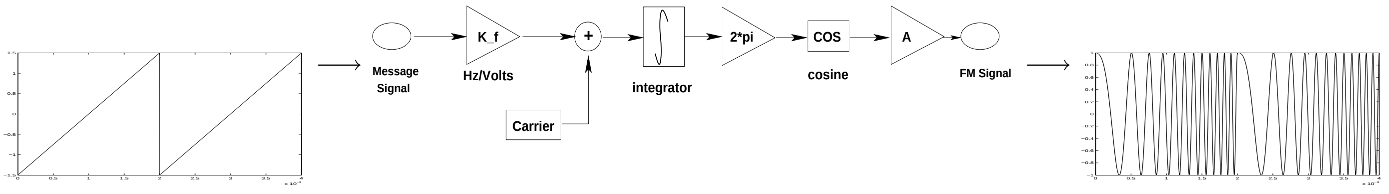
Experiment # 4

Frequency Modulation



1 Purpose

In Experiment # 3, you designed and built an AM modulator, and had the opportunity to see how an envelope detector works in demodulating an AM signal. In this experiment, another widely used modulation technique will be introduced: Frequency Modulation (FM). As the amplitude of the sinusoidal carrier wave was modulated in AM, this time the instantaneous frequency of a sinusoidal carrier wave will be modified proportionally to the amplitude variation of the message signal.

A three-step process will allow you to become more familiar with FM:

- First, a frequency modulation system will be created and simulated using `Simulink`.
- Second, an FM demodulator based on a Phase Locked Loop (PLL) will be designed and simulated, using the previous modulator as the "signal source".
- Third, an FM demodulator will be implemented on a TMS320C6711 DSP platform. The input FM signal to the demodulator will come from the signal generator.

You will be required to change various system parameters and observe the consequences of the changes.

2 Background Reading and Preparation

Frequency Modulation (FM), as well as Phase Modulation (PM), are types of Angle (Exponential) Modulation. These are covered in Chapter 5 of [2], or Chapter 3 of [1]. If you have the time, take a look at the theory behind Phase-Locked Loops (PLL), as in pp. 184-186 of [2], and its use on FM demodulators, as in pg. 236 of [2] or Section 2.14 of [1]. A good explanation about the role played by the Bessel Function on the frequency domain representation of FM signal is found on pg. 115 of [1].

Also *before* coming to the lab, you should complete the **lab preparation** and hand it to the T.A.. In order to perform this experiment effectively, a good understanding of `Simulink` and `Matlab` is required. Feel free to use the lab to practice your skills before the experiment.

3 Equipment

Hardware:

1. One Signal Generator;
2. One Two-Channel Oscilloscope;
3. One Spectrum Analyzer
4. One TMS320C6711DSK Module (with audio daughtercard) attached to a workstation;
5. Coaxial cables, BNC-to-BNC.

Software:

1. `Matlab` Release 13
2. `Simulink` with ECE416 Toolbox
3. `Code Composer Studio`, v.2.1

4 Experiment

The experiment is divided in three parts: the design and simulation of an FM modulator in `Simulink`; the design and simulation of a PLL-based FM demodulator; and the download and test of the designed demodulator on the DSP platfom. For this third step, the incoming FM signal will be produced by the signal generator.

All results are to be reported in the spaces provided in this outline. Make sure that the T.A. verifies every result you record.

4.1 Designing and Simulating an FM Modulator

Based on the FM signal equation and the block diagram for an FM modulator that you have submitted with your lab preparation, build your model in `Simulink`. Follow similar procedures for input sources and output sinks to what you have used in previous experiments. In particular, do not forget the following details:

- Your life will be made easier at the **downloading** part of the experiment if you utilize in your model the *constant block* found under the `ECE416` blockset. For the **simulation** part of the experiment, use the *constant block* found in the `DSP Sources` library in `Simulink`;
- The input to an FFT-based scope must be buffered, with a buffer size that is an integer multiple of the FFT size (could be the same size). It makes no sense to attempt to perform an FFT on a single sample, or buffer a smaller number of samples than the length of the FFT;
- A greater FFT length means a better resolution for the frequency domain representation of the signal (take 1024 points as a reasonable length). At this point you should be able to tell when the results are producing relevant information or insufficient information;
- All blocks that require a sampling frequency (or sampling period) to be defined **must have** the same sampling frequency. Since the daughtercard utilized in the lab makes use of a 48KHz sampling frequency, you should preferably use this number throughout your simulation exercise. However, feel free to change the sampling frequency during the simulation, but **do not forget** to change it back to 48KHz prior to downloading your model to the target hardware.
- The frequency of a carrier wave in practice is much higher than the one utilized in this experiment. One must keep in mind, however, that a limitation is imposed by the sampling frequency utilized by the CODEC daughtercard.

To start, follow the block diagram you drew on your lab preparation sheet and design a model for the FM modulator in `Simulink`. If the straight-forward approach to "build the model from the equation on the book" does not work, re-work the equation. Maybe bringing some elements "into the integral" will simplify your block diagram. Do not despise the factor 2π on the constant k_f if you are working with ω (in *rad/s*).

Use as input signal a sine wave with amplitude 0.5 (this would mean a 1Vpp simulated signal) and 500Hz, a carrier frequency of 6000Hz and a sensitivity factor of 3600Hz/V (that's the k_f constant on [2]). You should observe an output signal similar to the one you drafted on your lab preparation sheet.

Be prepared to answer questions asked by the T.A.

- *In the space below, sketch the output signal observed in the time domain. Indicate values where appropriate.*

- *Sketch the output signal observed in the frequency domain. Explain the displacement and amplitude of the frequency components. Show the calculation of the deviation ratio β and the bandwidth according to Carson's Rule.*

4.2 Designing and Simulating an FM Demodulator

Following the block diagram of the PLL-based FM demodulator you have drawn on your preparation sheet, build a new model in `Simulink`. The input signal for the demodulator simulation will be the FM-modulated signal from your modulator. Your demodulator should be made of a PLL and an output low-pass filter. The PLL, as you have drawn on your lab preparation, is made of a Voltage-controlled oscillator, a multiplier and a low-pass filter. It is not always necessary to have a second filter after the PLL. You may be able to avoid the use of a second filter by using a loop filter with a sharper roll-off at the cutoff frequency of interest. To make your life easier, you can create a "subsystem" with your modulator (VCO), and utilize this new block to implement your demodulator. To do this, select the blocks on yor model that implement the VCO, and click on `edit/create subsystem`.

The constants and gains that you have to adjust for the VCO are the same as the ones used on the modulator (the VCO **is** a frequency modulator). The filter is a low-pass FIR of order 40, designed using the windowing method with a Kaiser window. The cutoff frequency is the frequency of your **modulating** signal. The output filter is identical to the filter in the loop. Note that if you are looking for a loop filter with a sharper roll-off, you may need to redesign it to have a higher order, or utilize a different window (you are using a Kaiser window) or yet to go for another design method altogether. Feel free to explore with different filters, if time allows. The loop filter will separate the "detected" signal from all other frequency components, wich arise from the multiplication of the incoming (FM) signal with the one output by the VCO.

You can observe this by attaching three FFT-based scopes to your demodulator: one after the multiplier, one after the first low-pass filter (output of the PLL) and one after the output filter (the output of your demodulator). You will need them to answer the next questions. Also, attach a time-based scope to the output. Now run your system and observe the output scopes.

- *In the space below, sketch the output signal observed in the time domain and in the frequency domain. Does your demodulator work?*

- *The intention behind this step is to hinder the detection performed by the PLL. Vary the frequency of your carrier on the* **modulator** *by about 200Hz. Is your demodulator working properly? If it is, increase the frequency until your output is no longer a sinusoid. Present a brief explanation of what happened, in your own words.*

- *Similarly to the previous step, this time vary the frequency of your modulating signal (again, on the* **modulator***) by 200Hz up and down. Is your demodulator working properly? Sketch your results and explain them.*

4.3 Building and Running Your Model

In this part of the experiment you will modify your FM *d*emodulator and download it to the DSP target hardware. The input to your demodulator will be an FM signal coming from the "real" signal generator. Since you have simulated your demodulator with an FM signal with 1Vpp, 6000Hz carrier and deviation of 3600Hz, you should utilize these same parameters to generate the input FM signal on your signal generator. Use as a message signal a 500Hz sinusoid, 1Vpp.

You can cut and paste your whole FM demodulator system onto the basic model found in `work/template/direct.m` You do not need to substitute any blocks (such as the `DSP constant` block). However, two gains must be inserted for the demodulator to work properly with the `Simulink` blocks used. Insert at the input to the demodulator (PLL) a gain of 1/(2e29) and at the output of the PLL another gain of (2e29). This will cause the bit word at the input to be shifted *right* prior to entering the PLL and at the output it will be shifted *left* by the same amount prior to entering the channel combiner.

Follow these steps in order to get your system working:

- Build your model from the `Simulink` build. You can achieve this by using the build button or pressing Ctrl-B. The result of this action will be a project exported to `Code Composer Studio`.

- In `Code Composer Studio`, build your project. This means to compile the programs generated and creating an executable program to run on the board. You can do this by clicking on the "build all" button or going under project/build all. You will see the files being compiled, linked and assembled in the dialogue window.

- Now that an executable has been created, you must load it onto the target hardware. Go under file/load program. Select the program you just built as the one to be loaded. You will see a progress bar indicating that your file is being loaded into memory.

- Run your program. You can do this by pressing on the "running-man" button on the left-hand side of your screen, by pressing F5 or by going under project/run. If your board is connected and powered and if your signal generator is set to provide you with an output (these are common mistakes), you should see an FM signal and a sinusoid (i.e., your demodulated signal) on your scope. It can be the case that you may need to load the file once again for it to work. If everything looks fine but it does not work, try loading it again and running it. If the problem persists, call the TA or the Lab Engineer.

1. **Time Domain and Frequency Domain Results**

   Using the oscilloscope, observe the characteristics of the output signal. Make a comparison with the signal observed during the simulation stage. At every stage in this section, you will be required to observe the signal in both the time domain and the frequency domain. This is to say that for every change made to the input signal, you will be required to run the sample code given in previous experiments to retrieve the data from memory and plot it using `Matlab`. At this point you must know how to make the appropriate changes on the `Matlab` program to produce meaningful results. Remember, in particular, that plotting directly the results of an FFT operation does not give you values that represent "hertz" on the x-axis. You must know how to associate the results of an FFT with a vector representing frequency.

   Alternatively, you can use the FFT capability of the oscilloscope to view the frequency domain of each of the channels. If your oscilloscope does not allow you to perform an FFT on the input signal, use the program.

   - *On one channel you have the FM signal and on the other you have your demodulated signal. Explain how can you tell that your demodulation is working properly? Use whatever tool/plot that you deem appropriate to respond to this.*

   - *Using the signal generator, vary the frequency of the carrier by about 50Hz, up and down. Report and explain the effect on the* **spectrum** *of the output (demodulated) signal.*

- *If you change the cutoff frequency of the filters to 1KHz, what is the consequence?*

5 Conclusion

In this experiment, you were required to design and simulate an FM modulator and demodulator, and to implement an FM demodulator on a DSP platform. As a result, became familiar with the theory behind Angle Modulation, and with practical details pertaining to Voltage-Controlled Oscillators and Phase-Locked Loops as they were implemented on a DSP platform. All of these concepts are widely used in the study of communication systems.

Obs: The books cited in this outline contain relevant material for the student to better understand the topics related to the experiment. The student should note that by reading the course textbook only, one should be able to successfully perform the experiment. This is to say that it is not necessary for the student to purchase all the references.

References

[1] S. Haykin *Communication Systems*, 4th Edition. Toronto: John Wiley & Sons, Inc., 2001.

[2] B. P. Lathi, *Modern Digital and Analog Communication Systems*, 3rd Edition. New York: Oxford University Press, 1998.