Name: ____________________

Class: ____________________

19 - DSSS modulation and demodulation

Experiment 19 – DSSS modulation and demodulation

Preliminary discussion

Recall that when a sinusoidal carrier is DSBSC modulated by a message, the two signals are multiplied together. Recall also that the resulting DSBSC signal consists of two sets of sidebands but no carrier (refer to the preliminary discussion of Experiment 5 for a fuller discussion of this).

When the DSBSC signal is demodulated using product detection, both sidebands are multiplied with a local carrier that must be synchronised to the transmitter's carrier (that is, it has the same frequency and phase). Doing so produces two messages that are in-phase with each other and so add to form a single bigger message (refer to the preliminary discussion of Experiment 7 for a fuller discussion of this).

Direct sequence spread spectrum (DSSS or often just "spread spectrum") is a variation of the DSBSC modulation scheme with a pulse train (called a *pseudo-noise* sequence or just PN sequence) for the carrier instead of a simple sinewave. This may sound radical until you remember that pulse trains are actually made up of a theoretically infinite number of sinewaves (the *fundamental* and *harmonics*). That being the case, spread spectrum is really the DSBSC modulation of a theoretically infinite number of sinusoidal carrier signals. The result is a theoretically infinite number of pairs of tiny sidebands about a suppressed carrier.

In practice, not all of these sidebands have any energy of significance. However, the fact that the message information is distributed across so many of them makes spread spectrum signals difficult to deliberately interfere with or "jam". To do so, you would have to upset a significant number of the sidebands which is difficult considering their number.

Spread spectrum signals are demodulated in the same way as DSBSC signals using a product detector. Importantly, the product detector's local carrier signal must contain all the sinewaves that make up transmitter's pulse train at the same frequency and phase. If this is not done, the tiny demodulated signals will be at the wrong frequency and phase and so they won't add up to reproduce the original message. Instead, they'll produce a garbage signal that looks like noise.

The only way to obtain the right number of sinewaves at the right frequency and phase at the receiver is to use a pulse train with an identical sequence to that used by the transmitter. Moreover, it must be synchronised. This issue gives spread spectrum another of its advantages over other modulation schemes. The transmitted signal is effectively encrypted.

Of course, with trial and error it's possible for an unauthorised person to guess the correct PN sequence to use for their receiver. However, this can be made difficult by making the sequence longer before it repeats itself (that is, by making it consist of more bits or *chips*). Longer sequences can produce more combinations of unique codes which would take longer to guess using a trial and error approach. To illustrate this point, an 8-bit code has 256 combinations while a 20-bit code has 1,048,575 combinations. A 256-bit code has 1.1579×10^{77} combinations. That's 11579 with 73 zeros after it!

Increasing the sequence's chip-length has another advantage. To explain, the total energy in a spread spectrum signal is distributed between all of the tiny DSBSC that make it up (though not evenly because not all of the sinewaves that make up the carrier's pulse train are not the same amplitude). A mathematical technique called *Fourier Analysis* shows that the greater the number of chips in a sequence before repeating, the greater the number of sinewaves of significance needed to make it.

That being the case, using more chips in the transmitter's PN sequence produces more DSBSC signals and so the signal's total energy is distributed more thinly between them. This in turn means that the individual signals are many and extremely small. In fact, if the PN sequence is long enough, all of these DSBSC signals are smaller than the background electrical noise that's always present in free-space. This fact gives spread spectrum yet another important advantage. The signal is difficult to detect.

Spread spectrum finds use in several digital applications including: CDMA mobile phone technology, cordless phones, the global positioning system (GPS) and two of the 802.11 wi-fi standards.

The experiment

In this experiment you'll use the Emona Telecoms-Trainer 101 to generate a DSSS signal by implementing its mathematical model. You'll then use a product detector (with a stolen carrier) to reproduce the message. Once done, you'll examine the importance of using the correct PN sequence for the local carrier and the difficulty of jamming DSSS signals.

It should take you about 50 minutes to complete this experiment.

Equipment

- Emona Telecoms-Trainer 101 (plus power-pack)
- Dual channel 20MHz oscilloscope
- two Emona Telecoms-Trainer 101 oscilloscope leads
- assorted Emona Telecoms-Trainer 101 patch leads

Procedure

Part A - Generating a DSSS signal using a simple message

As DSSS is basically just DSBSC with a pulse train for the carrier instead of a simple sinusoid, it can be generated by implementing the mathematical model for DSBSC.

1. Gather a set of the equipment listed on the previous page.

2. Set up the scope per the instructions in Experiment 1. Ensure that:

    - the *Trigger Source* control is set to the *CH1* (or *INT*) position.
    - the *Mode* control is set to the *CH1* position.

3. Set the scope's *Trigger Source Coupling* control to the *HF REJ* position.

4. Locate the Sequence Generator module and set its dip-switches to 00.

    Tip: To do this, push both switches up.

5. Connect the set-up shown in Figure 1 below.

    Note: Insert the black plugs of the oscilloscope leads into a ground (*GND*) socket.

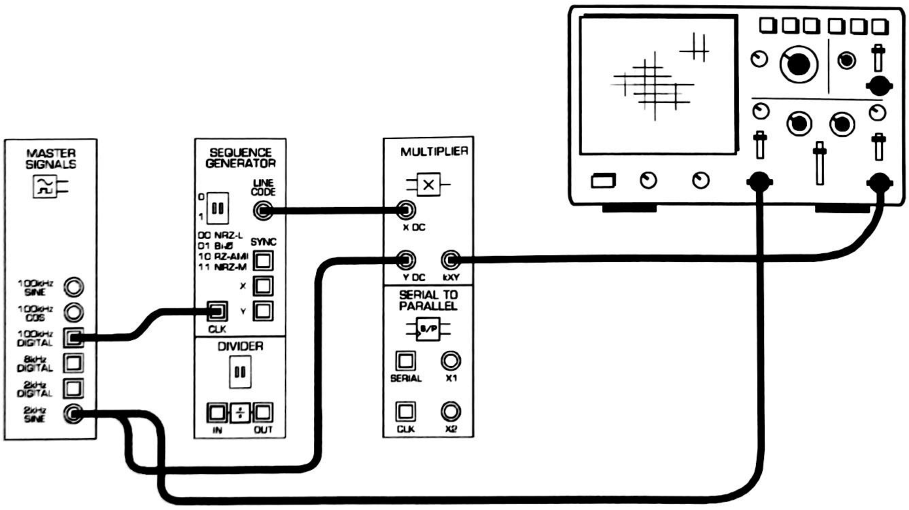

Figure 1

The set-up in Figure 1 can be represented by the block diagram in Figure 2 below. It multiplies the 2kHz sinewave message with a PN sequence modelled by the Sequence Generator's 32-bit pulse train output.

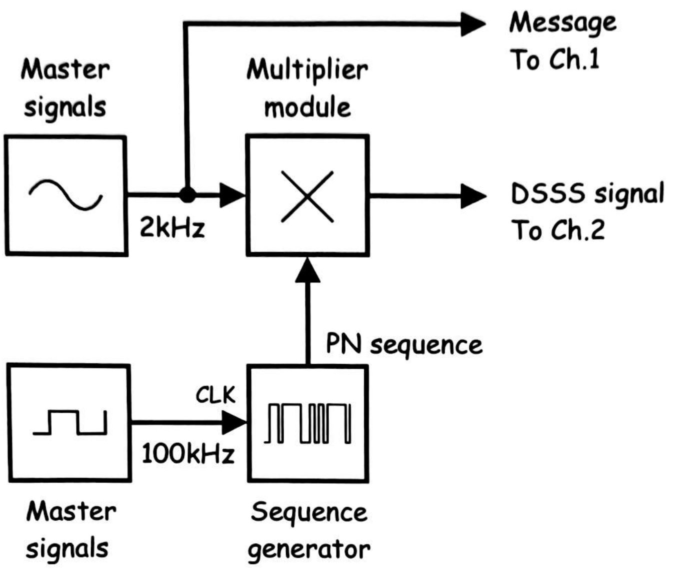

Figure 2

6. Adjust the scope's *Timebase* control to view two or so cycles of the Master Signals module's *2kHz SINE* output.

7. Set the scope's *Mode* control to the *DUAL* position to view the DSSS signal out of the Multiplier module as well as the message signal.

8. Adjust the scope's *Vertical Attenuation* controls to the appropriate settings for the signals.

9. Draw the two waveforms to scale in the space provided on the next page leaving room to draw a third waveform.

   Tip: Draw the message signal in the upper third of the graph and the DSSS signal in the middle third.

Ask the instructor to check
your work before continuing.

10. Use the scope's Channel 1 *Vertical Position* control to overlay the message with the DSSS signal's envelopes and compare them.

Question 1

What feature of the Multiplier module's output suggests that it's basically a DSBSC signal? **Tip:** If you're not sure, turn the scope's intensity control up and read the preliminary discussion for Experiment 5.

____

____

Question 2

Why is the DSSS signal so large when it's supposed to be small and indistinguishable from noise? **Tip:** If you're not sure, see the preliminary discussion for this experiment.

____

____

☑ Ask the instructor to check your work before continuing.

Part B - *Generating a DSSS signal using speech*
So far, this experiment has generated a DSSS signal using a sinewave for the message. The next part of the experiment lets you see what a DSSS signal looks like when modulated by speech.

11. Disconnect the plugs to the Master Signals module's *2kHz SINE* output.

12. Connect them to the Speech module's output as shown in Figure 3 below.

    Remember: Dotted lines show leads already in place.

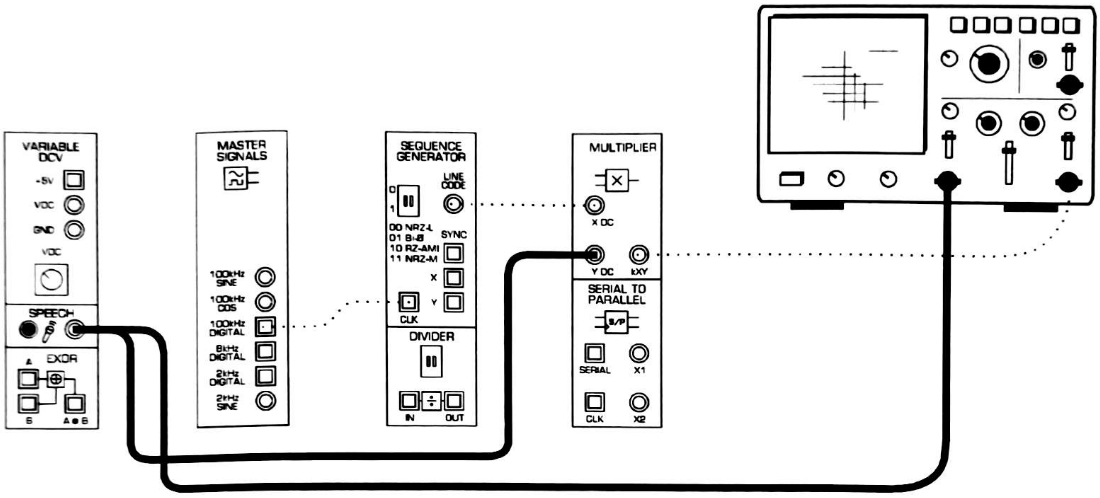

Figure 3

13. Set the scope's *Timebase* control to the *2ms/div* position.

14. Talk, sing or hum while watching the scope's display.

Question 3
Why isn't there any signal out of the DSSS modulator when you're not talking, etc?

____

____

Ask the instructor to check your work before continuing.

Part C – Using the product detector to recover the message

15. Return the scope's *Timebase* control to its original position (if you're not sure what it was, try the *0.1ms/div* position).

16. Locate the Tuneable Low-pass Filter module and set its *Gain* control to about the middle of its travel.

17. Turn the Tuneable Low-pass Filter module's *Cut-off Frequency Adjust* control fully anti-clockwise.

18. Disconnect the plugs to the Speech module's output and modify the set-up as shown in Figure 4 below.

    Note: Notice that the leads connect to the Multiplier module's *AC* inputs and not its *DC* inputs.

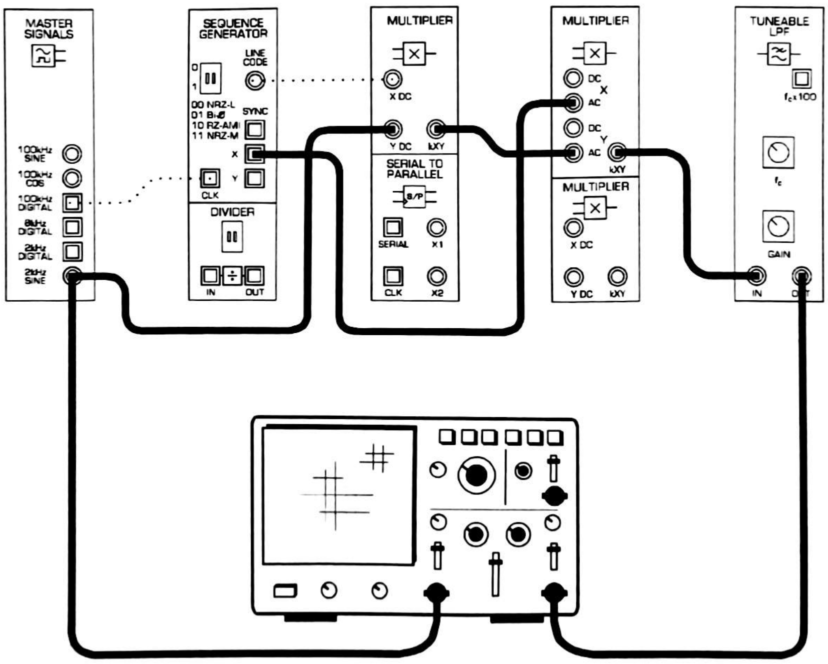

Figure 4

The additions to the set-up shown in Figure 4 can be represented by the block diagram in Figure 5 below. The Multiplier and *Tuneable low-pass Filter* modules implement a product detector which recovers the original message from the DSSS signal. To facilitate this, the PN sequence used for the modulator's carrier is "stolen" for the product detector's local carrier (though it's stolen from the module's *X* output but the sequence is the same).

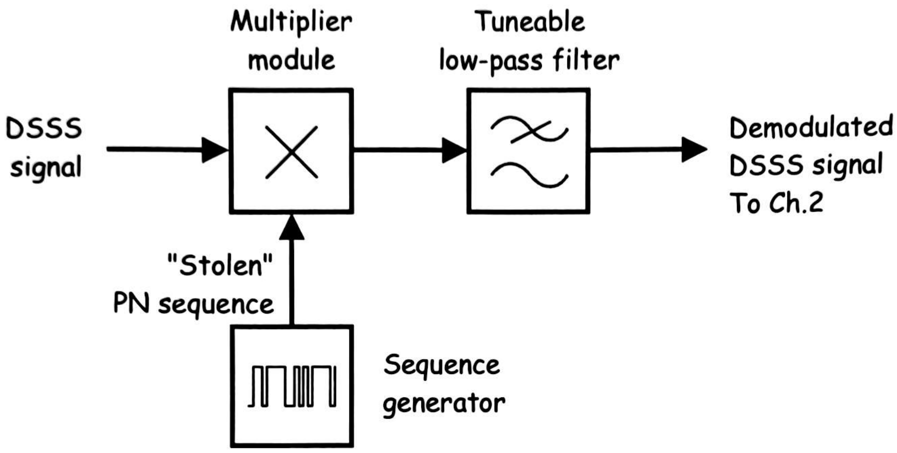

Figure 5

The entire set-up can be represented by the block diagram in Figure 6 below.

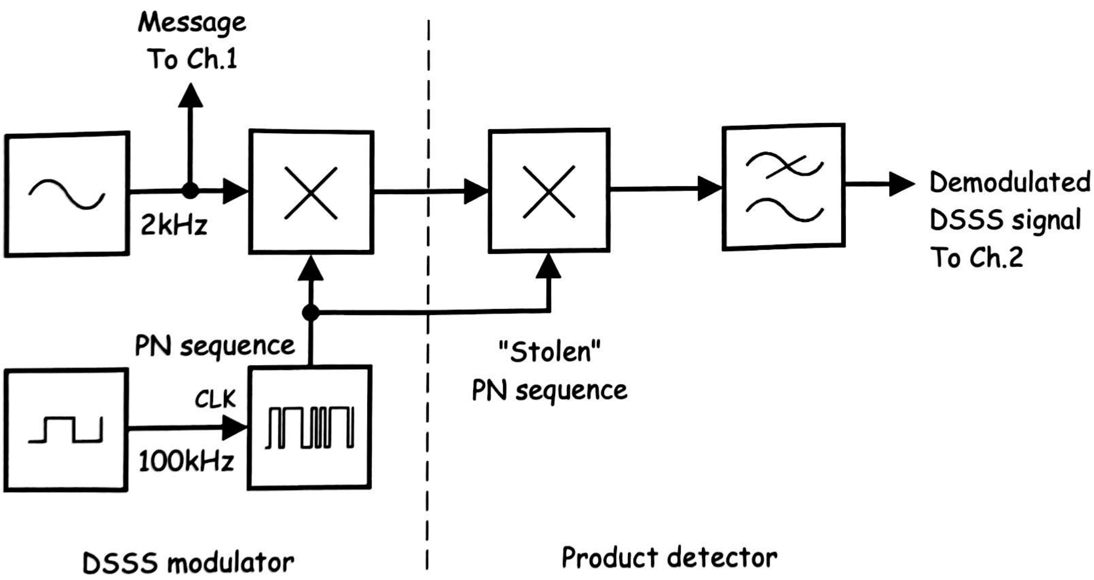

Figure 6

19. Slowly turn the Tuneable Low-pass Filter module's *Cut-off Frequency* control clockwise while watching the scope's display and stop when it's at about half its travel.

20. Draw the demodulated DSSS signal to scale in the space that you left on the graph paper.

Ask the instructor to check your work before continuing.

Recall that the message can only be recovered by the product detector if an identical PN sequence to the DSSS modulator's carrier is used. The next part of the experiment demonstrates this.

21. Modify the set-up as shown in Figure 7 below to make the demodulator's local carrier a different PN sequence to the transmitter's carrier.

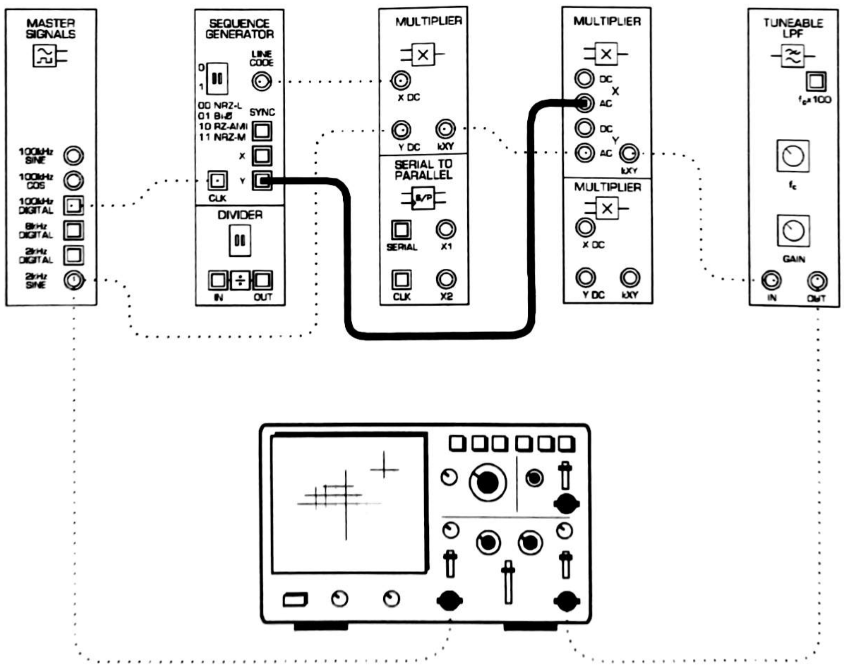

Figure 7

22. Compare the message with the product detector's new output.

Question 4

What does the signal out of the low-pass filter look like?

____

Question 5

Why does using the wrong PN sequence for the local carrier cause the product detector's output to look like this?

____

____

____

____

Ask the instructor to check your work before continuing.

Part D - DSSS and deliberate interference (jamming)

Interference occurs when an unwanted electrical signal gets added to the transmitted signal (typically in the channel) and changes it enough to change the recovered message. Electrical noise is a significant source of unintentional interference.

However, sometimes noise is deliberately added to the transmitted signal for the purpose of interfering or "jamming" it. The next part of the experiment models deliberate interference to show how spread spectrum signals are highly resistant to it.

23. Move the patch lead from the Sequence Generator's Y output back to its X output.

    Note: The product detector should now be recovering the message again.

24. Locate the VCO module and set its *Range* control to the *HI* position.

25. Set the VCO module's *Frequency Adjust* control to about the middle of its travel.

26. Locate the Adder module and turn its *g* control fully anti-clockwise.

27. Set the Adder module's *G* control to about the middle of its travel.

28. Modify the set-up as shown in Figure 8 below.

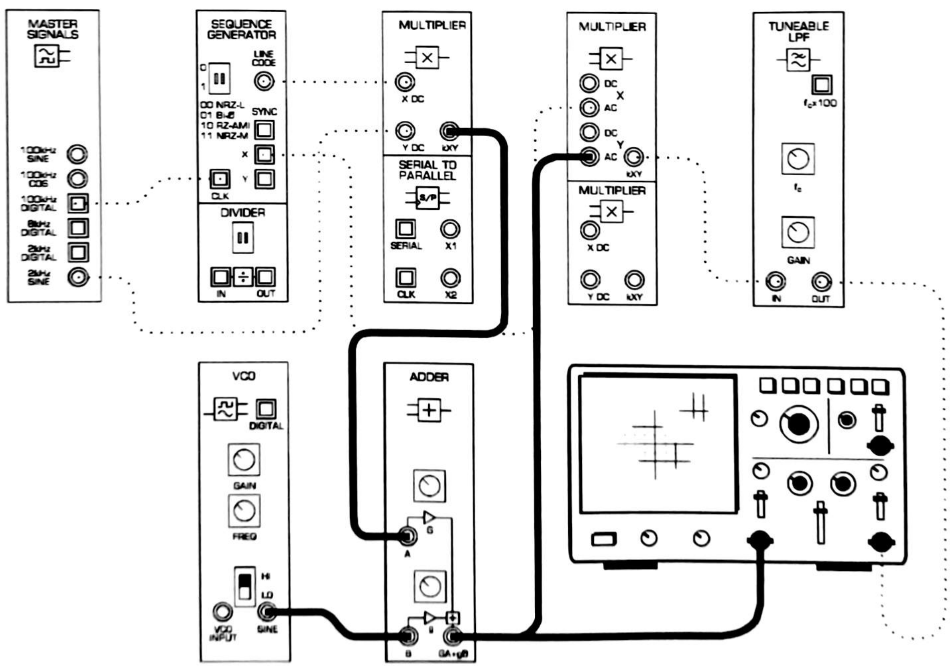

Figure 8

The set-up in Figure 8 can be represented by the block diagram in Figure 9 below. The VCO module is used to generate a variable frequency jamming signal that is added to the DSSS signal using the adder module.

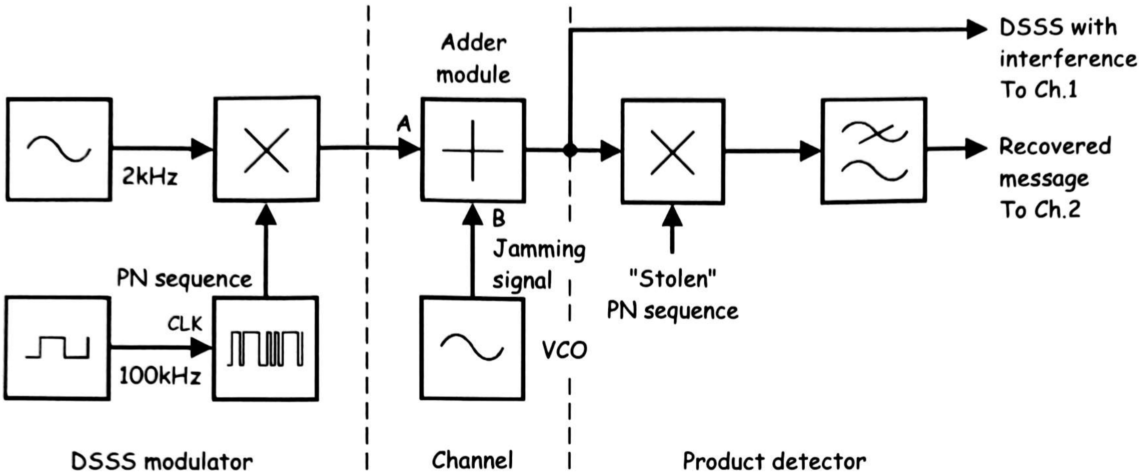

Figure 9

29. Set the scope's *Trigger Source* control is set to the *CH2* position.

    Note: The scope should be displaying a clean DSSS signal on Channel 1 like the one you drew earlier. The recovered message should be displayed on Channel 2.

30. Add the jamming signal to the DSSS signal by slowly turning the Adder module's *g* control clockwise. Stop when it's at about half its travel.

31. Compare the two signals and note the effect (if any) on the recovered message.

32. Vary the jamming signal's frequency by turning the VCO module's *Frequency Adjust* control left and right.

33. Note the effect (if any) on the recovered message.

34. Increase the size of the jamming signal to maximum by turning the Adder module's *g* control fully clockwise.

35. Note the effect (if any) on the recovered message.

Question 6

Why doesn't the jamming signal interfere with the recovery of the message?

A more sophisticated approach to jamming involves automatically sweeping the jamming signal through a wide range of frequencies to increase the chances of upsetting the transmitted signal. The next part of the experiment let's you see how spread spectrum handles this.

36. Return the Adder module's *g* control to about the middle of its travel.

37. Turn the VCO module's *Gain* control fully clockwise.

38. Modify the set-up as shown in Figure 10 below.

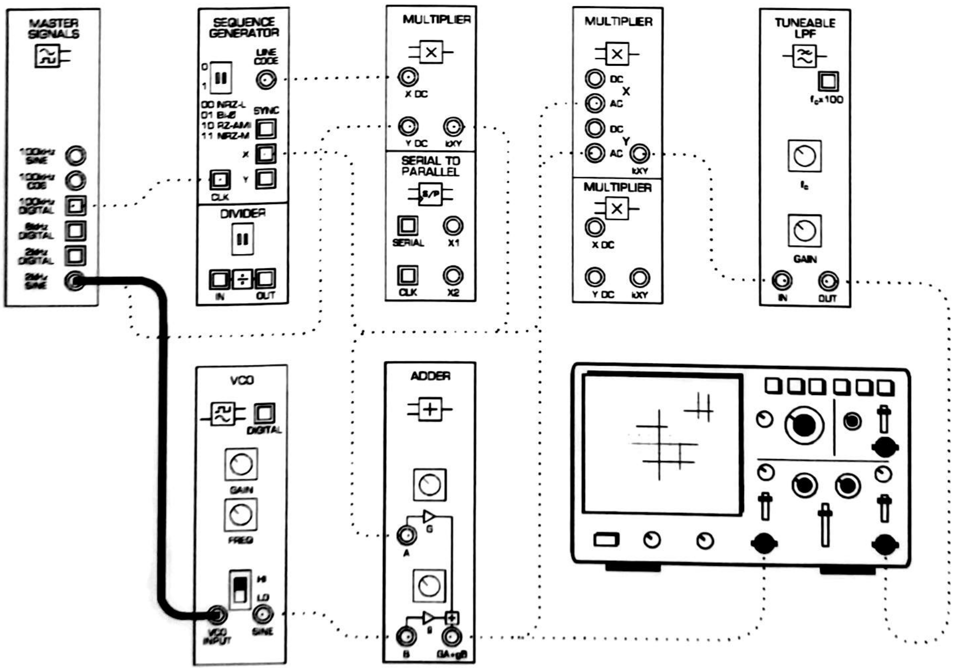

Figure 10

This modification forces the VCO module's output to sweep continuously through a wide range of frequencies.

39. Compare the two signals. Notice that the DSSS signal with interference is very distorted but the recovered message is only mildly affected.

Ask the instructor to check your work before continuing.

An even more sophisticated approach to jamming involves using many jamming signals at once (broadband jamming) to increase the chances of upsetting the transmitted signal. The next part of the experiment let's you see how spread spectrum handles this.

40. Modify the set-up as shown in Figure 11 below.

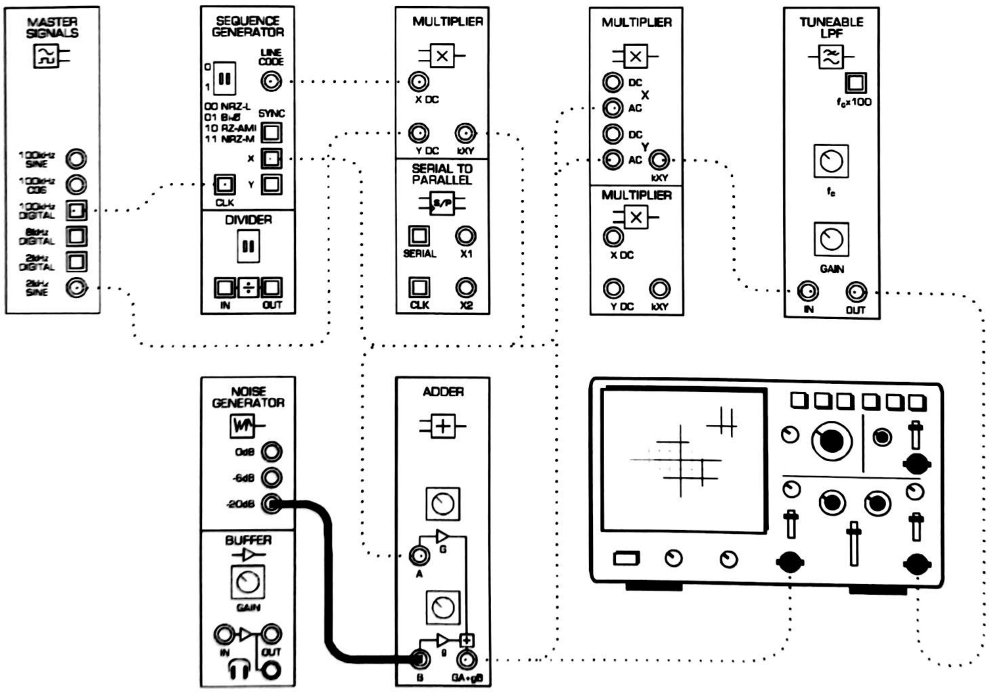

Figure 11

This modification uses the Noise Generator module to model a jamming signal that consists of thousands of frequencies.

41. Compare the two signals. Notice that the DSSS signal with interference is distorted but the recovered message is only mildly affected.

42. Increase the strength of the broadband jamming signal by connecting the Adder module's B input to the Noise Generator module's *-6dB* output.

43. Compare the DSSS signal and the recovered message.

44. Increase the strength of the broadband jamming signal even more by connecting the Adder module's B input to the Noise Generator module's *0dB* output.

45. Compare the two signals. Notice how distorted DSSS signal is but how little the recovered message is affected.

Ask the instructor to check your work before finishing.

If time permits...

If the instructor allows, let's see how DSSS performs when transmitting and receiving speech. You'll need a set of stereo headphones for this activity.

1. Remove the jamming signal by disconnecting the Adder module's B input from the Noise Generator module's *0dB* output.

2. Connect the Tuneable Low-pass Filter module's output to the Buffer module's input.

3. Turn the Buffer module's *Gain* control fully anti-clockwise.

4. Without wearing the headphones, plug them into the Buffer module's headphone socket.

5. Put the headphones on.

6. Adjust the Buffer module's *Gain* control until the 2kHz tone is a comfortable sound level.

7. Investigate what happens when the wrong PN sequence is used to demodulate the DSSS signal (like you did in Part C) by moving the patch lead from the Sequence Generator's X output to its Y output.

8. Return the patch lead from the Sequence Generator's Y output back to its X output.

9. Investigate what happens when a single sinewave is used to jam the DSSS signal (like you did in Part D) by connecting the VCO module's *SINE* output to the Adder module's B input.

10. Investigate what happens when a broad-band signal is used to jam the DSSS signal (like you did in Part D) by connecting the Noise Generator module's *-20dB* output to the Adder module's B input.

11. Repeat the step above for higher levels of jamming/noise by connecting the Noise Generator module's *-6dB* output to the Adder module's B input then the *0dB* output.

12. Substitute the 2kHz tone with the Speech module's output and investigate the effects on speech as you repeat steps 7 to 11 while talking, singing or humming.