Lab 4: Model Referencing and Angle Modulation

Objectives:

1) To learn to use model referencing in Simulink.
2) To simulate Frequency Modulation (FM) and Phase Modulation (PM).
3) To simulate FM modulation and demodulation with realistic signals.

Introduction:

Along with subsystems and libraries, model referencing is another way to implement hierarchical design. Compared to libraries, the main advantage in using model references is that they can be *accelerated*, meaning that the referenced model itself can be pre-compiled in order to create a performance increase. In this lab, model referencing will be used in the implementation of FM and PM modulation.

Angle modulation is a process in which the angle of the carrier wave is varied according to the message waveform. In this modulation technique, amplitude of the carrier wave is maintained constant, which makes the transmission power constant. However, this benefit is achieved at the cost of increased transmission bandwidth.

The two primary methods of achieving angle modulation are phase modulation (PM), in which the instantaneous phase is varied linearly with the message signal, and frequency modulation (FM), in which the instantaneous frequency is varied linearly with the message signal. Both techniques will be explored in this lab.

Preliminary:

Given that the message signal is $m(t) = \sin(2\pi(10)t)$ and the carrier frequency is $f_c = 100$ Hz, find the instantaneous frequency and phase of FM and PM signals. Assume that the frequency deviation constant is $f_d = 5$ Hz for FM, and the phase deviation constant is $k_p = 2\pi$ for PM. Put your results in a table.

Lab Procedure:

Part A – Designing a Simulink Submodel

The model reference feature is provided by the *Model* block in the Ports & Subsystems library in the Simulink library browser. The instance of a Model block which represents another model is called a *referenced model* or *submodel*. The model that contains a referenced model is called its parent model. In order to demonstrate these features, we will create a model file that performs PM modulation, then use that model as a reference in another model.

To start, create a new blank model and save it as "**pm_mod.mdl**". Next, go into the configuration options and change the solver to discrete time. In order for this model to function as a reference within another model, there is one more configuration option that must be set. On the side bar of the configuration panel, go to *Optimization* -> *Signals and Parameters* and check the box labeled "*Inline parameters*". If this box is not checked, the model will generate errors when you try to run it as a reference within a parent model.

Next, we will create the model itself. In PM modulation, the phase of the carrier is varied according to the amplitude of the message signal. There are a number of different ways to do this in Simulink, but perhaps the easiest is to manipulate the time variable of the carrier wave directly. Create a sine wave generator in the new model, then open the parameters and set *Use External Signal* for the time variable. This will essentially create a block that will output **sin(ωt)**, where **t** is a signal that you feed into the block.

In order to make the output of this block vary according to the current simulation time, we need to create a *Digital Clock* block which will supply the base value for t. By connecting one of these blocks directly into the sine wave generator, it will generate an ordinary sine wave which only varies with the simulation time, which works exactly like sine wave blocks that don't use an external signal for time. However, we also want this **t** variable to be influenced by the input signal, in order that we might change the current phase of the carrier. To do this, we add a *Sum* and *Gain* block, which has the net effect of adding or subtracting from the time that the sine wave generator sees, thus altering its phase. The complete system, along with all relevant parameters, is shown in Figure 1. Note that the gain block, as configured below, will vary the phase of the carrier between $-\pi$ and π so long as the message signal goes between -1 and 1. If the message exceeds these limits, the modulation will make some message signal levels indistinguishable from others as the phase wraps around. Make sure that any input to this model is limited between -1 and 1 for this reason.

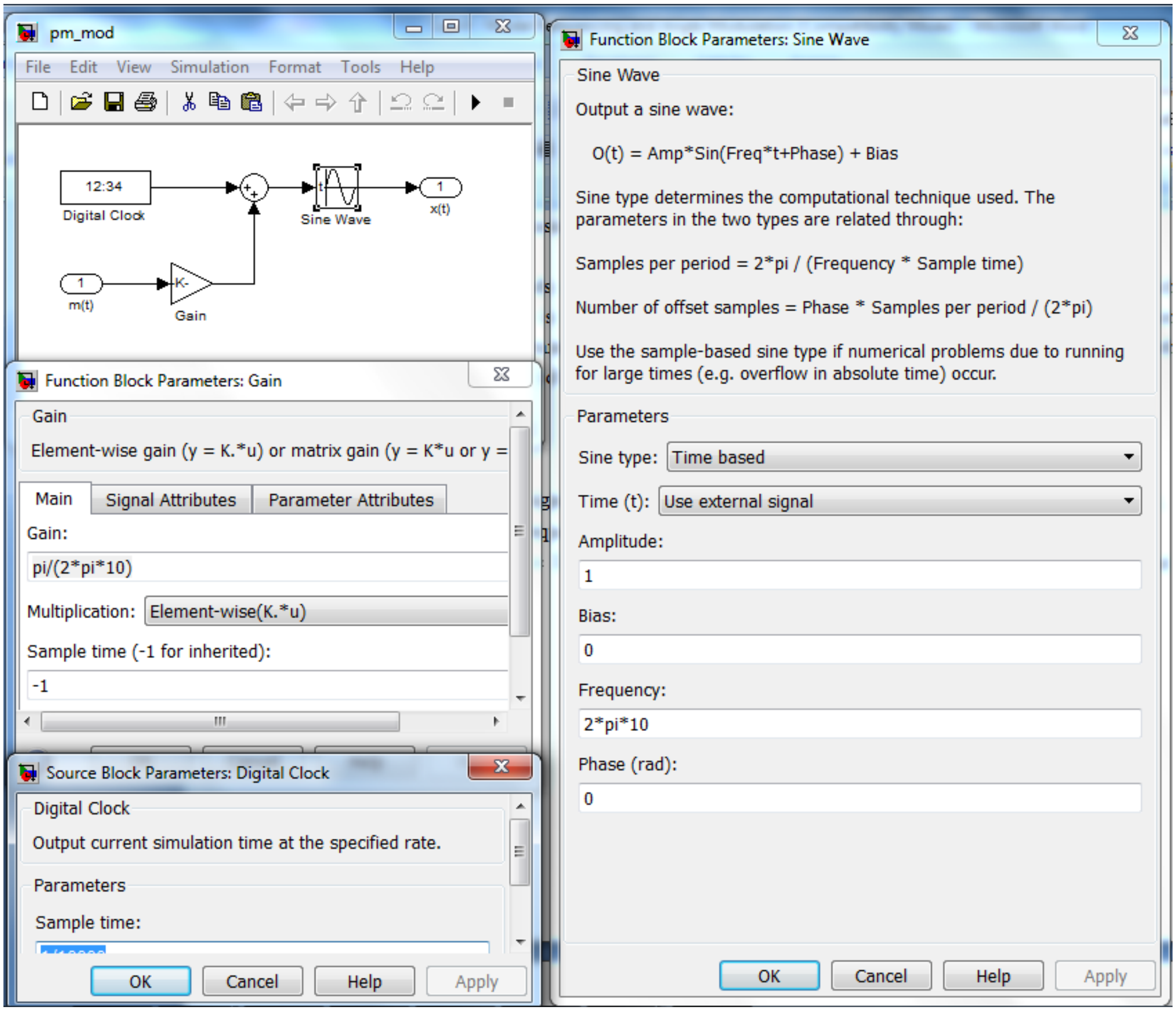

Figure 1 – Finished PM modulator model

Using these parameters, the sampling frequency is 1000 Hz and the carrier frequency is 10 Hz. Make sure that any input signal you use in the parent model is much less than 10 Hz, or else the modulator will distort the output. Increasing the sampling frequency and carrier frequency is also a solution, if you wish to modulate a signal with higher frequencies.

<u>Part B – Designing a Simulink Parent Model</u>

Now that we have a working modulator, it is time to put it to use. Create a new model file and save it as “**lab4.mdl**”. Make sure that the same configuration parameters are set for this new model as were set for the previous one; the solvers must be consistent across all models when referencing is used. However, it is not necessary to set *Inline parameters* for the top-level model.

In order to use the modulator we just built in this model, we must first create an empty submodel block. Find the *Model* block in the Simulink library and place it. Initially, it starts out red and empty, because no model has yet been placed in it. Now, open the parameters window and browse for the model file you created the PM modulator in. Make sure that the simulation mode is set to *Normal*. The Accelerated mode is useful for speeding up models, but it requires a MATLAB compiler to create machine-level code to do this. Many machines do not have the appropriate compilers installed, so we will not experiment with this feature in this lab.

We need to create an FM modulator next. FM modulation is remarkably easy in Simulink, as the only block it requires is a VCO, or voltage-controlled oscillator. Open a new model file and add the *Discrete-Time VCO* block, along with one input and one output port. Set the quiescent (or center) frequency to the carrier frequency, and the input sensitivity to approximately half the carrier frequency. Perform the same steps as in Part A to create a valid submodel, then save it as “**fm_mod.mdl”.** Then add this model to your parent model as a reference as before.

Now that we have referenced these models, we will create a message signal, send it through the modulators, then look at them on the scope. When this is done, you should get a result that looks like that of Figure 2. In this example, the carrier frequency for both modulators was 10 Hz, and the message signal was a sine wave of 1 Hz. Save a screenshot of the scope result, then comment on why the output waveforms look like they do. Next, change the message signal from a sine wave to a square wave and repeat this simulation. Compare the PM and FM modulated signals.

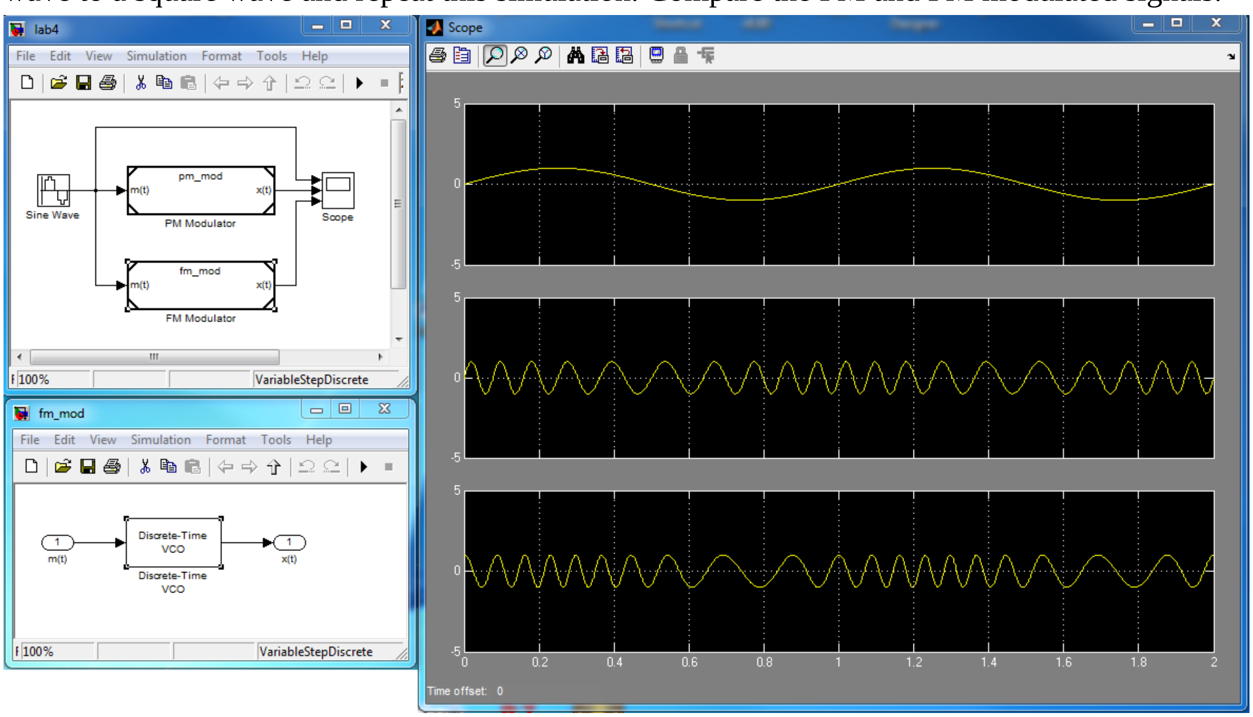

Figure 2 – Finished parent model and scope results

Part C – FM Modulation and Demodulation with Realistic Signals

Now you are to simulate a realistic FM modulation and demodulation system. First create a model call "**FM_radio.mdl**", then add the message signal source and the FM modulator you built in Part B. Find a demodulator to recover the message signal and use a sink (scope) to view the original message signal, FM modulator output, and demodulator output. You may use the built-in FM demodulator block from the Communications Toolbox. Be sure to look under mask to explore what the model is made of and read the help menu entry of this block to understand its operating principle. In your report, describe what you have learned about it.

Experiment the FM modem by varying your parameters in the three building blocks to observe the outputs on the scope and explain your results. Parameters in the three building blocks are listed in the table below:

Name of Block	Parameters
signal source	signal amplitude and frequency
FM modulator	carrier frequency (Quiescent Frequency of the VCO), frequency deviation constant (determined through the Input Sensitivity parameter of the VCO), initial phase, and sampling time
FM demodulator	carrier frequency, initial phase, frequency deviation, and Hilbert transform filter order

Note: In order for FM demodulation to be accurate, the frequency deviation should be orders of magnitude lower than the carrier frequency. Leaving the carrier at 10 Hz and the deviation at 5 Hz will make any attempt at demodulation very problematic. Therefore, finding the suitable parameters for your design is critical to your FM communication system.

Bonus: If you create your own discrete-time PLL model block-by-block and build the FM demodulator using the PLL, your properly working demodulator may be worth up to 50% extra credit. Make sure to submit both the demodulator model files and the scope output for the demodulated signal.

Post-Lab Questions:

Q1. In what ways is a referenced model block different from a subsystem block?

Q2. How would your model be different if the PM or FM is wideband modulation rather than narrowband modulation?

Q3. In Part B, look at the modulated signal for both PM and FM. If you know the input signal is a sine wave or a rectangular pulse, and you are given the modulated output without any other information, is it possible to tell for certain whether the modulation was FM or PM? Why or why not?

Q4. What is the difference between the sine wave blocks used in the PM modulator of Part A and in the signal source of Part B? How are they different from the sine wave source you used in Lab 2 and Lab 3?

Q5. In both types of angle modulation, the transmission power is kept constant no matter what the message signal is. Why is this advantageous over AM, especially for transmission of music?