Experiment # 4

Frequency Modulation

1 Purpose

In Experiment # 3, a modulator and demodulator for AM were designed and built. 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 the 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]. *Before* coming to the lab, you are encouraged to read the sections pertaining to FM. Also, 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. Three Coaxial cables BNC-to-BNC.

Software:

1. `Matlab` Release 12
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 *sinusoid generator* and *constant block* found under 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;

- A greater FFT length means a more accurate 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.*

- *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?*

- *Sketch the results observed on the three FFT-based scopes on your demodulator. Make sure to indicate at which point of your demodulator you are observing the signal. Explain how the demodulation process works, based on what you observe on the scopes. Remember, you may not need the second FIR filter, if your loop filter gives you the desired output. Point that out if this is the case*

- *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.

Some other details must be reviewed. When you substitute the sample-based signal generator by the ADC block for the C6711, do not forget to set the number of samples per frame to "1". You will notice that upon running your model (press the "play" icon), all arrows connecting the blocks must be *thin*. A value different than 1 will generate thick arrows. If you build your model with a ADC utilizing frames of samples, the code generated will not run properly. Also, remove all "sinks" from your previous `Simulink` model, since an attempt to generate code for oscilloscope icons (or any other "sink" icons) will produce an error.

Insert two gain blocks to your existing demodulator: one just after the ADC and one just before the DAC. These gain blocks will adjust the incoming and outgoing data to fit properly in the word exchanged between the Daughtercard CODEC and the DSP. The value of these gains will be given to you by the T.A.. If after you build your model and download it to the target you realize that the output signal is voltage is less than 1Vpp, correct the output gain and rebuild your system.

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 frequency points. You must know how to associate the results of an FFT with a vector representing frequency.

- *Using the signal generator, vary the frequency of the carrier by about 200Hz. What happens? Why does that happen?*

- *Now go to your* `Simulink` *model and modify the cutoff frequency of the filters (say, to 3KHz). Rebuild your system. Report what you observe before and after the modification, and explay why.*

5 Going The Extra Mile

For three experiments you were encouraged to look at the C code generated and to become familiar with the process of generating that code. Feel free to try and modify some parameters on your demodulator without having to go to the `Simulink` model. For instance, you can change directly the value of the gains, the carrier frequency and the sensitivity factor of your demodulator right from the code, which makes the process much simpler than going back to `Simulink`. If you already feel

confident, you could try to look at modifying also the filters, starting with the existing coefficients and later modifying the code to accomodate filters of different order than the one generated by `Simulink`.

Remember that for every change you make, you must recompile and rebuild your code, and load the object code onto the target hardware. After that is done, you have to run the program from Code Composer Studio. You are encouraged to explore the options Code Composer Studio gives to the designer. For instance, the sequence: compile, build and load can be done in one single step (look under "Option/Customize"). Also, there are tools for plotting data in memory directly using the visual interface provided by Code Composer Studio.

6 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, with the use of state-of-the-art software tools. As a result, you not only became familiar with the theory behind Angle Modulation, but also with practical details pertaining to Voltage-Controlled Oscillators and Phase-Locked Loops. 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 pertaining 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.

[3] The Mathworks, Inc., *Using Simulink*.

[4] The Mathworks, Inc., *Real-Time Workshop*.