The University of Calgary

Department of Electrical & Computer Engineering

ENEL 471-Introduction to Communication Systems and Networks

(Winter 2020)

Lab 3 : Phase Lock Loops and FM Introduction

Section date	B04: Tue. Mar. 10, 2020 B03: Tue. Mar. 17, 2020	B02: Wed. Mar. 11, 2020 B01: Wed. Mar. 18, 2020
Location	ENA 301	ENA 301

There is no hardware component in this lab. Do as much as you can on the Simulink components prior to the lab session.

This document consists of 2 parts:

Part I: Lab 3 Manual

- Read the Lab 3 Manual prior to the Lab 3 session
- Do any necessary pre-work that is required (i.e., familiarize with the simulation aspects of the Lab prior to the actual Lab session)
- Students must leave the Lab NO LATER than 5:00 pm

Part II: Lab 3 Questions

- Answers to these questions must be submitted to the TAs at the end of the Lab period.
- Each group submits one set of Lab questions. Ensure names and ID numbers of the group members are written on the Lab answer sheets.

Acknowledgements

The ENEL 471- Introduction to Communications and Networks Lab 3 document was originally prepared by Jennifer A. Hartwell and Dr. Mike Potter, revised (January 2013) by Warren Flaman and Dr. Abu Sesay, revised (January 2015) by Mohamed Al Masri and Dr. Mohamed Helaoui, and revised (January 2018) by Leanne Dawson and Dr. Mohamed Helaoui, and revised (March 2019) by Yulong Zhao.

Part 1

Lab 3 Manual

Introduction

There are three parts to this lab, all of which are simulation-based. The first part introduces and demonstrates the operation of phase lock loops (PLLs) which are feedback systems that lock onto the phase of an incoming signal so that operations such as coherent demodulation (recall Lab #2) can be performed. The second part of this lab introduces the two-stage coherent demodulation design that wasn't covered fully in Lab #2 (Superheterodyne Receiver). The third part of this lab introduces and demonstrates Frequency Modulated signals (FM) with an emphasis on bandwidth.

Important Formulas

- $\sin\varphi\sin\theta = \dfrac{\cos(\varphi-\theta)-\cos(\varphi+\theta)}{2}$
- $\cos\varphi\cos\theta = \dfrac{\cos(\varphi-\theta)+\cos(\varphi+\theta)}{2}$
- $\sin\varphi\cos\theta = \dfrac{\sin(\varphi+\theta)+\sin(\varphi-\theta)}{2}$

1 Simulating the Phase Lock Loops (PLLs)

1.1 The most Basic of PLLs

In Lab #2, coherent modulation was only possible when the receiver had knowledge of the carrier which was used to modulate the message signal. It was also shown that having phase error present in the demodulating carrier caused distortion of the received message.

A PLL is a method that can be used to 'lock on' to the phase of a received signal. The operation of a PLL is possible due to the behavior of a Voltage Controlled Oscillator (VCO). A VCO outputs a constant amplitude sinusoid that has a natural oscillation frequency (or 'quiescent frequency') of f_c when the input to the VCO is 0 V. As the input to the VCO changes, so does the frequency of the sinusoid that is generated. For now, do not consider the operation of the VCO in any further detail.

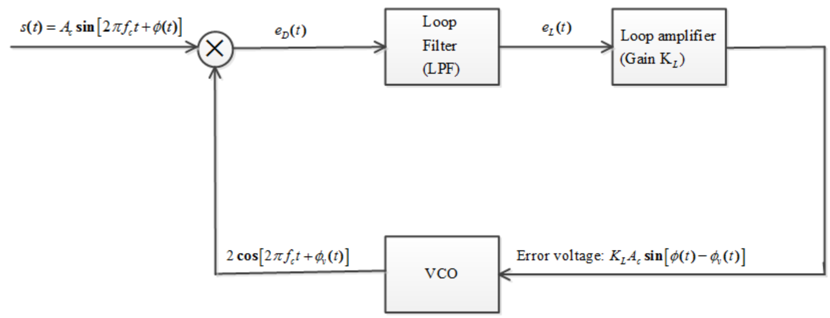

Figure 1: Block diagram of a basic PLL.

The basic block diagram of a PLL is shown in Fig. 1. If the natural frequency of the VCO is set to the same frequency as the incoming signal, then after multiplication (or mixing), the low-pass (loop) filtering and optional loop amplification, the output is proportional to the sine of the difference, $\mathbf{sin}(\phi_{\mathrm{e}})$, of the two different phases. As the difference between the phases gets very small, then $\mathbf{sin}(\phi_{\mathrm{e}}(\mathbf{t})) \approx \phi_{\mathrm{e}}(t) = \phi(t) - \phi_{\mathrm{v}}(t)$. The error voltage is then used as the input to the VCO so the larger the error, the more the VCO will adjust its frequency, and likewise, when the error gets really small the VCO will not adjust as much.

If you have been following everything up until now, you may be asking yourself '*Changing the output frequency of the VCO can reduce the phase error?*'. The answer is yes. There is a very direct relationship between phase and frequency. Frequency is the derivative of phase.

Now, without getting caught up in mathematics, just try to imagine what is happening in our basic PLL scenario: if there is a phase error present then the VCO's frequency will change, and since it is then going either slightly faster or slower than the input signal, their relative phase will change. As the difference in phase between them becomes smaller, so will the error voltage, and so the frequency of the VCO will return back to the natural frequency, but now the two signals are phase locked.

It is important to remember that this analysis was based on a *sine* input signal and a *cosine* VCO output signal. Therefore, when the phase is 'locked' it actually means a difference of $-90°$ between the input and the VCO output. This is okay though, as what we have done is taken an input signal with an unknown phase and created a local signal that we know has exactly a $-90°$ difference in phase; hence we know what the phase of the input signal is.

⇒**Open the Simulink file 'pllbasic.mdl'.**

The frequency of the incoming sine wave has been set to 500Hz with a phase of 0°. The same settings have been entered for the VCO's natural frequency and initial phase.

- Change the incoming sine wave's phase to $\frac{\pi}{3}$ and run the simulation. Looking at the plot of the error voltage: **(Q1)**
  - Does the error voltage converge?

  - If it does, then what value does it converge to?
  - Approximately, when does it appear to converge or approximately when does it appear that the PLL has 'locked-on'?
- Look at the plot of the input and tracking signals. **(Q2)**
  - Zoom in near the beginning, what is the phase difference?
  - Approximately, when does phase-lock happen (i.e., when does the phase difference become $\frac{\pi}{2}$)?
  - Do convergence of the error voltage and phase-lock occur at the same time?
- Change the value of the amplifier gain to '2', run the simulation and look at the plots.
- Change the value of the amplifier gain to '0.50', run the simulation and look at the plots.
- Describe how you think the gain factor affects the operation of the PLL. **(Q3)**
  - Which gain produces a faster convergence (or lock-on)?
  - In general, which gain do you conclude would produce faster convergence? high gain or low gain?
- **Restore the Gain to 1**

Now it is clear that the PLL can track an incoming signal that has a phase that is different from the VCO's initial phase. But what if the incoming signal has a slight frequency drift so that the received frequency is not exactly what we expect it to be? Could the PLL still create a signal that has exactly a 90° phase difference? The answer is yes, because a small difference in frequency can be viewed as a time varying difference in phase and you will see that as long as the frequency is 'close enough' (how close depends on each particular PLL) then the VCO can still output a signal that tracks the input (the range where they are 'close enough' is often referred to as the 'lock-in' range).

- Set the sine input's phase to 0, change its frequency to 500.01 Hz and run the simulation.
- Did the error voltage still converge to a value? Zoom-in on the Y-axis around 2 seconds. Did the error converge to zero? What did it converge to and why? **(Q4)**
- Look at the input/output signals and state whether the phase was successfully tracked. **(Q5)**
- Change the gain block to '2' and run the simulation again. What happened to the error voltage? **(Q6)**
- Set the gain back to 1. Change the sine input's frequency to 501 Hz, and run the simulation. Try again at 502 Hz and 505 Hz. Make comments about the error voltage and the input/output signals for these signals. **(Q7)**
  - Does the error voltage converge or diverge?
  - In general, is the PLL able to track the input phase when there is a large frequency error?

⇒ **Close 'pllbasic.mdl'**

1.2 The Costas PLL for Demodulating DSB Signals

The basic PLL in part 1.1 showed how, with a VCO and a feedback loop, the phase error between two signals can be reduced to a minimum. To make use of the basic PLL, the output of the VCO (once phase lock has been achieved) can be shifted 90° and then used in a coherent demodulation scheme. As long as the incoming signal has a carrier component (like an AM signal) then feeding it into the PLL will result in the VCO output being locked on to the carrier phase. (Picture an AM signal in time domain. It is just the carrier, but with changing amplitude). However, what if there is no carrier component, like with DSB? Recall that the DSB signal experience a phase change of 180° every time the message envelope crosses zero. This makes it impossible for the basic PLL from 1.1 to track it, but don't worry, your good friend J.P. Costas solved this problem back in the 50's with the Costas Loop.

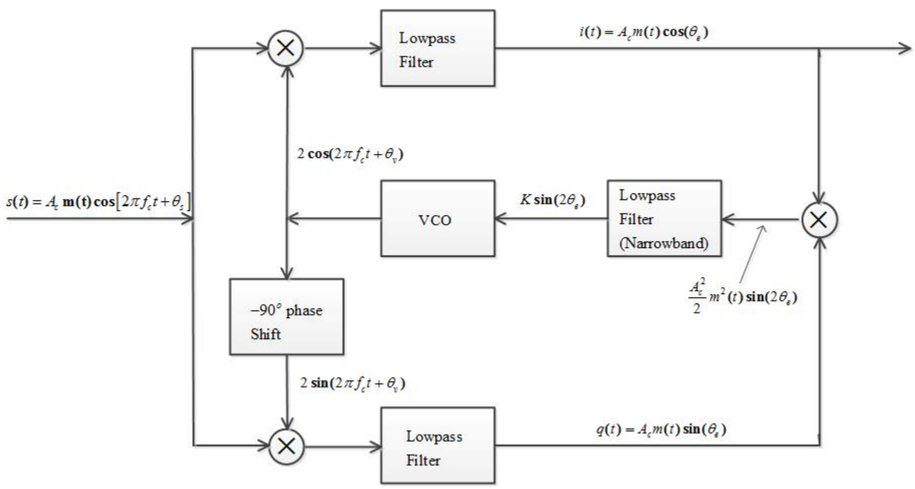

Figure 2: Costas Loop block diagram.

By using the trig functions given at the beginning of the lab, the Costas Loop diagram shown in Fig. 2 can be analyzed to prove that the in-phase output is $i(t) = A_\mathrm{c}m(t)\cos(\theta_\mathrm{e})$, while the quadrature is $q(t) = A_\mathrm{c}m(t)\sin(\theta_\mathrm{e})$. The terminology in-phase and quadrature often refers to orthogonal elements of a signal, which applies in this case because cosine and sine are orthogonal.

Since (by examining the diagram further) it can be seen that the input to the VCO is proportional to the phase error between the generated sinusoid and the incoming signal, then it can be expected that as long as the natural frequency of the VCO is close to the incoming frequency, the VCO will adjust until the phase error is approximately zero, just like with the basic PLL scheme. When this happens, the in-phase output is approximately $i(t) = A_\mathrm{c}m(t)$ and the quadrature output is approximately zero.

⇒**Open 'costas.mdl'**.

Look it over to be sure you understand what it is doing, (or at least how it relates to our block diagram). Do not worry about the delay blocks. Double-click on the DSB modulator to see what the carrier frequency, modulation index; message frequency and initial phase error have been set to. Notice the stylish mustard and relish colored scopes (condiments are so hot for 2015). Think about what you would expect to see on each scope as the PLL operates from time zero until it has locked on.

- Run the simulation. (This may take a couple seconds-don't worry you only have to do it once)
- Open the VCO input scope to verify that the PLL locked on. For SCOPES # 1-4, comment on how they changed from the beginning to the end of the simulation with reference to the VCO input scope. Don't get too detailed, just the big picture based on what you would expect. Hint: For Scopes # 1&2 you are interested in magnitude only **(Q8)** while for Scopes # 3&4 you are interested in phase only. **(Q9)**
- Why is the initial magnitude of the VCO input around 0.14? **(Bonus Marks)**

⇒**Close 'costas.mdl'**.

2 Superheterodyne Receiver

In Lab #1, the concept of Radio Frequency (RF) and Intermediate Frequency (IF) were introduced in the SSB section. Recall that the purpose was to achieve very efficient, narrow filtering on an analogue signal and it was easier to do this at an intermediate frequency where accurate equipment is cheaper and simpler to design. This is also one of the main purposes behind the Superheterodyne ('superhet' if you want to get on a first name basis) receiver.

When receiving an AM signal in an application such as radio, the signal received by the antenna will not be just the desired signal, it will be all the signals that have been sent over the channel. Therefore, the desired signal must be separated from the others. To make things more complex, a useful receiver would need to be tunable so that you could receive different signals (switch between radio stations). At the high radio frequencies which the signals are received at, a tunable and very narrow bandpass filter would be very difficult to design and build. These problems are solved by the Superhet.

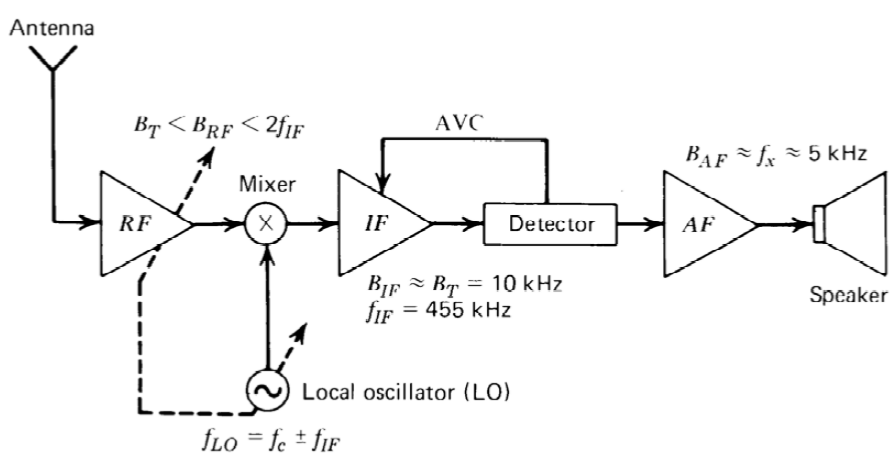

Figure 3: Superheterodyne block diagram.

The Superhet receiver has a tunable oscillator as well as tunable passband filter. The bandwidth of the filter can be fairly wide, as will be discovered in the simulations. The tunable RF filter and oscillator are coupled in such a way that tuning one automatically tunes the other. The objective of them being coupled is to relocate the desired signal to the IF frequency, which never changes. The frequency of the RF oscillator can be either $f_c - f_{IF}$ or $f_c + f_{IF}$ (where f_c is the carrier of the desired signal to be demodulated): either one will place a copy of the signal at f_{IF}.

Since the IF frequency is lower than the RF frequency, and is always the same, an IF filter can be designed that is very narrow and passes only the bandwidth of the desired signal. At this point the signal is ready to be demodulated. For our simulation we will do the final demodulation using direct coherent demodulation (which is just another mixer and filter stage, that is, the simple demodulator from Lab #2).

⇒**Open 'AMsuperrx.mdl'**.

Look it over and understand how it relates to the Superhet that was just described. One main difference is that in hardware, the tuning capabilities of the RF filter and oscillator would be continuous, allowing the radio operator to select any signal for detection. The '**AMsuperrx.mdl**' file has been simplified so that there are only two discrete tuning options: the signal being captured can be chosen as the signal on the 2 kHz carrier or the signal on the 5 kHz carrier. The IF for this system is 1 kHz.

- Check that the RF section is set to detect the 5 kHz signal. Run the simulation and look over all the spectra that pop up, noting which stage of the receiver each correspond to. **(Q10)**
  - How many spectra do you see at the RF filter input (from the channel)?
  - How many spectra do you see after RF filtering (input to IF filter)?
  - Why is it okay that the RF filter captures more than just one signal of interest?
- Change the 'knob' so that the 2 kHz signal is being selected and run the simulation. Did it work even though the IF filter and the demodulator section was not changed? **(Q11)**
- Double-click on the 'Tunable RF Oscillator'. Notice that the two different detection oscillators have been set to $f_c - f_{IF}$. To verify that $f_c + f_{IF}$ would also work, change the oscillator that detects the 2 kHz signal to 3 kHz instead of 1 kHz. (Run it and check!)
- Leaving the one RF oscillator at 3 kHz and the knob set for the 2 kHz signal, double-click the manual switch on the main AMsuperrx.mdl window that will allow the RF filter to be skipped.
- Run the simulation without the RF filter in the loop and look at what happens to the spectra! This is why even though the RF filter does not filter very tightly; it still needs to be there. Explain what happened to the demodulated signal due to the lack of the RF filter. **(Q12)**

⇒**Close 'AMsuperrx.mdl'**.

3 Introduction to FM

So far the labs have only dealt with Amplitude Modulation (AM). We took a carrier and we altered its amplitude to contain a message signal. By contrast, Angle Modulation alters what is inside of the brackets of the carrier sinusoid. Therefore, there are two types of angle modulation: Frequency Modulation (FM) and Phase Modulation (PM).

Therefore, given the signal $A(t)\cos[2\pi \mathrm{f_c t} + \phi(\mathrm{t})]$:

- **AM** alters $A(t)$
- **PM** alters $\phi(t)$
- **FM** alters $\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}$

However, since frequency and phase are related by $\Delta f(t) = \frac{\mathrm{d}\phi(t)}{2\pi \mathrm{d}t}$, the generation of FM and PM are similar. In both cases, the message is added as a time-varying phase but for FM, it is integrated first. The formula for FM signal is given by (1), where $m(t)$ is the message.

$$S_{\mathrm{FM}} = A_{\mathrm{c}}\cos[2\pi \mathrm{f_c t} + 2\pi \mathrm{k_f} \int_0^{\mathrm{t}} \mathrm{m}(\tau)\, \mathrm{d}\tau] \tag{1}$$

Note: *the frequency sensitivity (or frequency deviation constant* k_{f}*) effectively controls the depth (or 'strength') of the modulation. A higher sensitivity means a higher deviation from the carrier frequency. Also, in the above equation it must be in units of Hz/V or Amps/V.*

Let's not get caught up in the math too much for the purpose of this lab though. Just understand that the above equation is adding the message to the carrier frequency with a weighting of k_{f}. So, if the message is zero, then, $S_{\mathrm{FM}}(t)$ is just a cosine with frequency f_{c}. When the message gets larger than zero, the frequency will increase by $k_{\mathrm{f}}m(t)$. Now, this is exactly the operation of a VCO! So the simplest way to generate an FM signal, known as direct modulation, is by just inputting the message signal into a VCO that has a natural frequency of the desired carrier.

⇒**Open 'directFMl.mdl'**.

Double-clicking on the VCO will reveal that the carrier frequency for our FM signal is 1600 Hz. The 'input sensitivity' is the constant k_{f}. Double-clicking on the 'Message Signal Generator' will reveal the two signals that can be chosen via the manual switch to be supplied to the VCO.

- Run the simulation with Signal 1 selected. Look at the time and frequency domain of the FM signal. Explain why it is what it is. (Sketch it if it helps you describe it, but it's not necessary). **(Q13)**
- Switch the message to Signal 2 and run the simulation again. Look at the time and frequency domain again. Note: the frequency spectrum has 'persistence' on. This means that the screen does not get cleared over time, so you can see what the spectrum

looked like at all times. If persistence wasn't on, you would have only seen one spike at a time.

- Comment on why the plots (time and frequency) look the way they do with Signal 2 as the input. Be sure to check out the time domain at around 0.5 seconds and 1 second. (Again, feel free to sketch the plots if it helps your explanation). **(Q14)**

⇒**Close 'directFMl.mdl'**.

Of course in reality, the message signal is not going to be a constant value. When the message signal begins to constantly change with time, the frequency spectrum becomes very difficult to predict for FM. For this reason, we will return to our trusty tone modulation scenario where the message is a sinusoid and the math is not too impossible.

In Lab #4 a method for calculating the exact spectrum of an FM tone modulated signal will be analyzed and it will also be covered in your class lectures at some point. For now, we will focus exclusively on the bandwidth of FM signals and prove something called Carson's Rule.

Definition 1: Carson's Rule says that the bandwidth (in Hz) of an FM signal is as follows:

$$\begin{aligned} BW &= 2(\Delta f_{\mathrm{max}} + B) && (2) \\ &= 2B(\beta_{\mathrm{f}} + 1) && (3) \end{aligned}$$

where

- $\beta_{\mathrm{f}} = \dfrac{\Delta f_{\mathrm{max}}}{B} = \dfrac{k_{\mathrm{f}}|m(t)|_{\mathrm{max}}}{B}$
- B= bandwidth of $m(t)$
- β_{f} is the deviation ratio for the case of an arbitrary message signal, and can also be called the modulation index when the message is a sinusoid (or tone modulation).
  - If $\beta_{\mathrm{f}} \ll 1$, the signal is called Narrow-band FM and the bandwidth is BW=2B.
  - If $\beta_{\mathrm{f}} \gg 1$, the signal is called Wide-band FM and the bandwidth (by Carsons rule) is $BW = 2B\beta_{\mathrm{f}}$.

⇒ **Open 'directFM2.mdl'**.

This file is the same as the 'directFMl.mdl' file except that now there is a sinusoidal input as the message. Double clicking on the 'sinusoid message' block will display the frequency and amplitude of the message signal. Double clicking on the VCO will display the natural frequency and the input sensitivity k_{f}.

- What is the current value of β_{f}?
- Run the simulation. Since β_{f} is neither way larger nor way smaller than 1, the extremes of Carson's Rule do not apply. The spectrum is pretty complicated looking and the bandwidth cannot be approximated by $BW = 2\beta_{\mathrm{f}}B$ or $BW = 2B$. It is pretty close to $BW = 2B(\beta_{\mathrm{f}} + 1)$ though.

- Set k_{f} to 30 and the frequency of the message sinusoid to 300 Hz. What is the new value of β_{f}, and is the signal narrow-band or wide-band? **(Q15)**
- Run the simulation. What is the bandwidth? Show that it does or does not agree with Carson's Rule. **(Q16)**
- Set k_{f} to 2000 and the frequency of the message sinusoid to 20 Hz. What is the new value of β_{f}, and is the signal narrow-band or wide-band? **(Q17)**
- Run the simulation. What is the bandwidth? Show that it does or does not agree with Carson's Rule. **(Q18)**
- What other type of modulation did the narrow-band FM spectrum look like? **(Q19)**
- How does wide-band FM compare to Amplitude modulation with respect to bandwidth? **(Q20)**

⇒ **Close 'directFM2.mdl'**.

Done! Hand in your answer sheet to a TA and Go!

Part 2

Lab 3 Questions

Answer the following questions. Write down the names and ID # of the group members and hand in the answer sheets to the TAs before you leave the lab.

Simulation (Simulink) Questions:

Phase-Lock Loop (PLL): 'pllbasic.mdl'

Q1 Phase-lock test (input phase $\phi = \frac{\pi}{3}$) :

- i Does the error voltage converge?
- ii If it does, then what value does it converge to?
- iii Approximately, when does it appear to converge?

Q2 Phase-lock test (input phase $\phi = \frac{\pi}{3}$) : Compare the phase shift between the tracking and the input signals.

- i What is the phase difference at the beginning?
- ii Approximately, when does phase-lock happen (i.e., when does the phase difference become $\frac{\pi}{2}$)?
- iii Do convergence of the error voltage and phase-lock occur at the same time?

Q3 Effect of Amplifier gain (set gain to 2 and 0.5):

- i Which gain produces a faster convergence (or lock-on)?
- ii In general, which gain do you conclude would produce faster convergence? high gain or low gain?

Q4 Effect of small frequency error on PLL (gain= 1, $\phi = 0$, f_c=500.01 Hz):

- i Is there a phase-lock (does the error voltage converge) for a small frequency error?
- ii Did the error converge to zero? What did it converge to and why?

Q5 Can you say that the phase was successfully tracked?

Q6 What happened to the error voltage when the loop amplifier gain is set to 2 and the input phase is set to 0?

Q7 Effect of large frequency error on PLL (gain= 1, $\phi = 0$, f_c= 501, 502, 505 Hz)

i Does the error voltage converge or diverge?

ii In general, is the PLL able to track the input phase when there is a large frequency error?

Costas Loop: 'costas.mdl'

Q8 Costas loop test (in-phase output $\cos(\theta_e)$ 'Scope #1' and quadrature output $\sin(\theta_e)$ 'Scope #2'):

i (Scope #1) Approximately, at what value does $\cos(\theta_e)$ start from and to what value does it reach in steady-state?

ii (Scope #2) Approximately, at what value does $\sin(\theta_e)$ start from and to what value does it converge?

Q9 Convergence or lock-on time (Scope #3 and Scope #4):

i (Scope #3) Approximately, at what time does the input VCO reach steady state (or converge or lock-on state)?

ii (Scope #4) Approximately, at what time do the modulated input and the VCO output reach a lock-on state (i.e., when the phase difference reaches $\frac{\pi}{2}$)?

- Why is the initial magnitude of the VCO input around 0.14? (Bonus Marks)

Superheterodyne Receiver: 'AMsuperrx.mdl'

Q10 RF filter set to detect f_c =5 KHz, f_{LO} =4 KHz and f_{IF} =1 KHz:

i How many spectra do you see at the RF filter input (from the channel)?

ii How many spectra do you see after RF filtering (input to IF filter)?

iii Why is it okay that the RF filter captures more than just one signal of interest?

Q11 RF filter set to detect $f_c = 2$ KHz:
Did the demodulation work even though the IF and the rest of the receiver were not changed?

Q12 What happened to the demodulated signal without the RF filter?

Narrowband FM Modulation: 'directFM1.mdl'

Q13 Explain why the plots (time and frequency) look the way they do with Signal 1 as the input. (Sketch it if it helps you describe it, but it's not necessary)

Q14 Comment on why the plots (time and frequency) look the way they do with Signal 2 as the input.

Wideband FM Modulation: 'directFM2.mdl'

Q15 i What is the value of β_{f}, when k_{f} is set to 30 and the frequency of the sinusoid is set to 300 Hz?

ii Is the signal narrow-band or wide-band?

Q16 i What is the bandwidth?

ii Does it agree with Carson's Rule?

Q17 i What is the value of β_{f}, when k_{f} is set to 2000 and the frequency of the sinusoid is set to 20 Hz?

ii Is the signal narrow-band or wide-band?

Q18 i What is the bandwidth?

ii Does it agree with Carson's Rule?

Q19 What other type of modulation did the narrowband FM spectrum look like?

Q20 How does wideband FM compare to Amplitude modulation with respect to bandwidth?