Staffordshire
UNIVERSITY

Matlab, simulink

Digital Modulation

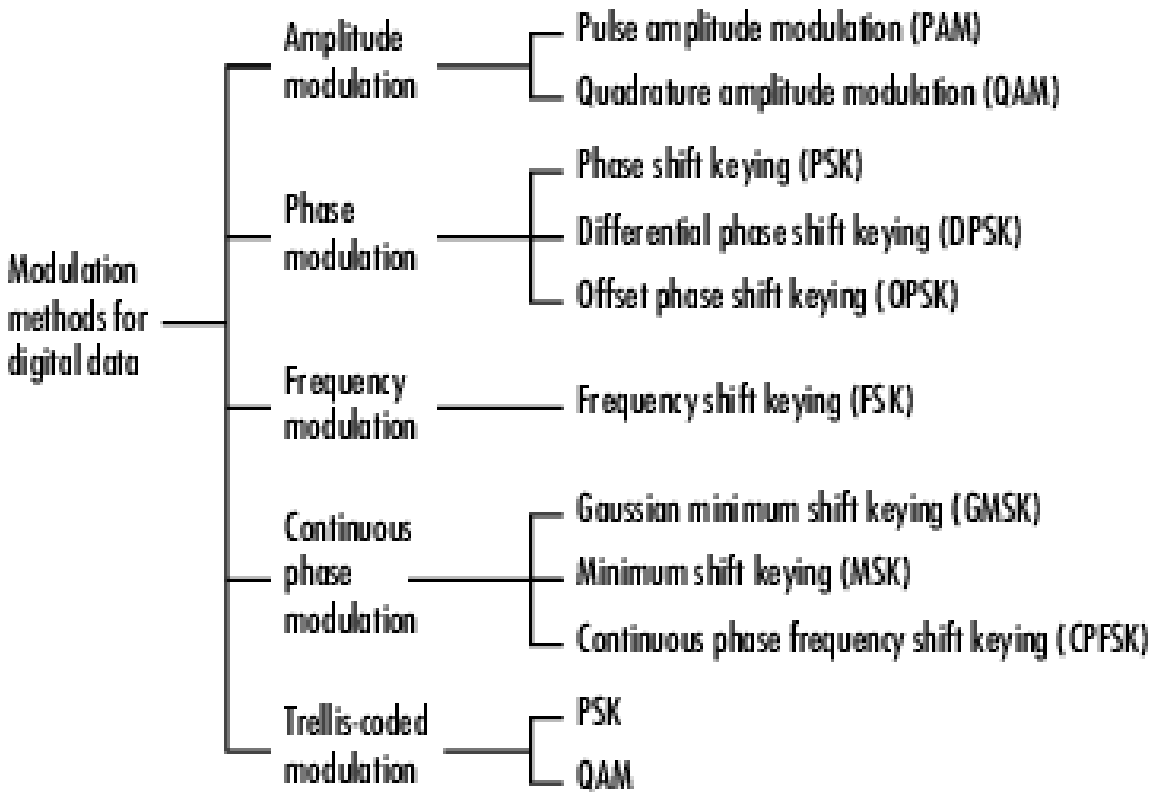
Satellite Communications
CE74027-3

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

Digital Modulation

Digital Modulation Features of the Blockset

The figure below shows the modulation techniques that the Communications Blockset supports for digital data. All of the methods at the far right are implemented in library blocks.



General and Specific Modulation Methods

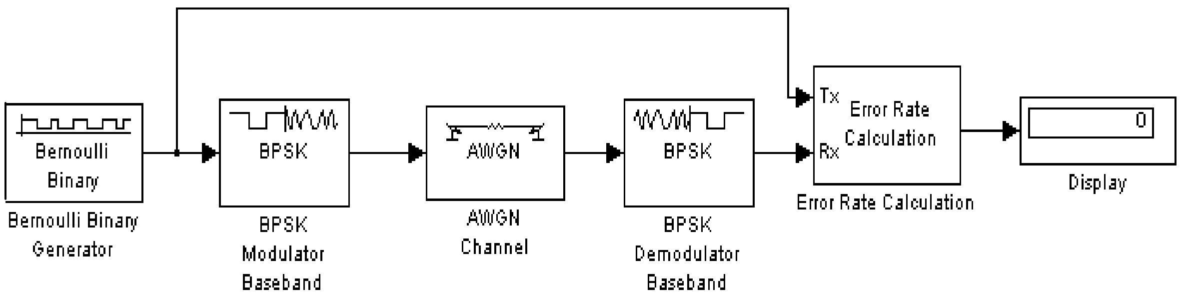
Some digital modulation sublibraries contain blocks that implement special cases of a more general technique and are, in fact, special cases of a more general block. These special-case blocks use the same computational code that their general counterparts use, but provide an interface that is either simpler or more suitable for the special case. The table below lists special-case modulators, their general counterparts, and the conditions under which the two are equivalent. The situation is analogous for demodulators.

General and Specific Blocks

General Modulator	Specific Modulator	Specific Conditions
General QAM Modulator Baseband	Rectangular QAM Modulator Baseband	Predefined constellation containing 2^K points on a rectangular lattice
M-PSK Modulator Baseband	BPSK Modulator Baseband	**M-ary number** parameter is 2.
	QPSK Modulator Baseband	**M-ary number** parameter is 4.
M-DPSK Modulator Baseband	DBPSK Modulator Baseband	**M-ary number** parameter is 2.
	DQPSK Modulator Baseband	**M-ary number** parameter is 4.
CPM Modulator Baseband	GMSK Modulator Baseband	**M-ary number** parameter is 2, **Frequency pulse shape** parameter is `Gaussian`.
	MSK Modulator Baseband	**M-ary number** parameter is 2, **Frequency pulse shape** parameter is `Rectangular`, **Pulse length** parameter is 1.
	CPFSK Modulator Baseband	**Frequency pulse shape** parameter is `Rectangular`, **Pulse length** parameter is 1.
General TCM Encoder	Rectangular QAM TCM Encoder	Predefined signal constellation containing 2^K points on a rectangular lattice
	M-PSK TCM Encoder	Predefined signal constellation containing 2^K points on a circle

Modelling a Channel with Modulation

The Binary Symmetric Channel block, which simulates a channel with noise, is useful for building models of channel coding. For other types of applications, you might want to construct a more realistic model of a channel. For example, you can add modulation and demodulation, and replace the Binary Symmetric Channel block with an AWGN Channel block, which adds white Gaussian noise to the channel. This following figure shows an example that uses binary phase shift keying (BPSK).



BPSK Modulation Model

The topics in this section are as follows:

- Building the BPSK Model.
- Setting Parameters in the BPSK Model.
- Running the BPSK Model.

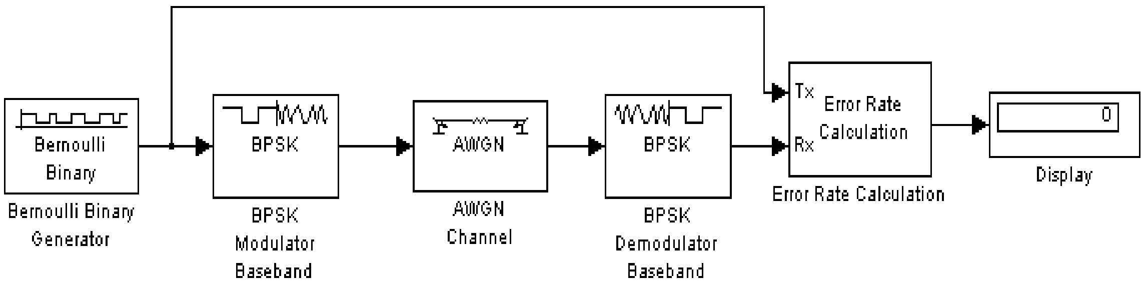
We encourage you to build the model for yourself. Alternatively, to open a completed version of the model, type bpskdoc at the MATLAB prompt.

Building the BPSK Model

You can build the BPSK model from the one shown in the figure Channel Noise Model. To build the model, follow these steps:

1. Enter channeldoc at the MATLAB prompt to open the channel noise model, and then save the model as my_bpsk in the directory where you keep your work files.
2. Delete the Binary Symmetric Channel block from the model by right-clicking the block and selecting Clear.
3. Move the following blocks from the Simulink Library Browser into the model window, and insert them into the model as shown in the followingfigure:
   - BPSK Modulator Baseband block, from PM in the Digital Baseband Modulation sublibrary of the Modulation library
   - AWGN Channel block, from the Channels library
   - BPSK Demodulator Baseband block, from PM in the Digital Baseband Modulation sublibrary of the Modulation library

The model should now appear as in the figure below.



Binary Phase Shift Keying

The BPSK Modulator and Demodulator Baseband blocks implement binary phase shift keying (BPSK) modulation. BPSK is a method for modulating a binary signal onto a complex waveform by shifting the phase of the complex signal. In digital baseband BPSK, the symbols 0 and 1 are modulated to the complex numbers exp(jt) and -exp(jt), respectively, where t is a fixed angle. In this example, t = 0, so these numbers are just 1 and -1.
You can set the value of t in the Phase offset parameter in the dialogs for the BPSK Modulator Baseband block and the BPSK Demodulator Baseband block. The default value is 0.
To learn more about the digital modulation features of the Communications Blockset, see "Digital Modulation" in the online Communications Blockset documentation.

Setting Parameters in the BPSK Model

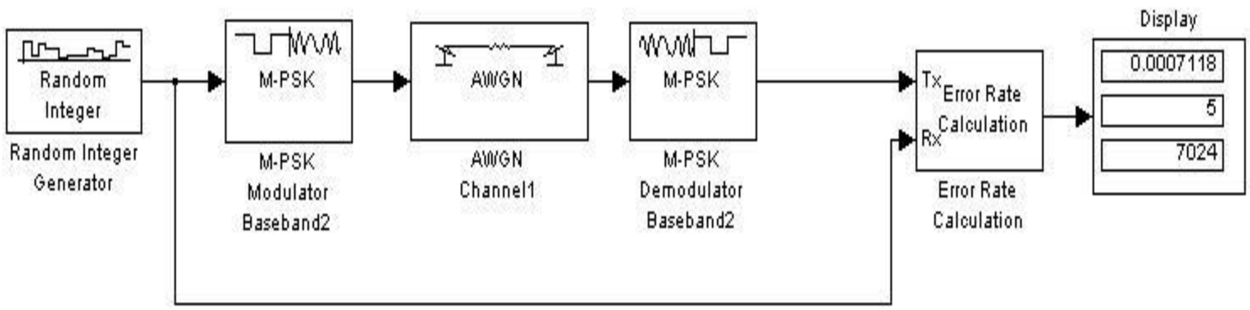
To set block parameters in the BPSK model, do the following:

1. Double-click the AWGN Channel block and set Es/No to 4.2.
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.
   - Check the box next to Stop simulation.

Running the BPSK Model

When you run the model, the Display block shows an error rate of approximately 0.01, the same as in the channel noise model. The BPSK model uses the BPSK Modulator Baseband, the AWGN Channel, and the BPSK. Demodulator Baseband blocks to simulate a channel with noise. This provides a more realistic model of a channel than using just the Binary Symmetric. Channel block. You can also model other types of channel noise using blocks from the Communications Blockset Channels library.

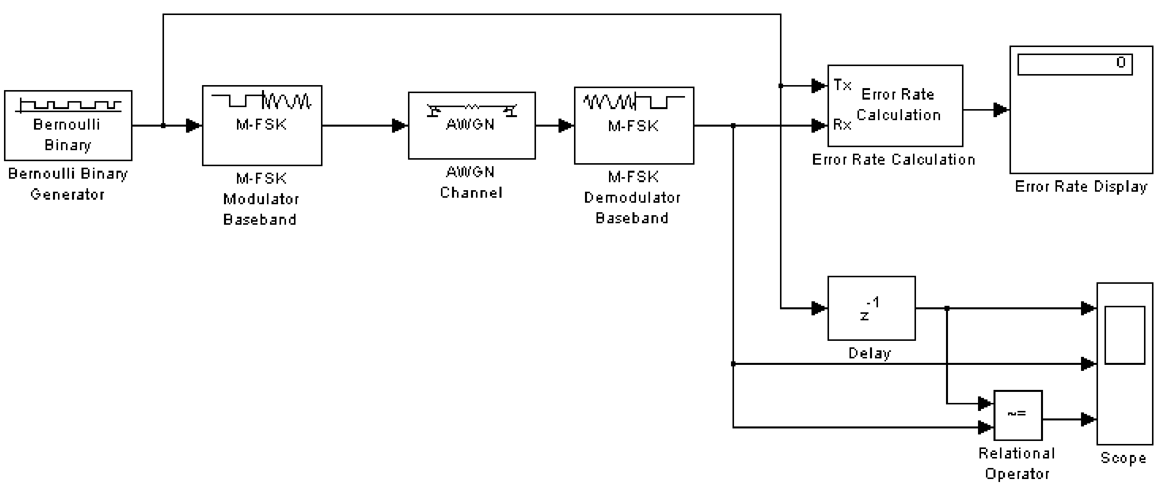
Now try to construct this model



Building a Frequency-Shift Keying Model

Frequency-shift keying (FSK) is a standard modulation technique in which a digital signal is modulated onto a sinusoidal carrier whose frequency shifts between different values. The Bell Telephone System first used this technique in their Model 103 modem. The model shown in the following figure is an example of the baseband representation of FSK.

FSK Model



This section explains how to

- [Building the FSK Model]
- [Setting Parameters in the FSK Model]
- [Running the FSK Model]
- [Learning About Delays in the Model]
- [Finding the Delay in a Model]

Building the FSK Model

You can build the FSK model by adding blocks to the model shown in the figure Channel Noise Model. To open the channel noise model, enter `channeldoc` at the MATLAB prompt. Then save the model as `my_fsk` in the directory where you keep your work files. See Saving a Model.

You need to add the following blocks to the model.

M-FSK Modulator and Demodulator Baseband

The M-FSK Modulator Baseband block, from FM in the Digital Baseband sublibrary of the Modulation library, modulates the binary signal using a baseband representation of FSK modulation.

The M-FSK Demodulator Baseband block, from FM in the Digital Baseband sublibrary of the Modulation library, demodulates the baseband signal.

AWGN Channel

The AWGN Channel block, from the Channels library, models a channel using additive white Gaussian noise. In this model, AWGN is more suitable than a binary symmetric channel.

Relational Operator

The Relational Operator block, from the Simulink Logic and Bit Operations library, compares the transmitted signal, from the Bernoulli Binary Generator block, with the received signal, from the M-FSK Demodulator Baseband block. The block outputs a 0 when the two signals agree, and a 1 when they differ.

Scope

The Scope block, from the Simulink Sinks library, displays the transmitted signal, the received signal, and the output of the Relational Operator block. To create three input ports for the block, follow these steps:

1. Double-click the block to open the scope.
2. Click the **Parameters** button on the toolbar.
3. Set **Number of axes** to `3`.
4. Set **Time range** to `150`.
5. Click **OK**.

To set the limits on the vertical axes:

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**.

Repeat these steps for the middle and lower vertical axes.

Delay

The Delay block, from the Signal Processing Blockset Signal Operations library, delays the transmitted signal so that it can be accurately compared with the received signal. Its purpose is explained further in Learning About Delays in the Model.

Drag these blocks into the model window and connect them as shown in the figure FSK Model. Also, remove the Binary Symmetric Channel block because it is no longer needed. The next section explains how to set the parameters for these blocks.

Error Rate Calculation

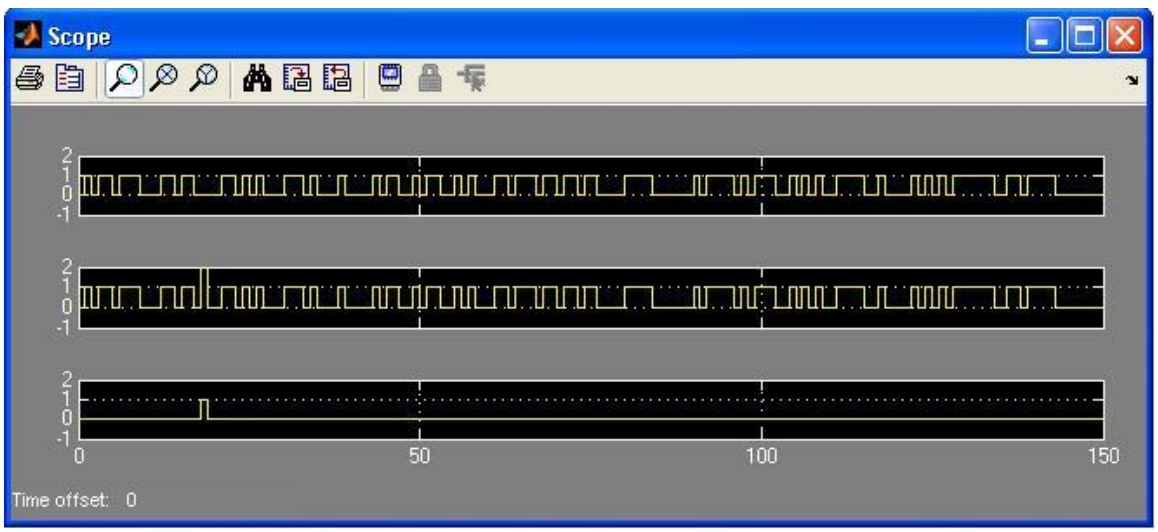
Set the receive delay parameter in the error rate calcuation to 1

Running the FSK Model

Set the **Stop time** parameter to `150` and run the model.

Displaying the Errors in the Scope

Double-click the Scope block to open the scope.



The top window displays the transmitted signal. The middle window displays the received signal. The bottom window displays a 0 where the two signals agree.

Learning About Delays in the Model

Some blocks cause a signal to be delayed as it passes through a model, due to the way they process data. For example, there is a delay of 1 symbol period between the input signal to the M-FSK Modulator Baseband block and the output signal from the M-FSK Demodulator Baseband block. As a result, there is a delay of 1 between the transmitted and received signals in the model. Finding the Delay in a Model shows how to find the value of this delay.

To compare these two signals and calculate the bit error rate correctly, you need to delay the transmitted signal by 1 to synchronize it with the received signal. This is why you set the **Receive delay** to `1` in the dialog for the Error Rate Calculation block, and left the **Delay** at its default value of `1` in the Delay block.

> **Note** If the Error Rate Calculation block in a model gives an error rate close to .5 for a random binary signal, you might not have taken into account delays in the model. The block is probably comparing two unsynchronized signals.

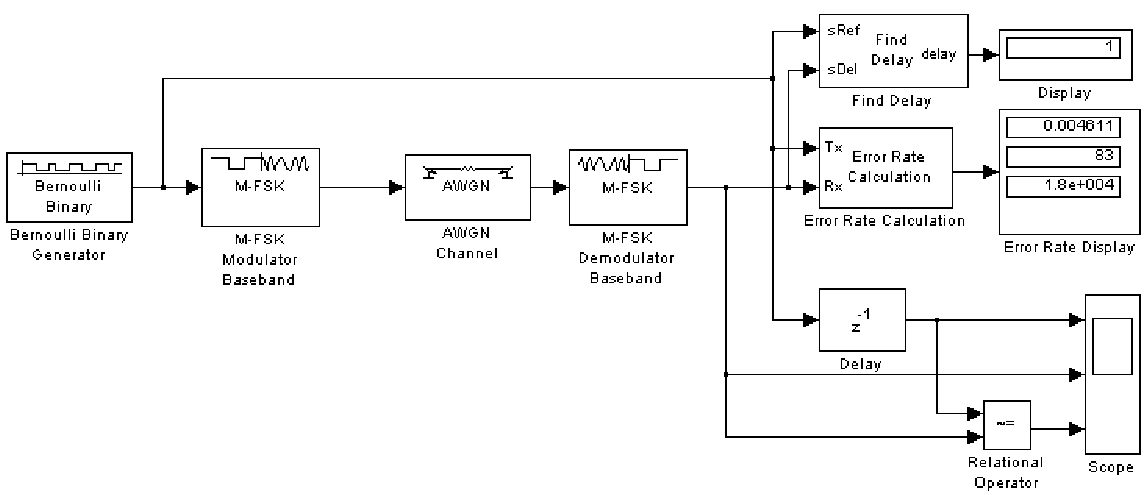
Several blocks in the Communications Blockset Modulation library produce delays. A list of these is given in Delays in Digital Modulation in the online Communications Blockset documentation. The Viterbi Decoder block, from the Convolutional sublibrary of the Error Detection and Correction library, also produces a delay equal to its **Traceback depth** parameter; see the section Viterbi Decoder.

Finding the Delay in a Model

To find the delay between the transmitted and received signals in a model, you can use a Find Delay block. To do so, insert the following blocks into the FSK model:

- Find Delay, from the Utility Blocks library
- Display, from the Simulink Sinks library

Connect the blocks as shown at the top right of the following figure and set the correlation windonw length to 150 in the find delay .



When you run the simulation, the Display block labeled "Display" shows that the delay is 1. You can use this information to set the **Receive delay** parameter in the Error Rate Calculation block, and the **Delay** parameter in the Delay block.