Introduction:

There are three sections to this procedure. The first, Section 2.1 will show you how to start the Simulink utility and introduce you to the Simulink workspace. In section 2.2, you will determine the step response of a second-order control system, and in Section 2.3, you will modify our model to produce examples of overdamped and undamped system responses.

Start Matlab- Double click the Matlab icon on the desktop. It will take a minute or two for Matlab to finish initializing. The Matlab window should appear as show. Note: Setup may differ depending on MATLAB version you are using.

Start Simulink by clicking on the Simulink Library. A Simulink window will open as show. This window shows the libraries of Simulink Blocks that are available for you to use in constructing Simulink models. Open an existing model or create a new one. To create a new model, select File->New>Model, and new model window will appear as show.

From the Simulation tab, select Save > Save as. In the File name text box, enter a name for your model. For example, simple_model. Click Save. The model is saved with the file extension

Open Simulink Library Browser Simulink provides a set of block libraries, organized by functionality in the Library Browser. The following libraries are common to most workflows:

- Continuous — Blocks for systems with continuous states.
- Discrete — Blocks for systems with discrete states.
- Math Operations — Blocks that implement algebraic and logical equations.
- Sinks — Blocks that store and show the signals that connect to them.
- Sources — Blocks that generate the signal values that drive the model.

From the Simulation tab, click the Library Browser button.

Set the Library Browser to stay on top of the other desktop windows. On the Simulink Library Browser toolbar, select the Stay on top button.

To browse through the block libraries, select a category and then a functional area in the left pane. To search all of the available block libraries, enter a search term. For example, find the Pulse Generator block. In the search box on the browser toolbar, enter pulse, and then press Enter. Simulink searches the libraries for blocks with pulse in their name or description and then displays the blocks.

Performance Task: **18-39263-3**

You have a signal ‘m(t) = (2*sin(2*pi*4*t)+3*cos(2*pi*6*t))’. Apply amplitude modulation (AM) on the given signal with carrier signal ‘c(t) = (2*cos(2*pi*50*t))’, and then do demodulation to get back the original message signal m(t). Use appropriate modulation index (μ). Remember your demodulated signal should have same amplitude and frequency as m(t) has.

Formula for modulation: s(t) = (1+ μ*(m(t)))*c(t)

Formula for demodulation: m’(t) = (s(t)*2*c(t)-1)*(1/ μ) [Remember you have to use a low pass filter here to match m’(t) with m(t)]

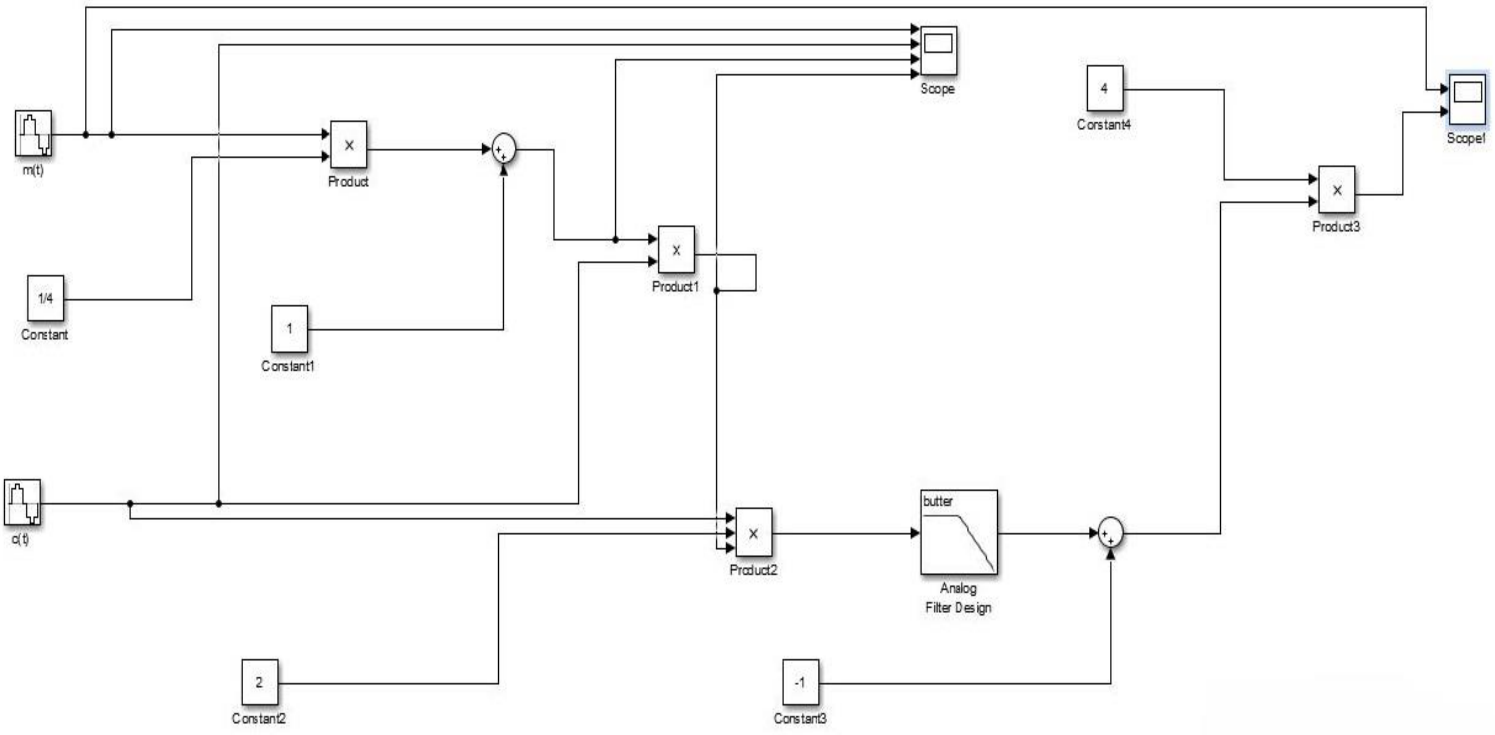
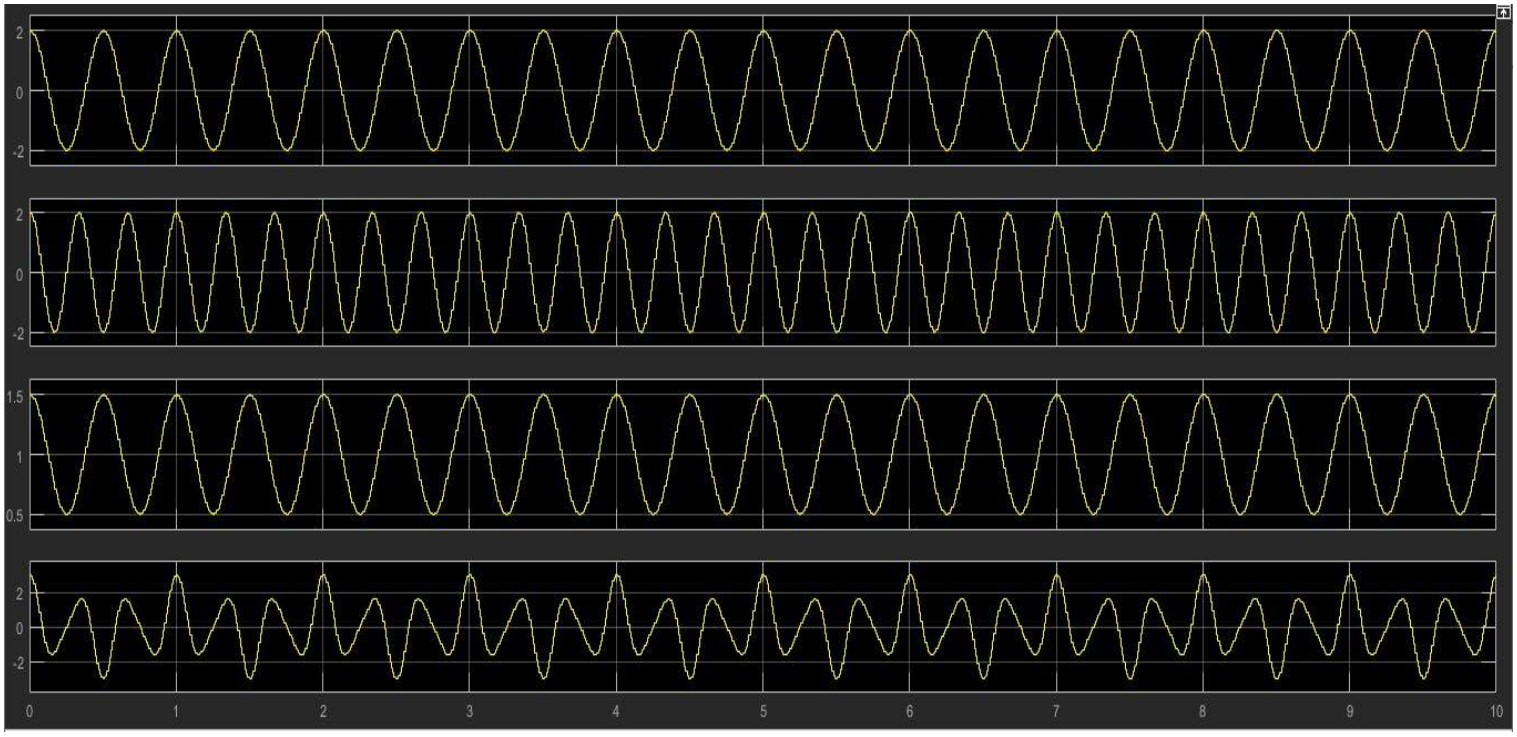
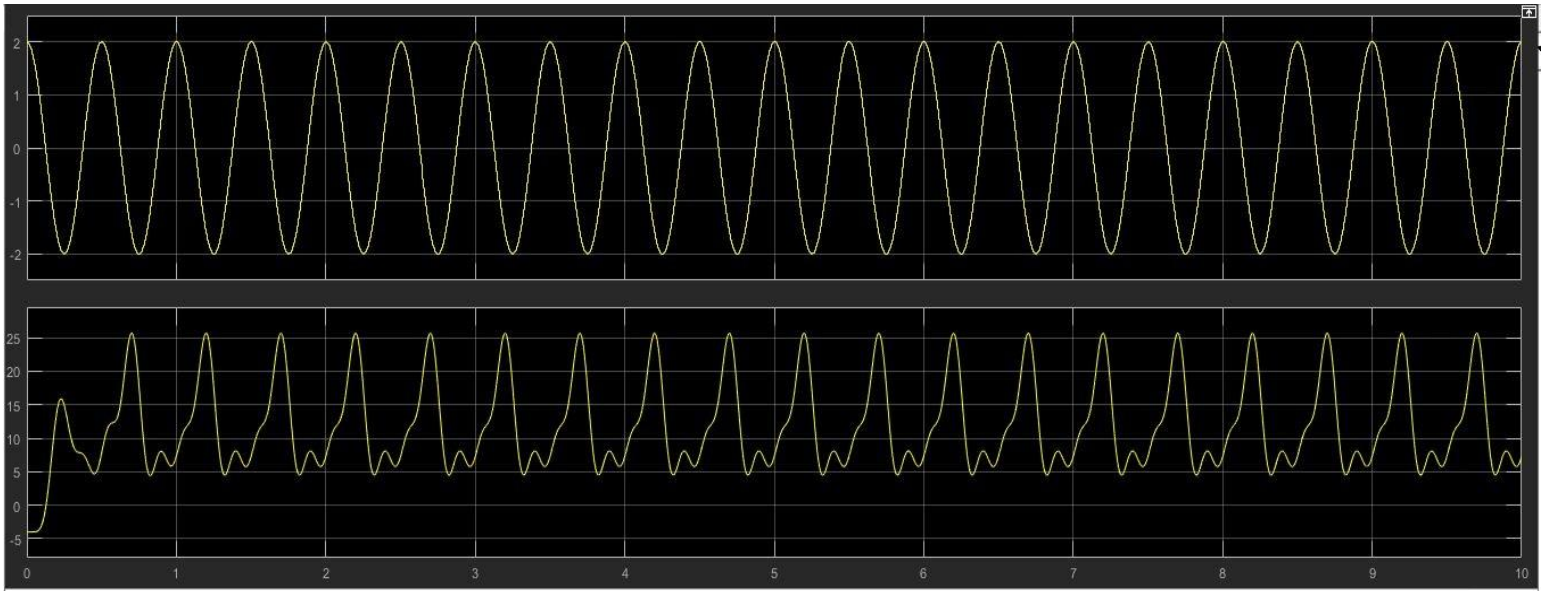


Figure: Amplitude modulation and demodulation block diagram.




2
1
0
-1
-2
25
20
15
10
5
0
-5
0
1
2
3
4
5
6
7
8
9
10

Conclusion:

In this experiment, we were presented with the key issues involved in designing, simulating and implementing an AM modulator and demodulator. We have used Simulink library tool in Matlab to practically implement the equation of AM with the help of blocks in library. This was done by using a simulation model, a DSP platform in which the model was implemented and an envelope detector for the demodulation of the signal.