complete, the graph should look something like what is shown in Figure 9 (colours/shapes will vary).

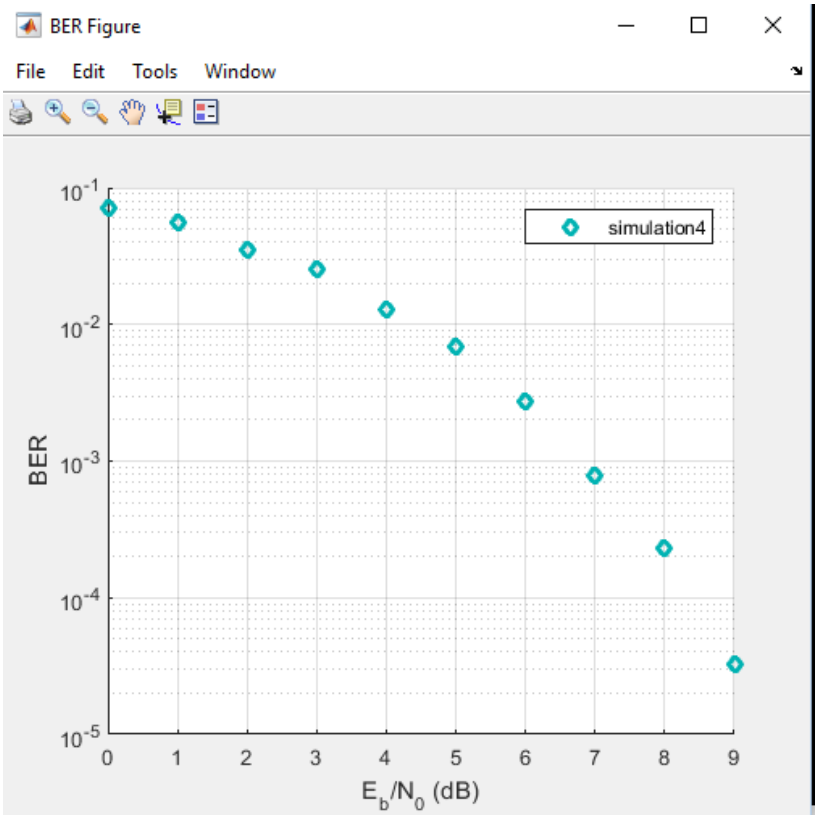

Figure 9 - Graphical Results

Additionally, you will see the simulation results appear on the list at the top of the BERTool window.

Confidence Level	Fit	Plot	BER Data Set	E_b/N_0 (dB)	BER	# of Bits
off	☐	☑	simulation4	[0 1 2 3 4 5 6 7 8 9]	[0.0717 0.0557 0.03...	[1394 1795 2877 39...

Figure 10 - Plotting Options

Check the **Fit** box to fit a curve for the points that have been plotted. Additionally, under *BER data set* it's a good idea to give it a meaningful name so that the results can be understood later. Your graph should then reflect Figure 11.

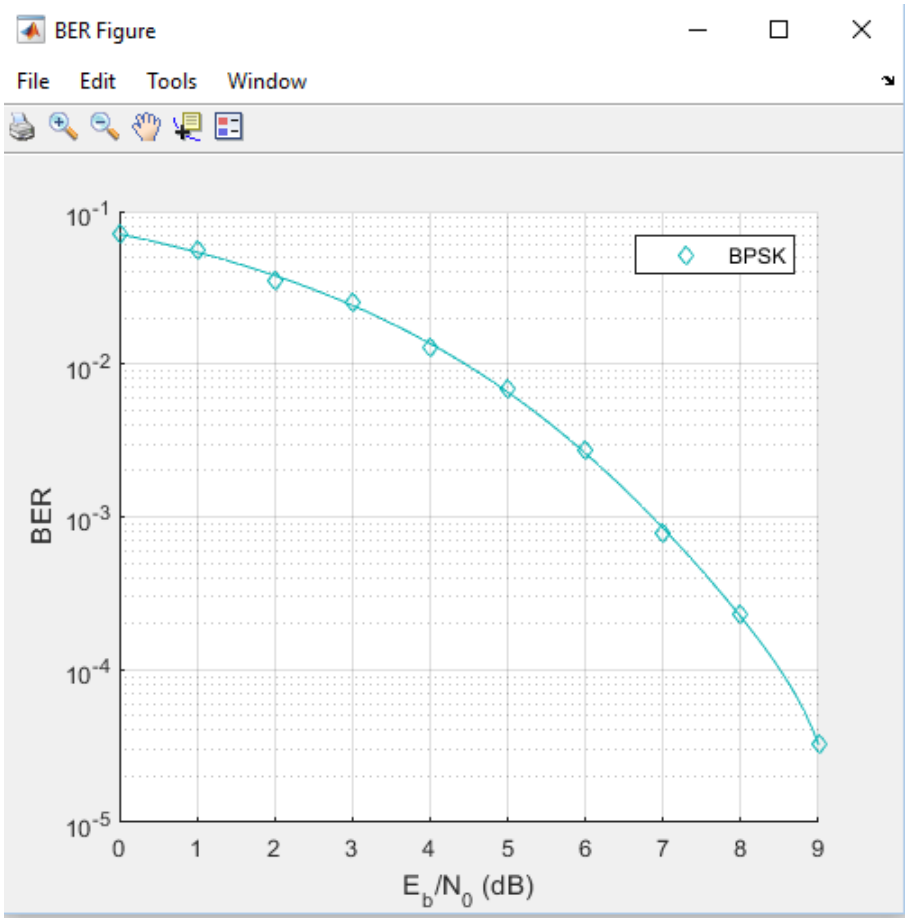

Figure 11 - Results with curve fitted

Comparing actual results with theoretical results

The BERTool allows us to use theoretical results to compare with the results obtained from our own models. We can plot both on the same graph and observe the differences.

To try this out, go to the *theoretical* tab in the BERTool (as opposed to Monte Carlo).

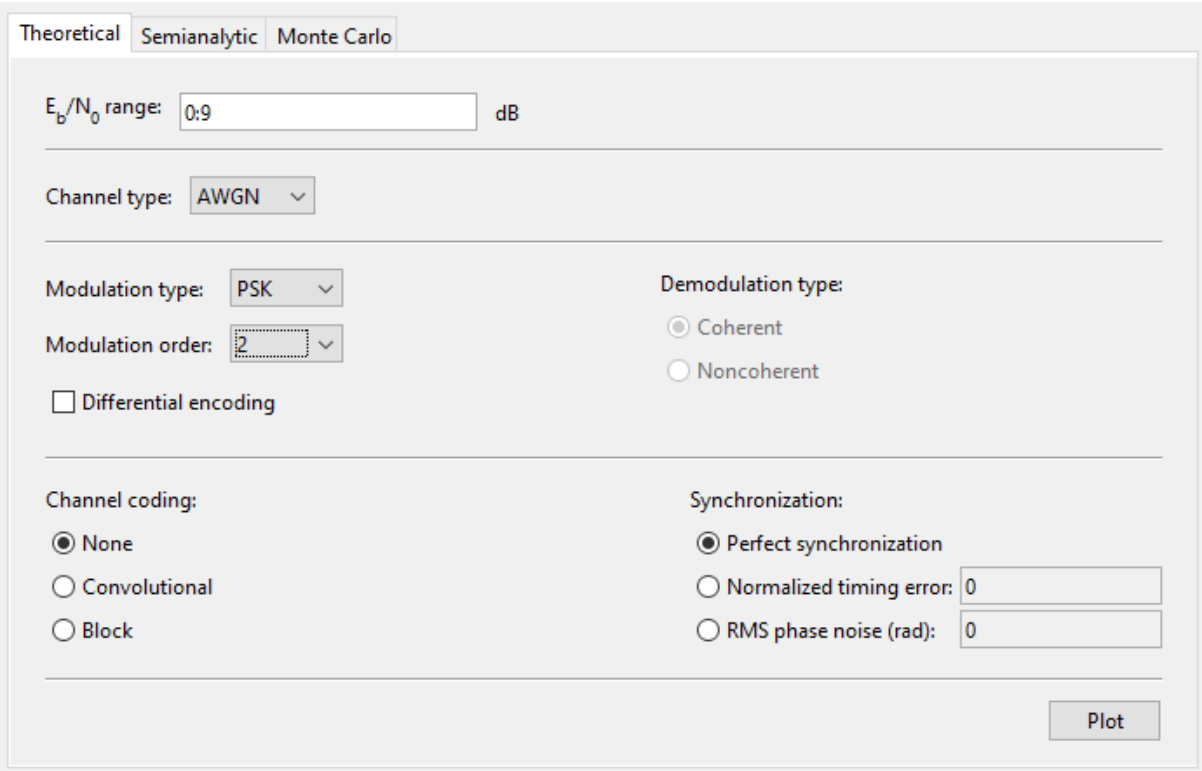

Figure 12 - Gathering theoretical results

As shown in Figure 12, configure the simulation parameters to be the same as the one used for the Monte Carlo simulations.

- EbNo range – **0:9**
- Channel type – **AWGN**
- Modulation type – **PSK** with a modulation order of **2** (i.e. BPSK)

Click **Plot**, and the theoretical value curve should then appear on your graph, as shown in Figure 13, as well as in the list of results (refer to Figure 10). Again, it's a good idea to give it a meaningful name.

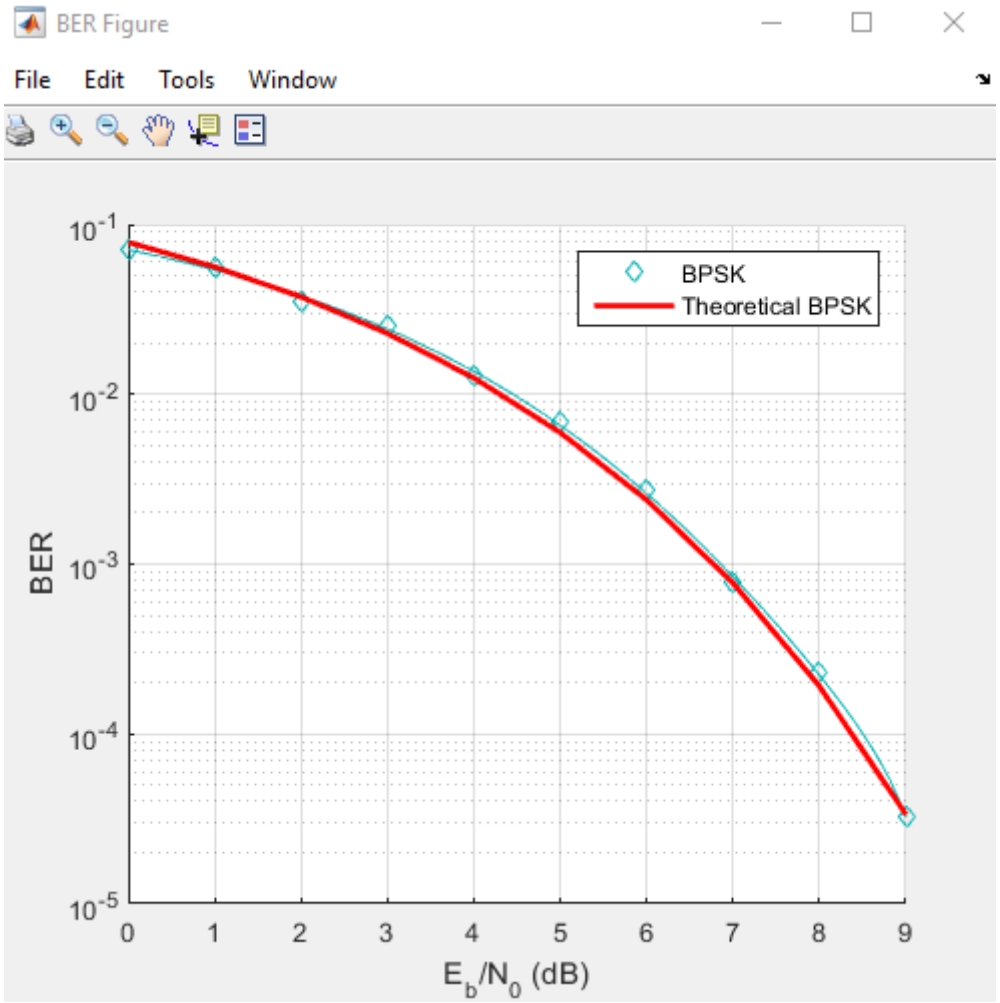

Figure 13 - Theoretical and actual results

Plotting results for other modulation schemes

You can now repeat these steps for your other simulation models, i.e. QPSK and QAM. This should ultimately give you something like this:

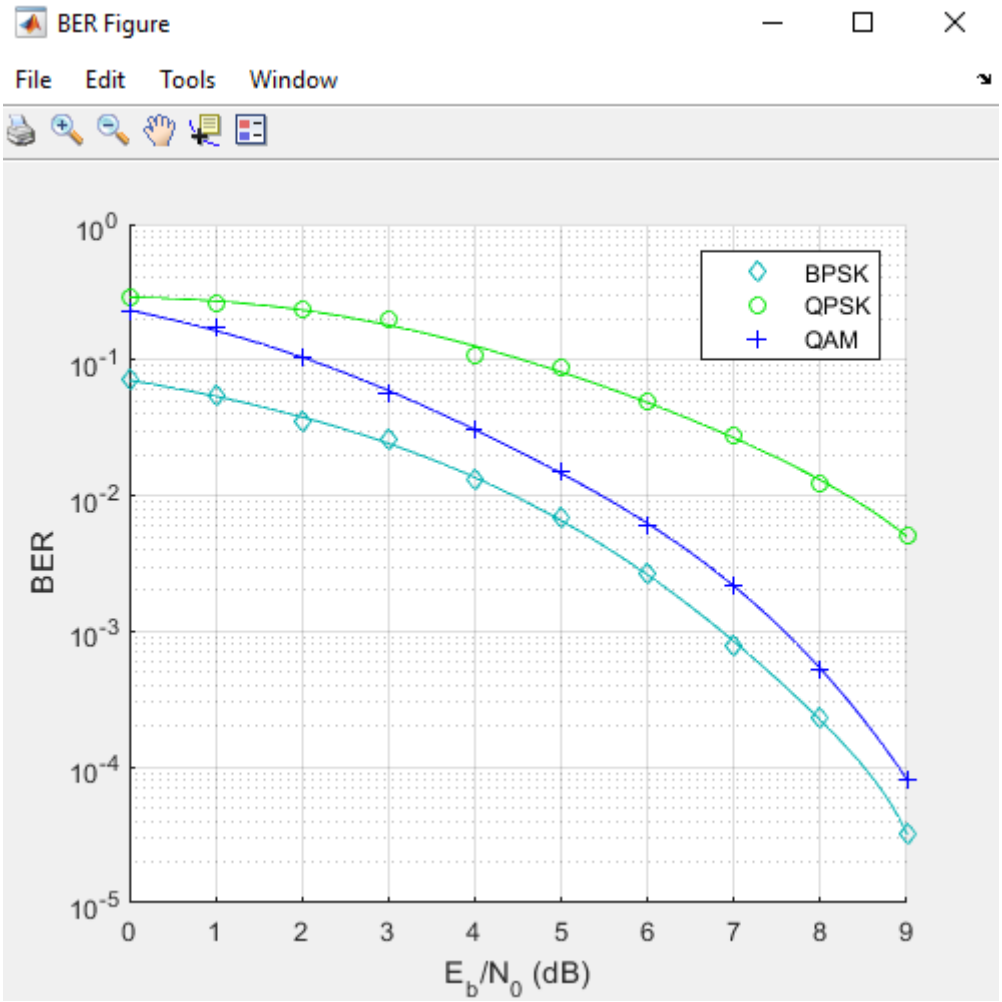

Figure 14 - Comparison of Modulation Schemes