Staffordshire
UNIVERSITY

Matlab, simulink

Building a Channel Noise Model

&

Reducing the Error Rate Using a Hamming Code

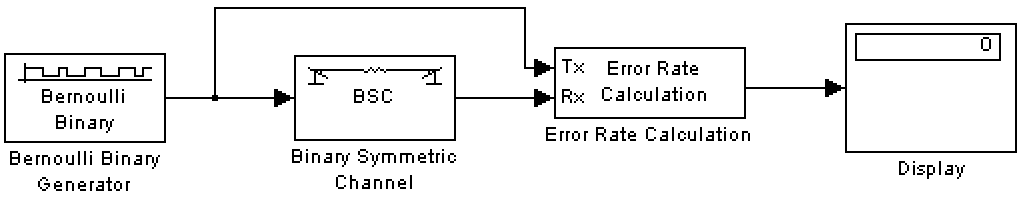
Satellite Communications
CE74027-3

Tutor: A. El-HELW
Email: elhelw@staffs.ac.uk

Building a Channel Noise Model

This section shows how to build a simple model of a communication system. The model, shown in the following figure, contains the most basic elements of a communication system: a source for the signal, a channel with noise, and means of detecting errors caused by noise.

Channel Noise Model



We encourage you to build the model for yourself, as this is the best way to learn how to use the Communications Blockset.

The topics in this section are as follows:

- Overview of the Model
- Selecting Blocks for the Channel Noise Model
- Setting Parameters in the Channel Noise Model
- Connecting the Blocks
- Running the Channel Noise Model

Overview of the Model

The channel noise model generates a random binary signal, and then switches the symbols 0 and 1 in the signal, according to a specified error probability, to simulate a channel with noise. The model then calculates the error rate and displays the result. The model contains the following components.

Source

The source for the signal in this model is the Bernoulli Binary Generator block, which generates a random binary sequence.

The Channel

The Binary Symmetric Channel block simulates a channel with noise. The block introduces random errors to the signal by changing a 0 to a 1 or the reverse, with a probability specified by the **Error probability** parameter in the block's dialog.

Error Rate Calculation

The Error Rate Calculation block calculates the error rate of the channel. The block has two input ports, labeled `Tx`, for the transmitted signal, and `Rx`, for the received signal. The block compares the two signals and checks for errors. The output of the block is a vector with three entries:

- Bit error rate, which you expect to be approximately .01, since this is the probability of error in the channel
- Number of errors
- Total number of bits that are transmitted

Display

The Display block displays the output of the Error Rate Calculation block, as described in Displaying the Error Rate.

Selecting Blocks for the Channel Noise Model

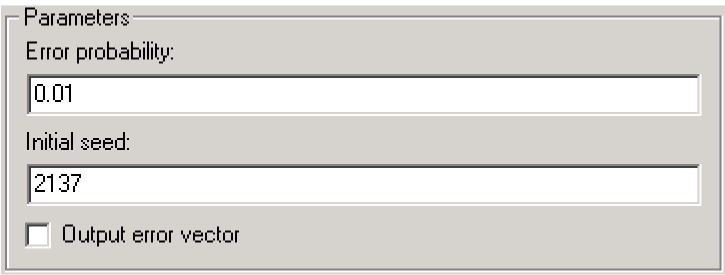
To build the model, first move its blocks into a new model window, as follows:

1. Type `commstartup` at the MATLAB prompt to set simulation parameters for the model.
2. Type `simulink` at the MATLAB prompt to open the Simulink Library Browser.
3. From the **File** menu, select **New**, and then **Model**. This opens a new model window.
4. Drag the following blocks from the Simulink Library Browser into the model window:
    - Bernoulli Binary Generator block, from the Data Sources sublibrary of the Comm Sources library
    - Binary Symmetric Channel block, from the Channels library
    - Error Rate Calculation block, from the Comm Sinks library
    - Display block, from the Simulink Sinks library

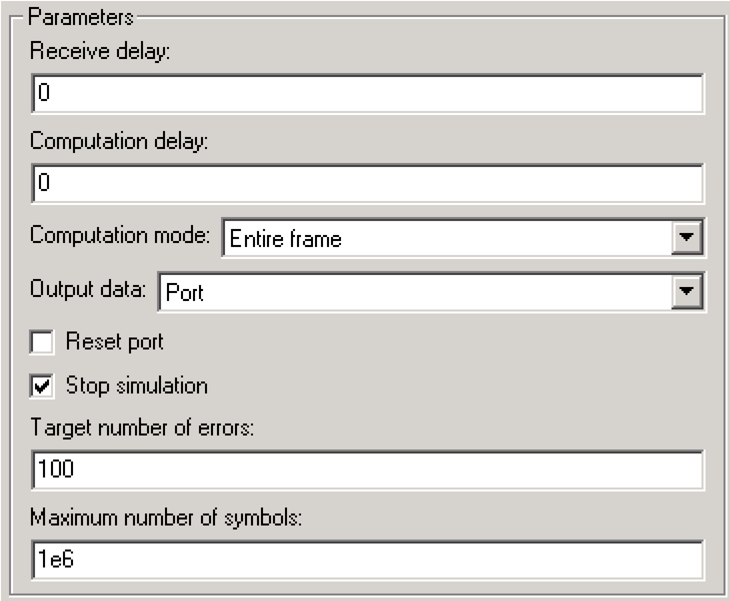
Setting Parameters in the Channel Noise Model

To set block parameters in the channel noise model, do the following:

1. Double-click the Binary Symmetric Channel block and make the following changes to the default parameters in the block's dialog:
    - Set **Error probability** to `0.01`.
    - Clear the **Output error vector** box. This removes the block's lower output port, which is not needed for this model.



2. Double-click the Error Rate Calculation block and make the following changes to the default parameters in the block's dialog:
    - Set **Output data** to `Port` to create an output port for the block.
    - Select **Stop simulation**.



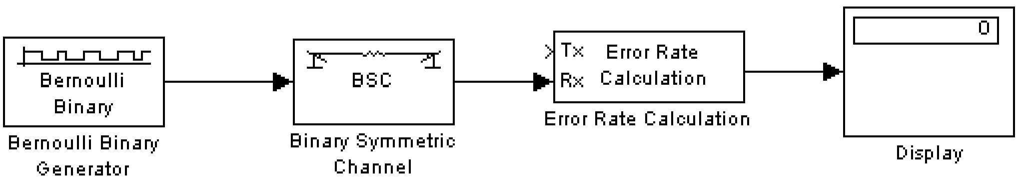
Selecting **Stop simulation** causes the simulation to stop after the target number of errors occurs or the maximum number of symbols is reached.

Initial Seeds

The Bernoulli Binary Generator block and the Binary Symmetric Channel block both use a random number generator to generate random sequences of bits. In both blocks, the **Initial seed** parameter initializes the random sequence. The initial seeds in the two blocks should have different values to ensure that the source signal and the channel noise are statistically independent. In general, initial seeds should have a different values in all blocks that have an **Initial seed** parameter.

Connecting the Blocks

Next, connect the blocks as shown in the following figure. Make sure to connect the arrow from the Binary Symmetric Channel block to the input port labeled `Rx` on the Error Rate Calculation block. To learn how to do this, see Connecting Blocks.



The next section explains how to draw the upper branch line in the model.

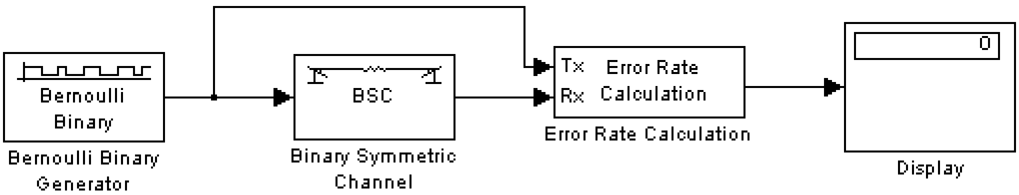
Drawing a Branch Line

The upper line leading from the Bernoulli Binary Generator block to the Error Rate Calculation block, shown in the following figure, is called a *branch line*. Branch lines carry the same signal to more than one block.

To draw the branch line, follow these steps:

1. Right-click the line between the Bernoulli Random Generator block and the Binary Symmetric Channel block.
2. Move the mouse pointer to the input port labeled `Tx` on the Error Rate Calculation block, while pressing the right mouse button.
3. Release the mouse button. The end of the branch line should connect to the input port of the Error Rate Calculation block.
4. Click the horizontal section of the branch line and drag it upward until the line is above the Binary Symmetric Channel block.

The model should now appear as in the following figure.

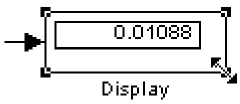


Running the Channel Noise Model

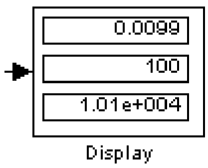
To run the model, select **Start** from the **Simulation** menu. After a few seconds, the model will stop automatically.

To see all three boxes in the Display block, you must enlarge the block slightly, as follows:

1. Select the Display block and move the mouse pointer to one of the lower corners of the block, so that a diagonal arrow appears on the corner, as shown.



2. Drag the corner of the block down with the mouse until three windows appear, as shown.



The Display block displays the following information:

- The bit error rate
- The number of errors
- The total number of bits that are transmitted

Note that the exact values that appear will vary, depending on the **Initial seed** parameters in the Bernoulli Binary Generator block and the Binary Symmetric Channel block.

Since the **Target number of errors** in the dialog for the Error Rate Calculation block is set to `100`, the simulation stops when 100 errors have been detected.

To save the model, select **Save** from the **File** menu, type a name for the model, such as `channelnoise`, in the **File name** field, and click **Save**.

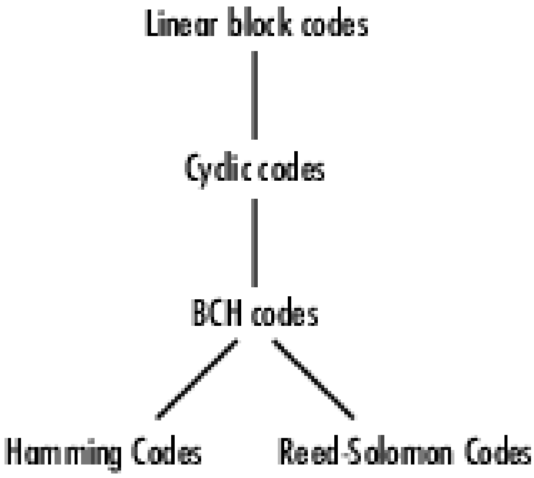
Reducing the Error Rate Using a Hamming Code

This section describes how to reduce the error rate in the model shown in the figure Channel Noise Model by adding an error-correcting code.

Block Coding

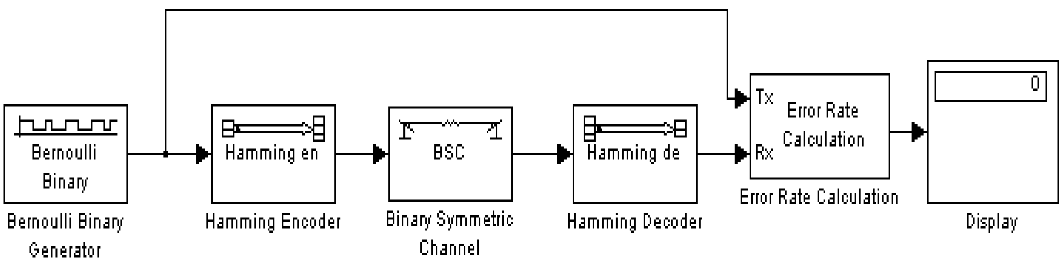
Error-control coding techniques detect and possibly correct errors that occur when messages are transmitted in a digital communication system. To accomplish this, the encoder transmits not only the information symbols but also extra redundant symbols. The decoder interprets what it receives, using the redundant symbols to detect and possibly correct whatever errors occurred during transmission. You might use error-control coding if your transmission channel is very noisy or if your data is very sensitive to noise. Depending on the nature of the data or noise, you might choose a specific type of error-control coding.
Block coding is a special case of error-control coding. Block coding techniques maps a fixed number of message symbols to a fixed number of code symbols. A block coder treats each block of data independently and is a memoryless device.

The class of block coding techniques includes categories shown in the diagram below.



The Communications Blockset supports general linear block codes. It also includes blocks that process cyclic, BCH, Hamming, and Reed-Solomon codes (which are all special kinds of linear block codes). Blocks in the blockset can encode or decode a message using one of the techniques mentioned above. The Reed-Solomon and BCH decoders indicate how many errors they detected while decoding. The Reed-Solomon coding blocks also let you decide whether to use symbols or bits as your data.

Hamming Code Model



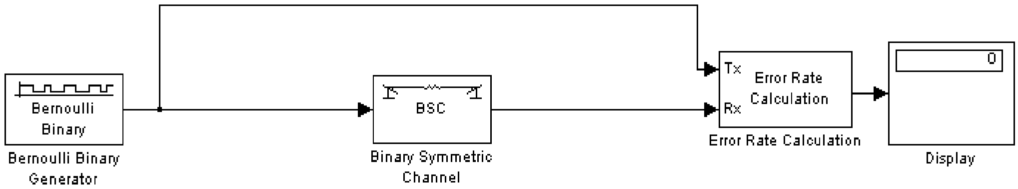
The topics in this section are as follows:

- Building the Hamming Code Model
- Using the Hamming Encoder and Decoder Blocks
- Setting Parameters in the Hamming Code Model
- Labeling the Display Block
- Running the Hamming Code Model
- Displaying Frame Sizes
- Adding a Scope to the Model
- Setting Parameters in the Expanded Model
- Observing Channel Errors with the Scope

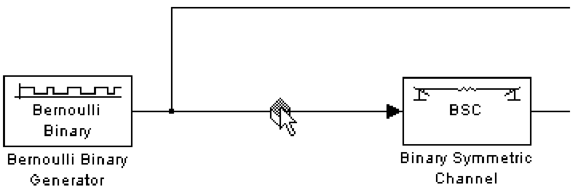
Building the Hamming Code Model

You can build the Hamming code model by adding blocks to the model shown in the figure Channel Noise Model. To do so, follow these steps:

1. Type `channeldoc` at the MATLAB prompt to open the channel noise model. Then save the model as `my_hamming` in the directory where you keep your work files. See Saving a Model.
2. Drag the following two Communications Blockset blocks from the Simulink Library Browser into the model window:
    - Hamming Encoder block, from the Block sublibrary of the Error Detection and Correction library
    - Hamming Decoder block, from the Block sublibrary of the Error Detection and Correction library
3. Click the right border of the model and drag it to the right to widen the model window.
4. Move the Binary Symmetric Channel block, the Error Rate Calculation block, and the Display block to the right by clicking and dragging. This creates more space between the Binary Symmetric Channel block and the blocks next to it. The model should now look like the following figure.



5. Click the Hamming Encoder block and drag it on top of the line between the Bernoulli Binary Generator block and the Binary Symmetric Channel block, to the right of the branch point, as shown in the following figure. Then release the mouse button. The Hamming Encoder block should automatically connect to the line from the Bernoulli Binary Generator block to the Binary Symmetric Channel block.



6. Click the Hamming Decoder block and drag it on top of the line between the Binary Symmetric Channel block and the Error Rate Calculation block.

Using the Hamming Encoder and Decoder Blocks

The Hamming Encoder block encodes the data before it is sent through the channel. The default code is the [7,4] Hamming code, which encodes message words of length 4 into codewords of length 7. As a result, the block converts frames of size 4 into frames of size 7. The code can correct one error in each transmitted codeword.

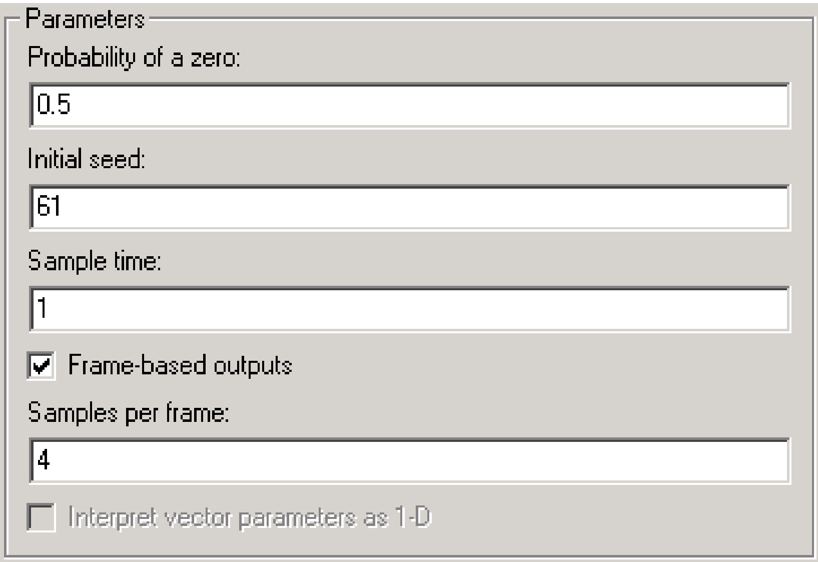
For an [n,k] code, the input to the Hamming Encoder block must consist of vectors of size k. In this example, k=4.

The Hamming Decoder block decodes the data after it is sent through the channel. If at most one error is created in a codeword by the channel, the block decodes the word correctly. However, if more than one error occurs, the Hamming Decoder block might decode incorrectly.

Setting Parameters in the Hamming Code Model

Double-click the Bernoulli Binary Generator block and make the following changes to the parameter settings in the block's dialog, as shown in the following figure:

1. Select the box next to **Frame-based outputs** in the dialog for the Bernoulli Binary Generator block.
2. Set **Samples per frame** to `4`. This converts the output of the block into frames of size 4, in order to meet the input requirement of the Hamming Encoder Block. See [Frames and Frame-Based Processing] for more information about frames.



> **Note** Many Communications Blockset blocks, such as the Hamming Encoder block, require their input to be a vector of a specific size. If you connect a source block, such as the Bernoulli Binary Generator block, to one of these blocks, you should select the box next to **Frame-based outputs** in the dialog for the source, and set **Samples per frame** to the required value.

Labeling the Display Block

You can change the label that appears below a block to make it more informative. For example, to change the label below the Display block to "Error Rate Display," first select the label with the mouse. This causes a box to appear around the text. Then enter the changes to the text in the box.

Running the Hamming Code Model

To run the model, select **Start** from the **Simulation** menu. The model terminates after 100 errors occur. The error rate, displayed in the top window of the Display block, is approximately .001. Note that you get slightly different results if you change the **Initial seed** parameters in the model or run a simulation for a different length of time.

You expect an error rate of approximately .001 for the following reason. The probability of two or more errors occurring in a codeword of length 7 is

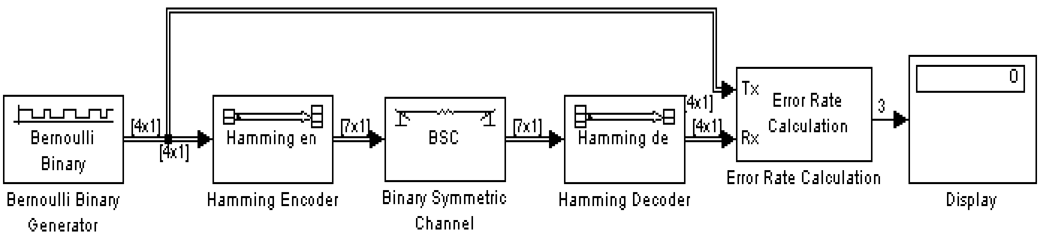
$$1 - (0.99)^7 - 7(0.99)^6(0.01) = 0.002$$

If the codewords with two or more errors were decoded randomly, you would expect about half the bits in the decoded message words to be incorrect. This indicates that .001 is a reasonable value for the bit error rate.

To obtain a lower error rate for the same probability of error, you can try using a Hamming code with larger parameters. To do this, change the parameters **Codeword length** and **Message length** in the Hamming Encoder and Decoder block dialogs. You also have to make the appropriate changes to the parameters of the Bernoulli Binary Generator block and the Binary Symmetric Channel block.

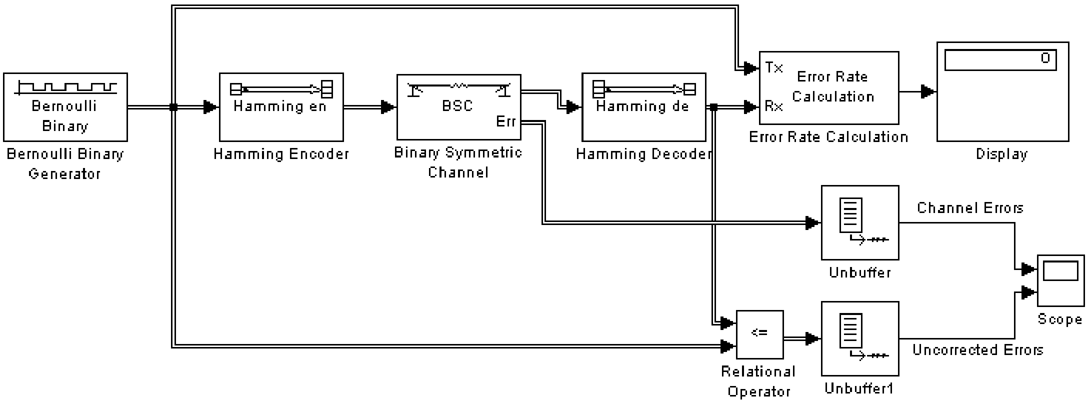
Displaying Frame Sizes

You can display the sizes of data frames in different parts of the model by selecting **Signal dimensions** from the **Port/signal displays** submenu of the **Format** menu at the top of the model window. This is shown in the following figure. Note that the line leading out of the Bernoulli Binary Generator block is labeled `[4x1]`, indicating that its output consists of column vectors of size 4. Since the Hamming Encoder block uses a [7,4] code, it converts frames of size 4 into frames of size 7, so its output is labeled `[7x1]`.



Adding a Scope to the Model

To display the channel errors produced by the Binary Symmetric Channel block, you can add a Scope block to the model. This is a good way to see whether your model is functioning correctly. The example shown in the following figure shows where to insert the Scope block into the model.



To build this model from the one shown in the figure Hamming Code Model, follow these steps:

1. Drag the following blocks from the Simulink Library Browser into the model window:
   - Relational Operator block from the Simulink Logic and Bit Operations library
   - Scope block from the Simulink Sinks library
   - Two copies of the Unbuffer block from the Buffers sublibrary of the Signal Processing Blockset Signal Management library
2. Double-click the Binary Symmetric Channel block to open its dialog, and select **Output error vector**. This creates a second output port for the block, which carries the error vector.
3. Double-click the Scope block and click the **Parameters** button on the toolbar. Set **Number of axes** to `2` and click **OK**.
4. Connect the blocks as shown in the preceding figure.

Setting Parameters in the Expanded Model

Make the following changes to the parameters for the blocks you added to the model.

Error Rate Calculation Block

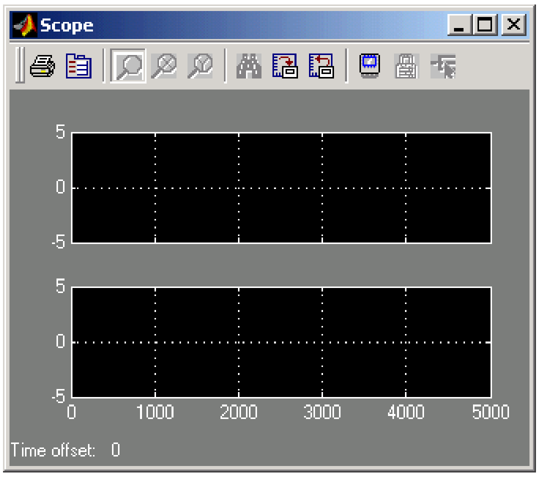
Double-click the Error Rate Calculation block and clear the box next to **Stop simulation** in the block's dialog.

Scope Block

The Scope block displays the channel errors and uncorrected errors. To configure the block:

1. Double-click the block to open the scope, if it is not already open.
2. Click the **Parameters** button on the toolbar.
3. Set **Time range** to `5000`.
4. Click the **Data history** tab.
5. Type `30000` in the **Limit data points to last** field.
6. Click **OK**.

The scope should now appear as shown.



To configure the axes, follow these steps:

1. Right-click the vertical axis at the left side of the upper scope.
2. In the context menu, select **Axes properties**.
3. In the **Y-min** field type `-1`.
4. In the **Y-max** field type `2`.
5. Click **OK**.
6. Repeat the same steps for the vertical axis of the lower scope.
7. Widen the scope window until it is roughly three times as wide as it is high. You can do this by clicking the right border of the window and dragging the border to the right, while pressing the mouse button.

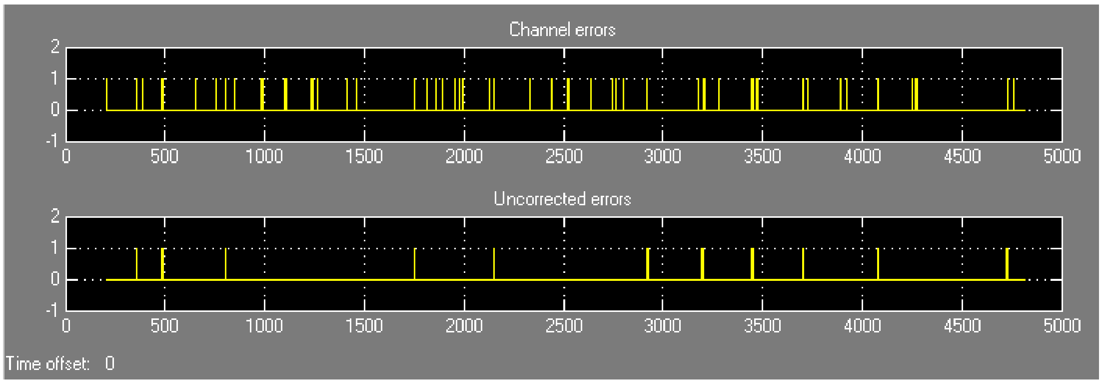
Relational Operator

Set **Relational Operator** to `~=` in the block's dialog. The Relational Operator block compares the transmitted signal, coming from the Bernoulli Random Generator block, with the received signal, coming from the Hamming Decoder block. The block outputs a 0 when the two signals agree and a 1 when they disagree.

Observing Channel Errors with the Scope

When you run the model, the Scope block displays the error data. At the end of each five thousand time steps, the scope appears as shown in the following figure. The scope then clears the displayed data and displays the next five thousand data points.

Scope with Model Running



The upper scope shows the channel errors generated by the Binary Symmetric Channel block. The lower scope shows errors that are not corrected by channel coding.

Click the **Stop** button on the toolbar at the top of the model window to stop the scope.