GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2025 Fall 2001
Lab #9999: The FSK Challenge

Date: 1 Dec. 2001

Rules of the contest:

1. The first person to finish decoding the set of messages wins an exemption from the Final Exam. Your final grade would then be determined by your average going into the final.
2. The time of the first successful decoding will be determined by receipt of a special e-mail message to Dr. McClellan. When you decode the last message in the set, you will find instructions about sending this e-mail to confirm that you have finished.
3. Decoding alone is not sufficient to win. To be declared the winner, you must *also make an oral presentation* of how you did the signal analysis and decoding. Plots of signals and spectrograms will be helpful in making this (informal) presentation.
4. You can use any signal processing tricks that you think will work to analyze the signals, but in the end you should produce a working demodulator/decoder that is somewhat similar to the FSK system in Lab #12 (also shown in Fig. 1).
5. Hints and clarifications will be posted to the ECE-2025 WebCT bulletin board. Since this is a "challenge," don't expect many hints, but if you uncover a bug in the posted data, please let us know.
6. There will be more than one set of signals to attack. The first one is being posted on Sunday, 2-Dec. Another one will be posted a couple of days later.

This project is optional. However, if you find it interesting to work with these sorts of "unknown signals," there are a variety of employers who would find your skills valuable. Have fun.

1 Scenario

Imagine that a friend of yours has just purchased a brand new modem, the *Bulldog-300,* but when she starts to use it to connect to your modem at Georgia Tech you notice that all the messages appear to be jibberish. This new state-of-the-art modem is supposed to use the V.21 FSK standard for sending messages, but apparently has some design or implementation flaws (lawsuits are pending). Luckily, you have been able to record some of the raw signal data coming into your modem and you plan to use those signals to "reverse engineer" the *Bulldog-300* and modify your own modem to compensate for the design problems, and thus receive the messages.

One of the advanced features of the *Bulldog-300* is its built-in *encryption* capability that scrambles the text messages being sent. Therefore, your new modem will also have to be able to undo the encryption. Dr. Rad Bleck, designer of the *Bulldog-300*, decided to use the time-honored XOR encryption method that exploits a property of the exclusive-OR function, and works in the following manner:

(a) The XOR method needs a *key* which is a sequence of binary bits. In addition, the key could be a text phrase that has been turned into a binary stream. For example, if the key is "GT" then the binary sequence derived from 8-bit ASCII is `0100011101010100`. In this case, the key is 16 bits long.

(b) Assume that the message is a very long stream of bits and that the length of the key is K bits. The encryption is done in the following manner: Take the first K bits of the message stream and XOR with the key to produce the first K bits of the encrypted message. Then take the next K bits of the input stream, perform the XOR, and append these bits to the encrypted output stream. Continue until you reach the end of the message. If the last block of the message is smaller than K bits, then "pad" the input stream with blanks or with random bits to make the block length exactly K bits. Thus the final output size of the encoded bit stream will be an integer multiple of K.

(c) A simple property of the exclusive-OR (XOR) operator is that another XOR will recover the original message. Mathematically, we would write `XOR(XOR(msg,key),key) = msg`.

(d) In the *Bulldog-300,* only the message data bits are encrypted, not the preamble nor the trailer.

(e) Only one key is used to encode an entire message.

(f) The *Bulldog-300* also produces a *calibration tone* at the beginning of each modem call. This is supposed to be a 2100 Hz tone that is on for 250 millisecs. It comes before the preamble. It is possible use this tone to learn some parameters of the modem.

2 Modem Flaws

The *Bulldog-300* modem tries to implement the V.21 FSK standard, but the oscillator in the modem that produces sinusoids is low quality, so the frequencies for each bit are not what the V.21 standard calls for. Fortunately, the modem does produce the same signal frequency for each "0" bit in the encrypted message, and a different (but consistent) signal frequency for each "1" bit.

The oscillator deficiency also causes a problem with the actual bit rate. Even though the *Bulldog-300* modem is advertised as a 300 bps modem, it might not be running at that rate. Luckily, the oscillator is stable, so the bit rate does not change during a message.

Despite all these flaws, the basic blocks needed in the demodulator will still be those shown in Fig. 1, and used in Lab #12. However, the parameters of the system will have to be adjusted through "trial and error" to get a receiver that works.

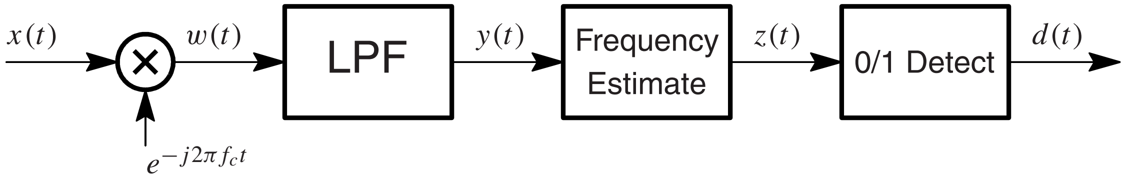

Figure 1: Block diagram of the FSK V.21 demodulator.

3 Encryption

There are several messages that must be demodulated and decoded to complete the FSK Challenge. The last message will contain information about validating the time when you have finished.

Since the messages are encrypted, the next part of this quest is to *decrypt* the messages. This requires that you know the encryption key. Each message was encoded with a different key. Whenever you decode one of the messages, it will contain a clue that will enable you to figure out the key that was used to encrypt the next message. In general, these clues should be solvable by searching the web for information, or by using your knowledge of DSP. To get you started, the first message was encoded with the key `JeffMiller`.

4 Recorded Signals Available

There are several recordings that are available for your analysis. A little bit of random noise has been added to each signal, because the "real world" always has some noise. In each case, the signal was sampled with an A/D converter whose sampling frequency is

$$\boxed{f_s = 9000 \text{ Hz}} \tag{1}$$

You can get the first signal from the ZIP file called

`FSK_1_start.zip`

which can be downloaded from WebCT. The ZIP file contains a MAT file, called `FSK_1_start.mat`. The MAT file contains one signal that was sampled at 9000 Hz.