Staffordshire
UNIVERSITY

Matlab, simulink

ADC and DAC

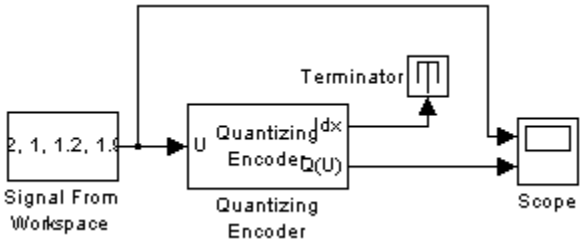
Satellite Communications
CE74027-3
Tutor: A. El-HELW
Email: elhelw@staffs.ac.uk

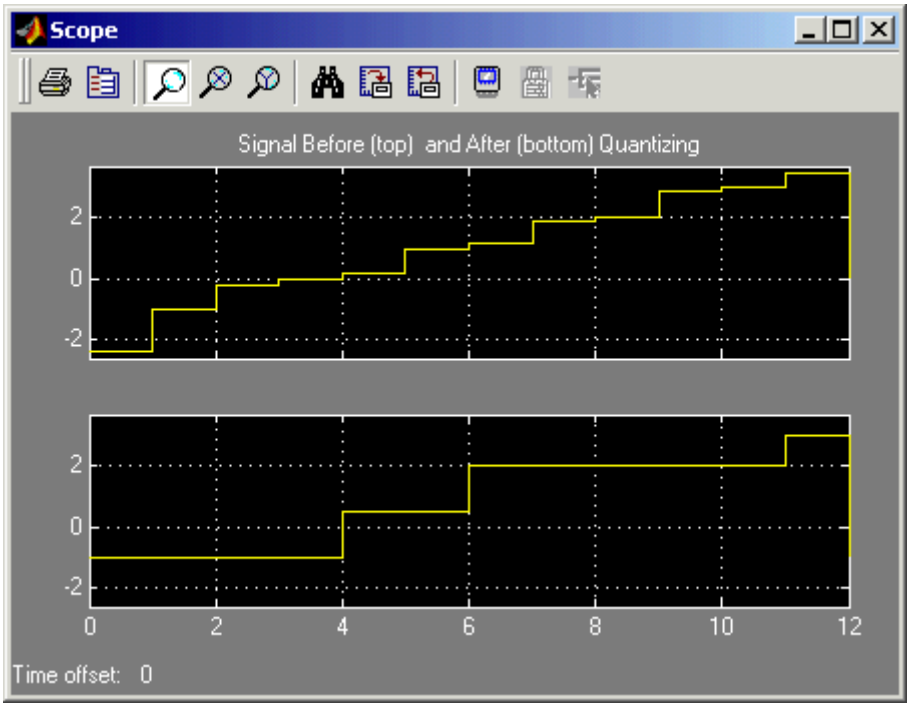
Quantizing a Signal

This section shows how the Quantizing Encoder and Quantizing Decoder blocks use the partition and codebook parameters.

Scalar Quantization Example 1

The figure below shows how the Quantizing Encoder block uses the partition and codebook as defined above to map a real vector to a new vector whose entries are either -1, 0.5, 2, or 3. In the Scope window, the bottom signal is the quantization of the (original) top signal.





To build the model, gather and configure these blocks:

Signal From Workspace,

in the Signal Processing Blockset DSP Sources library. The output signal samples obtained from the MATLAB workspace at successive sample times. A signal matrix is interpreted as having one channel per column. Signal columns be buffered into frames by sepecfiying a number of sample per frame greater than 1.

Set **Signal** to [-2.4,-1,-.2,0,.2,1,1.2,1.9,2,2.9,3,3.5]'.

Quantizing Encoder

The Quantizing Encoder block quantizes the input signal according to the Partition vector and encodes the input signal according to the Codebook vector. The input signal can be either a scalar or a vector. This block processes each vector element independently.

The first output is the quantization index. The second output is the quantized signal. The values for the quantized signal are taken from the Codebook vector.

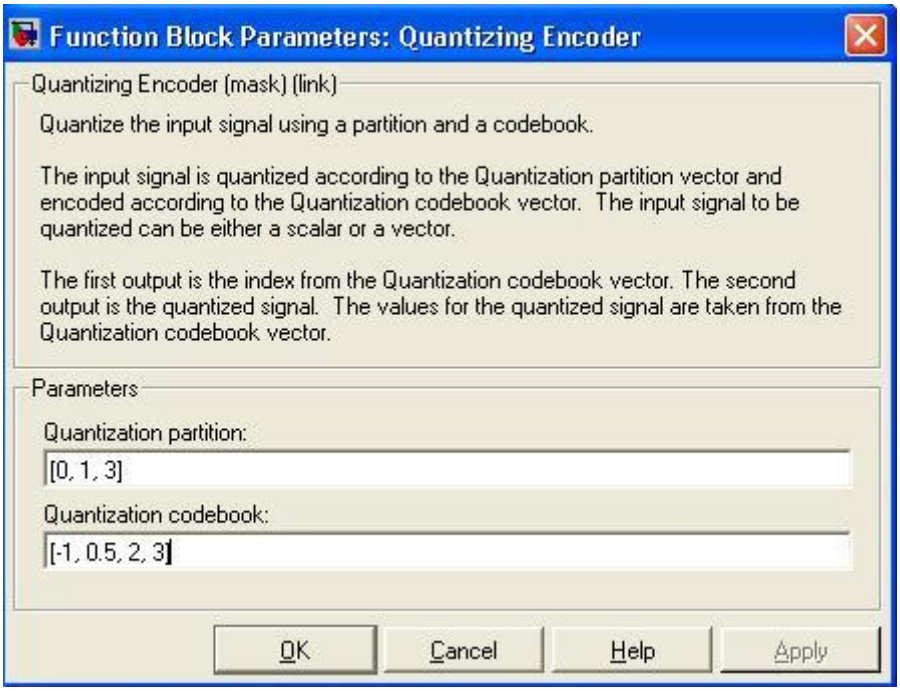
The Quantization partition parameter, P, is a real vector of length n whose entries are in strictly ascending order.

The Quantization codebook parameter, whose length is n+1, prescribes a value for each partition in the quantization. The first element of Quantization codebook is the value for the interval between negative infinity and the first element of P. The second output signal from this block contains the quantization of the input signal based on the quantization indices and prescribed values.

You can use the function lloyds in the Communications Toolbox with a representative sample of your data as training data, to obtain appropriate partition and codebook parameters

Set **Quantization partition** to `[0, 1, 3]`.

Set **Quantization codebook** to `[-1, 0.5, 2, 3]`.



Terminator

In the Simulink Sinks library. The Terminator block can be used to cap blocks whose output ports are not connected to other blocks. If you run a simulation with blocks having unconnected output ports, Simulink issues warning messages. Using Terminator blocks to cap those blocks avoids warning messages

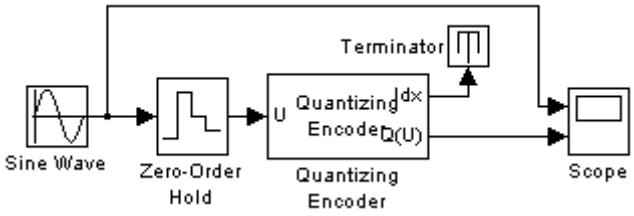
Scope In the Simulink Sinks library

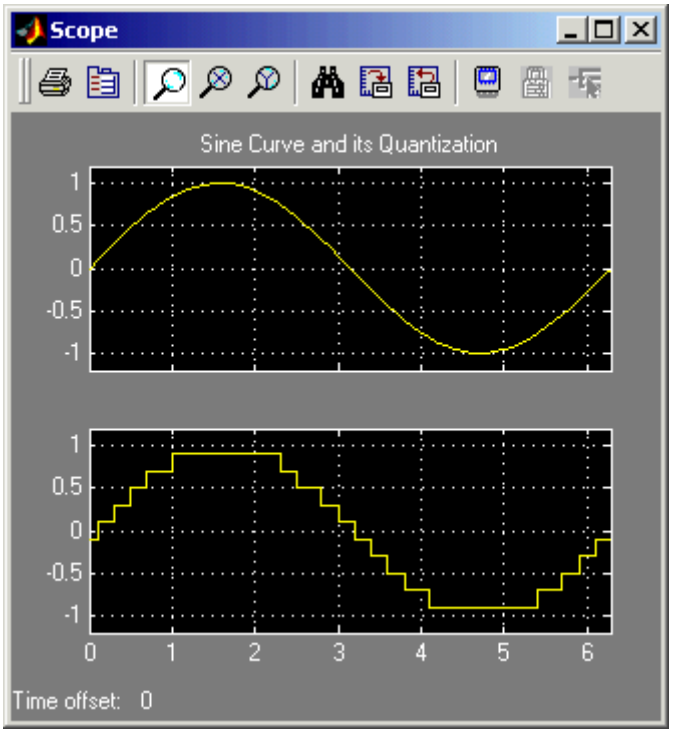
- After double-clicking the block to open it, click the **Parameters** icon and set **Number of axes** to `2`.

Connect the blocks as shown in the figure. Also, from the model window's **Simulation menu**, choose **Configuration parameters**; then in the **Configuration Parameters** dialog box, set **Stop time** to `12`. Running the model produces a scope image similar to the one above. (To make the axis ranges and title exactly match those in the figure, right-click each plot area in the scope and select **Axes properties**.)

Scalar Quantization Example 2

This example, shown in the figure below, illustrates the nature of scalar quantization more clearly. It samples and quantizes a sine wave and then plots the original (top) and quantized (bottom) signals. The plot contrasts the smooth sine curve with the polygonal curve of the quantized signal. The vertical coordinate of each flat part of the polygonal curve is a value in the **Quantization codebook** vector.





To open the completed model, click here in the MATLAB Help browser. To build the model, gather and configure these blocks:

- Sine Wave, in the Simulink Sources library (*not* the Sine Wave block in the Signal Processing Blockset DSP Sources library)
- Zero-Order Hold, in the Simulink Discrete library.
- The Zero-Order Hold block samples and holds its input for the specified sample period. The block accepts one input and generates one output, both of which can be scalar or vector. If the input is a vector, all elements of the vector are held for the same sample period.

You specify the time between samples with the Sample time parameter. A setting of -1 means the Sample time is inherited.

This block provides a mechanism for discretizing one or more signals in time, or resampling the signal at a different rate. If your model contains multirate transitions, you must add Zero-Order Hold blocks between the fast-to-slow transitions. The sample rate of the Zero-Order Hold must be set to that of the slower block. For slow-to-fast transitions, use the Unit Delay block. For more information about multirate transitions, refer to the Simulink or the Real-Time Workshop documentation.

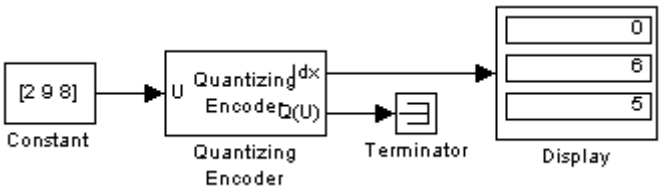
  - Set **Sample time** to `0.1`.

- Quantizing Encoder
  - Set **Quantization partition** to `[-1:.2:1]`.
  - Set **Quantization codebook** to `[-1.1:.2:1.1]`.
- Terminator, in the Simulink Sinks library
- Scope, in the Simulink Sinks library
  - After double-clicking the block to open it, click the **Parameters** icon and set **Number of axes** to `2`.

Connect the blocks as shown in the figure. Also, from the model window's **Simulation menu**, choose **Configuration parameters**; then in the **Configuration Parameters** dialog box, set **Stop time** to `2*pi`. Running the model produces the scope image as shown above. (To make the axis ranges and title exactly match those in the figure, right-click each plot area in the scope and select **Axes properties**.)

Determining Which Interval Each Input Is in

The Quantizing Encoder block also returns a signal, at the first output port, that tells which interval each input is in. For example, the model below shows that the input entries lie within the intervals labeled 0, 6, and 5, respectively. Here, the 0th interval consists of real numbers less than or equal to 3; the 6th interval consists of real numbers greater than 8 but less than or equal to 9; and the 5th interval consists of real numbers greater than 7 but less than or equal to 8.



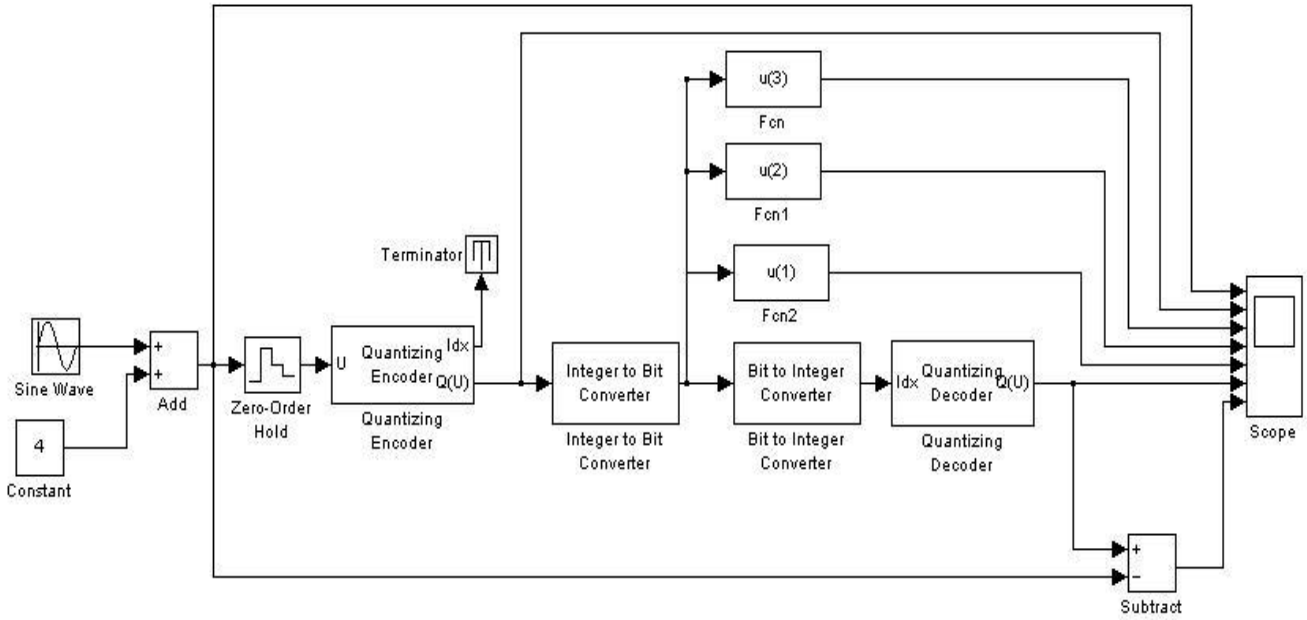
To open the completed model, click here in the MATLAB Help browser. To build the model, gather and configure these blocks:

- Constant, in the Simulink Sources library
  - Set **Constant value** to `[2, 9, 8]`.
- Quantizing Encoder
  - Set **Quantization partition** to `[3, 4, 5, 6, 7, 8, 9]`.
  - Set **Quantization codebook** to any vector whose length exceeds the length of **Quantization Partition** by one.
- Terminator, in the Simulink Sinks library
- Display, in the Simulink Sinks library
  - Drag the bottom edge of the icon to make the display big enough for three entries.

Connect the blocks as shown above. Also, from the model window's **Simulation menu**, choose **Configuration parameters**; then in the **Configuration Parameters** dialog box, set **Stop time** to `10`. Running the model produces the display numbers as shown in the figure.

You can continue this example by branching the first output of the Quantizing Encoder block, connecting one branch to the input port of the Quantizing Decoder block, and connecting the output of the Quantizing Decoder block to another Display block. If the two source coding blocks' **Quantization codebook** parameters match, then the output of the Quantizing Decoder block will be the same as the second output of the Quantizing Encoder block. Thus the Quantizing Decoder block partially duplicates the functionality of the Quantizing Encoder block, but requires different input data and fewer

Now try to construct this model



- **Quantizing Encoder**
  - Set **Quantization partition** to `[0.5,1.5,2.5,3.5,4.5,5.5,6.5]`.
  - Set **Quantization codebook** to `[0, 1, 2, 3, 4, 5, 6, 7]`.