Chapter 2

Synchronous Undersampling and the LazyFish

This chapter describes the LazyFish, a board I designed that implements 8 channels of Electric Field Sensing in a very small footprint. Its small footprint is made possible by my Synchronous Undersampling technique, which is explained in this chapter. Before doing so, I will review some other implementations of electric field sensing, starting with the Theremin.

2.1 Background

2.1.1 Theremin

Figure 2-1 shows a schematic of Clara Rockmore’s Theremin, drawn by Bob Moog in 1989. Here is a “back of the envelope” analysis of the sensing and pitch synthesis. Changes in hand proximity affect the capacitance in one LC resonant circuit (the one on the right side of the figure, with the labeled pitch antenna), changing its resonant frequency. Though not all component values are provided in this diagram, if we assume $L = 38\mu H$ and $C = 663pF$, the resonant frequency $f = \frac{1}{2\pi \sqrt{LC}} = 1MHz$ and $Q = \sqrt{\frac{L}{C}} = 240$. The sound is synthesized by demodulating this hand “detunable” signal $f_1$ with a fixed frequency carrier $f_2$. The fixed carrier is generated by the resonator on the left side of the schematic, and the mixing occurs in the vacuum tube in the center of the figure. The audible output frequency is the difference $f_1 - f_2$. With our estimated component values, producing a difference frequency of 20KHz (the upper frequency limit on human hearing) requires a change in capacitance of 27pF, which is somewhat large but not unreasonable estimate of how much a hand could change the capacitance. These figures are just ballpark estimates, but they are sufficient to illustrate the basic idea that a change in capacitance that could be produced by the motion of a hand near an electrode causes the difference frequency to span the entire human audible frequency range.

2.1.2 Classic Fish

Neil Gershenfeld built the “small box” shown at the top of figure 2-2, which was a hand-wired, all analog implementation of Electric Field Sensing that required an external analog-to-digital converter board in a PC. The Classic Fish, shown below the “small box,” was a
Figure 2-1: A schematic of Clara Rockmore’s Theremin, drawn by Bob Moog.
printed circuit board that had four analog channels of Electric Field Sensing, plus a Motorola 68HC11 8 bit microcontroller for analog to digital conversion and serial communication to the host PC or MIDI synthesizer. Following Neil’s initial guidelines, Joe Paradiso designed the analog portion of the Classic Fish, Tom Zimmerman did the digital portion, and I wrote the board’s firmware.

The Classic Fish had one transmitter, tunable (via a potentiometer) from 10kHz to 100kHz. Another pot controls transmit amplitude. Each of the four receive channels consists of a transimpedance gain stage, analog multiplier, and low pass filter gain stage. Four phase shifting circuits provide independently shifted versions of the transmitted signal to the multiphers, so that the received signal can be synchronously demodulated. The phase for each receive channel is hand adjusted with four potentiometers, to compensate for phase changes due to cable capacitance. The DC offset and gain of the final stages is adjustable via eight additional pots, to match the analog output to the useful working range of the ADC.

Below the Classic Fish in figure 2-2 is the Smart Fish, which was a brute force attempt to do the demodulation in software. Depending on the version, it had one or two transmitters and eight or nine programmable gain receive channels, a fast ADC, fast DSP, as well as a 68HC11. The Smart Fish was plagued by problems and never worked well.

The sensing portion of the LazyFish is shown at the bottom of the figure.
2.2 Synchronous Undersampling

This section explains Synchronous undersampling, the measurement technique used by the LazyFish. First I will explain ordinary synchronous detection, the method used by the Classic Fish.

2.2.1 Synchronous detection

Figure 2-4 illustrates traditional synchronous detection. A 100kHz carrier voltage is applied to the transmit electrode. A 100kHz current whose magnitude depends on the hand position is induced on the receive electrode. This signal is amplified in a transimpedance gain stage, and then mixed down to DC by an analog multiplier (with access to the original transmitted signal) followed by a low pass filter. As shown in figure 2-4, multiplying the received signal by the original transmitted signal produces sidebands at $+2f$ and $-2f$ as well as a DC value. The low pass filter eliminates these sidebands, and the amplitude of the remaining DC signal contains the desired information about the hand proximity.

Quadrature detection

If the phase of the received signal is unknown, it can additionally be demodulated with a $\frac{\pi}{2}$ phase shifted version of the transmitted signal. These two demodulated components (called the in phase and quadrature components) are a cartesian representation of the magnitude and phase of the received signal.

2.2.2 Synchronous sampling

Figure 2-5 illustrates synchronous sampling. If the carrier frequency is $f$, then the signal after multiplication (but before the low pass filter) has components at $+2f$ and $-2f$. By Nyquist’s theorem, it should be sufficient to sample this signal at $4f$.\(^1\)

The sampling operation can be viewed as multiplication with a train of delta functions. Since multiplication is commutative, we can also regard the received signal to have been sampled before the demodulation operation occurs. If the amplitude of the demodulating (former) cosine is 1, then its sampled version consists of a train of +1 and −1 height delta functions. If the low pass filter is replaced by a zero order hold (that is, if we simply add all the signal amplitude), then the demodulation operation becomes simply addition.

\(^1\)It could be argued that because the original signal can be reconstructed from a train of samples at $2f$ it should not be necessary to sample at $4f$. In fact, the $4f$ figure is simply a very natural sampling rate to consider in the context of quadrature demodulation; the point being developed in this chapter is that the signal can actually be sampled at much less than $2f$. 

34
Figure 2-4: Synchronous detection.
and subtraction of samples of the received signal. The analog multiplier and low pass filter have been eliminated. To implement quadrature detection, we perform the same sequence of operations on samples spaced 90 degrees apart, keeping two separate accumulation variables, one for the in phase channel and one for the quadrature channel. Note that this technique leaves harmonic windows open to noise at multiples of $2f$. For this reason, it is desirable to include a bandpass filter centered on $f$ in the front end gain stage, although this feature is not present in the (original) LazyFish.

### 2.2.3 Undersampling

Unfortunately, the ADC on a typical microcontroller such as the PIC16C711 is quite slow. On this microcontroller, the maximum sampling rate turns out to be about 40kHz. If we require our sampling frequency to be $4f$, where $f$ is the carrier frequency, then our carrier is limited to 10kHz. As explained in section 1, the received signal $I = 2\pi fCV$ is proportional to frequency. Thus signal to noise is linear in the frequency. But there is an additional advantage to moving to higher frequencies. As will be explained below, in practice the $V$ term (transmit amplitude) also scales linearly with $f$, so actually the signal to noise scales at least quadratically with frequency.

The transmitter is a series LC circuit with resonance at $f = \frac{1}{2\pi\sqrt{LC}}$. When it is driven on resonance, its steady state amplitude is given by the product of the driving amplitude
and the resonator’s quality factor \( Q = \frac{1}{R} \sqrt{\frac{L}{C}} \). Thus \( V \), the final transmitted amplitude, is given by \( V = vQ \), where \( v \) is the amplitude of the PIC’s driving signal. So, substituting in the expression for \( Q \), the final received current \( I = 2\pi f CVQ \).

In practice, \( L \) is a limited resource: 2.5mH was the largest inductance commonly available in a surface mount package at the time the LazyFish was designed. \( L \) should be as large as possible in order to maximize the \( Q \). If \( L \) is a constant (2.5 mH), then the only way to increase the \( Q \) is to go to a higher frequency. By manipulating the expressions for \( f \) and \( Q \), we can write \( Q \) in terms of \( f \):

\[
Q = \frac{2\pi L}{f} \tag{2.1}
\]

Because \( L \) is constant, the frequency \( f \) can be adjusted only by changing \( C \), not \( L \). With \( L \) constant, it is clear from equation 2.1 that \( Q \) is linearly proportional to frequency.

Now we can rewrite the expression for the received current for the last time

\[
I = (2\pi f)^2 LC \frac{v}{R}
\]

It is clear from this expression that signal strength scales quadratically with frequency. Thus if we could find a way to move our synchronous sampling scheme from 10kHz to 100kHz, it would increase the received signal strength by a factor of at least 100.

Synchronous undersampling allows us to do precisely this. Although the ADC on the PIC16C711 is slow to convert voltages to bits, the ADC has a built in sample and hold with a much shorter aperture time. Furthermore, this sampling aperture can be placed very precisely in time. So while the actual conversion operation apparently limits us to sampling 10kHz signals, in reality the PIC can sample much higher frequency signals as long as they are repetitive. This makes sense if we consider the fact that a short aperture that can placed precisely in time corresponds to a high input bandwidth.

Figure 2-6 shows the basic idea of synchronous undersampling. The PIC generates a squarewave burst that causes the resonator to ring up. Then it samples at a precisely known time—call it phase 0—and waits for the value to be converted. Then it starts over, generating a new burst, this time waiting a bit longer before sampling. In this way, 0 degree, 90 degree, 180 degree, and 270 degree samples of a 100kHz signal can be collected.

The spikes in figure 2-6 indicate roughly when the sample and hold was open. Figure 2-7 through 2-10 are “closeups” of figure 2-6. These figures show the 0 degree sample \( S_0 \), as well as \( S_{90} \), \( S_{180} \), and \( S_{270} \). Note that some additional immunity to noise with the sampling periodicity could be gained by pseudorandomly varying the time between samples. Another strategy worth considering would be pseudorandomly varying the order in which \( S_0 \), \( S_{90} \), \( S_{180} \), and \( S_{270} \) are taken.

As the samples are collected, the demodulation operation is performed by updating the inphase channel accumulator using \( I' = I + S_0 - S_{180} \), where \( I' \) is the new value and \( I \) is the old value. Similarly, the quadrature accumulator \( Q' = Q + S_{90} - S_{270} \). The PIC’s ADC returns 8 bit values, and the I and Q accumulators must be 16 bit variables because multiple samples are added together. The number of samples integrated is controlled in software, which allows signal to noise to be traded off with update rate. The LazyFish firmware computes an approximate magnitude from the I and Q components. The true value that should be returned is \( \sqrt{I^2 + Q^2} \); the approximate value returned by the LazyFish is \( |I| + |Q| \). As explained in chapter 3, the error of this approximation depends on the phase (0 error at 0
or 90 degrees, maximum error at 45 degrees). Since (unlike the School of Fish) the LazyFish is doing straightforward homodyne detection, with the receiver and transmitter absolutely synchronous, and since the phase offset between transmitter and receiver is constant in most applications, the approximation corresponds to a constant scale factor, and causes no problems.

2.2.4 Controlling gain by adjusting TX burst length

The LazyFish provides software programmable gain control, by making use of the resonator’s transient response. This feature is important because if the front end clips (due to a received signal that is too strong), the synchronous demodulation operation does not work correctly. The obvious but more complex approach would be to use programmable gain in the receiver. The LazyFish controls the transmit gain instead, leaving the receive gain fixed. This approach turns out to require simpler hardware.

To adjust the transmit gain, the LazyFish firmware controls the length (number of periods) of the square wave burst used to excite the resonator. By using a shorter burst, it can avoid exciting the resonator all the way. Figures 2-11 through 2-18 show progressively shorter bursts (starting with 20 and working down to 1). In a “continuous wave” (rather than pulsed) application, one can also imagine controlling the frequency or duty cycle of the square wave drive. In our “pulsed” situation, controlling the resonator amplitude by controlling the duration of the square wave burst is the simplest to implement.

Quenching the resonator as quickly as possible is also important to achieving a high update rate. After the sample has been taken, the output pin driving the resonator is put into a high impedance state, which causes the resonator to quench in just a few cycles. If the output pin was left at a fixed voltage, the resonator would continue to ring for much longer (20 cycles) because of its high Q.
Figure 2-7: 0 degrees.

Figure 2-8: 90 degrees.
Figure 2-9: 180 degrees.

Figure 2-10: 270 degrees.
Figure 2-11: Burst length: 20.

Figure 2-12: Burst length: 15.
Figure 2-13: Burst length: 10.

Figure 2-14: Burst length: 7.
Figure 2-15: Burst length: 5.

Figure 2-16: Burst length: 3.
Figure 2-17: Burst length: 2.

Figure 2-18: Burst length: 1.
2.3 The LazyFish

Because of the synchronous undersampling technique described in the first chapter, the LazyFish is able to implement 8 channels of electric field sensing using surprisingly little hardware. It has 4 resonant transmit channels and 2 receive front ends. All demodulation is performed in software on a PIC16C711 microcontroller, which has a built in analog-to-digital converter.

The LazyFish board consists of two initially integrated portions that may be split in two: the sensing portion and an RS-232 digital communication interface. When the LazyFish is used as a computer input device, it would typically be deployed in the integrated configuration, shown at the top of this page. When the LazyFish is built in to a handheld device, or connected to an RF transceiver or other device that uses TTL voltage levels, the sensing portion would be used without the communications portion. The sensing and RS-232 portions each have a stereo audio jack that can be used to connect the two halves using an ordinary stereo audio cable. The RS-232 interface portion can be connected permanently to the computer, and handheld LazyFish devices can be connected to the computer temporarily for debugging purposes through the stereo cable and RS-232 interface. This “remote sensor” configuration is shown below. For applications in which a smaller footprint RS-232 capable device is required, the stacked configuration, in which the two sections are mounted together, can be used. This configuration is also pictured below. Finally, the RS-232 interface may be used by itself to connect other TTL devices to a computer.
Figure 2-20: Stacked configuration.
Figure 2-21: LazyFish Schematic.