

# Code-Division Multiplexing of a Sensor Channel: A Software Implementation

Joshua R. Smith, Christopher Salthouse, and Neil Gershenfeld, *Member, IEEE*

**Abstract**—This paper demonstrates the use of software radio techniques in the context of sensing, rather than communications. It describes code-division multiplexing (CDMA) and time-division multiplexing (TDMA) of a receiver channel in an electric field sensing system. The only hardware used is a front-end gain stage consisting of two opamps and a microcontroller. The modulation and demodulation operations are implemented entirely in the microcontroller software. Multiple coded waveforms are transmitted simultaneously, and induce a combined signal on a single receive electrode. The combined signal, after passing through a single analog front end terminating in an analog-to-digital converter, is separated into the four original component signals by a software demodulation operation.

The signal-to-noise ratio (SNR) achieved by the code-division multiplexed system given a fixed measurement time is compared to the SNR achieved by a time-division multiplexed implementation given the same total measurement time. The paper also compares the scaling of TDMA and CDMA performance with the number of transmitted channels and the number of demodulated channels.

**Index Terms**—CDMA, electric field sensing, sensing, software radio, spread spectrum.

## I. INTRODUCTION

THIS paper demonstrates the use of software radio techniques in the context of sensing, rather than communications. It describes code-division multiplexing (CDMA) and time-division multiplexing (TDMA) of a receiver channel in an electric field sensing system. The demodulation operation is performed entirely in software. Several sensor signals can simultaneously share the same analog front-end hardware, including ADC, because software performs different processing operations on one set of samples to extract several distinct signals. This means that the number of channels that may simultaneously be received is not rigidly fixed by hardware. Additional receiver channels are extra processing steps on the data already being collected. Additional coded waveforms do appear as noise to the other channels (in the CDMA case), or require additional sensing time (for TDMA), so additional channels are not “free,” but the number of channels is not fixed by the hardware. As long as one is able to pay the price in either signal-to-noise ratio (SNR) or measurement time, and also in processing time, additional channels may

Manuscript received September 1997; revised June 19, 1998 and July 31, 1998.

J. R. Smith is with the Escher Laboratories, Cambridge, MA 02142 USA. C. Salthouse and N. Gershenfeld are with the Physics and Media Group, M.I.T. Media Laboratory, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0733-8716(99)02976-5.

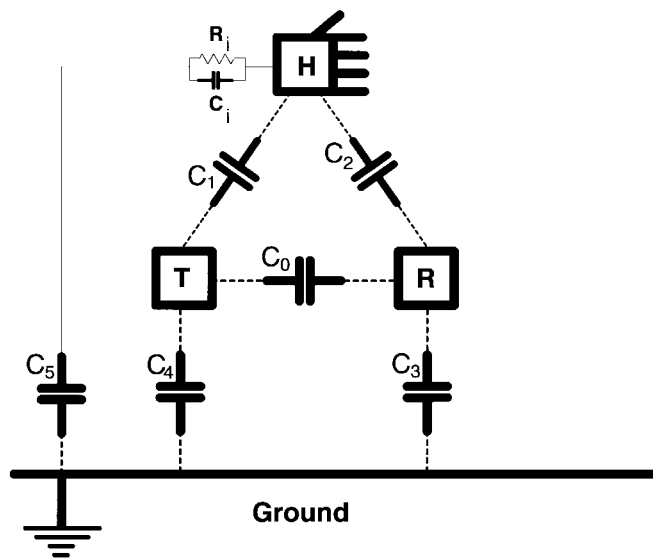


Fig. 1. Lumped circuit model of electric field sensing.

be added. Furthermore, the software receiver readily changes which particular channels it is demodulating.

The work is set in the context of electric field sensing, which is introduced below, but the same principles could be applied in virtually any sensing or measurement system in which the quantity being sensed is modulated by a carrier. For example, these techniques could be applied to systems of ultrasound or infrared emitters and detectors.

## II. MOTIVATION: ELECTRIC FIELD SENSING

The term electric field sensing refers to a family of non-contact methods for measuring the position and orientation of the human body, or parts of the human body such as a hand. Electric field sensing has been used for human-computer interface [1], to make a three-dimensional noncontact mouse [2], for creating new musical instruments (3-D) [3], and in the automotive industry as a solution to the rear-facing infant seat problem.<sup>1</sup>

In a typical implementation of electric field sensing, a low-frequency (from 10–100 kHz) voltage is applied to a transmit electrode (labeled *T* in Fig. 1), and the displacement current induced at a receiver *R* is synchronously detected. Fig. 1 shows a lumped circuit model of the electrodes and the body.

<sup>1</sup>Because the violent inflation of an airbag can injure infants in rear-facing infant seats, it is desirable to sense the orientation of the child and disable the airbag as appropriate [4].

The term capacitive sensing ordinarily refers to measuring the change in the loading of the transmitter as the hand approaches, increasing the value of  $C_1$ . There is no distinct receiver in this measurement, so  $C_1$  is the only significant capacitance in the problem. This type of measurement is called a *loading mode* measurement.

But the other current pathways in the diagram suggest other measurement techniques. In transmit mode, the transmitter is coupled strongly to the body— $C_1$  is very large—and the body is essentially at the potential of the transmitter. As the body approaches the receiver, the value of  $C_2$  (and  $C_0$ —the two are not distinct in this mode) increase, and the received signal increases.

*Shunt mode* measurements are most relevant to this paper. In the shunt mode regime,  $C_0$ ,  $C_1$ , and  $C_2$  are of the same order of magnitude. As the hand approaches the transmitter and receiver,  $C_1$  increases and  $C_0$  decreases, leading to a drop in received current: the displacement current that had been flowing to the receiver is shunted by the hand to ground (hence, the term shunt mode). The sensed signal is defined to be the magnitude of the decrease in received current as the hand moves in from infinity. In other words, one measures a baseline received current when the hand is at infinity, and then subtracts later readings from this baseline. Thus, when the hand is at infinity, the signal is zero, and as the hand approaches the sensor, the signal increases.

With  $n$  ordinary capacitive sensors (loading mode), one can collect  $n$  numbers. These  $n$  numbers turn out to be the diagonal of the capacitance matrix for the system of electrodes. In shunt mode, one measures the  $n(n-1)$  off-diagonal elements.<sup>2</sup>

As the size of the electrode array grows, it becomes possible to infer more about the geometry of the hand or other body parts—as  $n$  gets large (say, above 10), a kind of fast, cheap, low resolution 3-D imaging becomes possible [5]. However, the large amounts of data are not without a cost. If one were to measure sequentially each value of the capacitance matrix, the time required would be  $O(n^2)$ . Fortunately, it is possible to make all of the receive measurements for a single transmitter simultaneously. This brings the measurement time down to  $O(n)$ . One might expect the measurement time to be constant rather than linear since all of the measurements are being made simultaneously. However, as will be explained in the section on resource scaling, the simultaneous measurements interfere with one another, leading to linear rather than constant scaling. But before the scaling is considered in more detail, the basics of code-division multiplexing will be reviewed.

### III. DIRECT-SEQUENCE SPREAD-SPECTRUM AND CODE-DIVISION MULTIPLEXING

In direct-sequence spread spectrum [6], [7], the signal is modulated with a pseudorandom carrier, usually generated by a maximum length linear feedback shift register (LFSR). With code-division multiple access (CDMA), multiple users share the same physical channel by choosing different coded

waveforms. Another common channel-sharing technique is time-division multiple access, in which transmitters avoid transmitting simultaneously.

The simple implementation of direct sequence spread spectrum for electric field sensing can be understood in terms of linear algebra. Each measurement is made using a single short pseudorandom measurement burst, during which the hand is assumed to be stationary.

If the pseudorandom carrier signal for transmitter  $i$  at time  $t$  is  $\phi_i(t)$ , then the signal received on electrode  $j$ , as modified by the capacitance matrix to be measured, is

$$R_{ij}(t) = C_{ij}\phi_i(t) + N_{ij}(t)$$

where  $C_{ij}$  is a constant because the hand geometry, and therefore the capacitance matrix, are assumed to be static on the time scale of a sensing burst, and  $N_{ij}(t)$  is noise. The noise is indexed by the transmitter and receiver because, for each demodulation operation, the signal from the other transmitters appears as noise. Thus, the noise depends not only on the receiver, but also on which transmitter is being demodulated.

In the ideal case, the transmitted waveforms would be orthogonal to one another, so that channels do not interfere

$$\langle \phi_i, \phi_j \rangle = \frac{1}{s} \sum_{t=1}^s \phi_i(t)\phi_j(t) = G^2\delta_{ij}$$

where  $s$  is the number of chips in a burst and  $G^2$  is the average power per chip. One can view time division multiplexing as a case in which the carriers do not overlap, and thus satisfy this orthogonality condition exactly.

In CDMA systems, there are nonzero cross correlations between different code sequences. This can be a serious problem in a communications scenario in which receivers may spuriously lock onto the wrong code sequence. However, in the sensing application, there is no synchronization problem because the transmitter and receiver are on the same circuit board, so the only problem is a decrease in SNR due to interference between the channels.

It would be best if the basis functions were also orthogonal to the ambient noise  $N$ . In reality, they are not completely orthogonal to  $N$ ; if they were, one could sense using arbitrarily little energy or time. Since the noise is uncorrelated with the carrier, then by the law of large numbers, the fluctuations around zero of the inner product of the carrier and the noise are on the scale of  $1/\sqrt{s}$ , where  $s$  is the number of samples

$$\langle \phi_i, N \rangle = \frac{1}{s} \sum_{t=1}^s \phi_i(t)N(t) \approx 0 \pm \frac{1}{\sqrt{s}}.$$

To measure an entry in the capacitance matrix, form the inner product of the received signal  $R_{ij}$  with the transmitted signal  $\phi_i$

$$D_{ij} = \langle R_{ij}, \phi_i \rangle = \frac{1}{s} \sum_{t=1}^s R_{ij}(t)\phi_i(t) = G^2C_{ij} \pm \frac{1}{\sqrt{s}}. \quad (1)$$

<sup>2</sup>Because the capacitance matrix is symmetrical, there are ideally only  $(1/2)n(n-1)$  distinct values. In practice, apparent deviations from symmetry can be used to calibrate the measurement system.

#### IV. RESOURCE SCALING

With the code-division sensor multiplexing techniques explored in this paper, one might argue that it should be possible in constant time to collect data that completely characterize the  $n^2$  entries in the capacitance matrix. Each transceiver unit would simultaneously transmit its coded carrier, measure the combined signal induced by the other transmitters, and store its measurement samples in memory. The vector of samples in unit  $\hat{i}$  contains a complete, but encoded, representation of all of the  $C_{ij}$  values.

Of course, the “constant time” figure ignores deviations from orthogonality of the coded waveforms. A simple algebraic argument suggests that the number of samples necessary to maintain orthogonality grows linearly with  $n$ . If sinusoids are used instead of pseudorandom sequences, then the length of the measurement vector must be equal to (or greater than) the number of sensor channel amplitudes to be extracted since the DFT matrix is square. Thus, more realistically, one can form a complete but computationally encoded representation of the capacitance matrix using measurement time proportional to  $n$ , and storage of  $n$  values in each of the  $n$  units.

So far, this discussion has ignored the computational operations necessary for unit  $\hat{i}$  to do signal separation, transforming the raw measurement vector into an explicit representation of the  $n$  capacitance values. The naive separation algorithm (for a single receiver unit—multiple receivers can process in parallel) requires time  $mT$ , where  $T$  is the number of samples in the measurement vector and  $m$  is the number of channels being demodulated. By the argument from the previous paragraph,  $T$  cannot be less than  $n$ , and in practice, should be some constant multiple of  $n$ . In the case in which one wishes to demodulate all  $n$  channels (so that  $m = n$ ), then the naive demodulation algorithm requires  $n^2$  time. (It is natural to wonder whether an  $n \log n$  fast spread-spectrum demodulation algorithm exists, analogous to the fast Fourier transform, that could be used with coded rather than sinusoidal carriers. A pseudorandom generator with hidden symmetry properties would probably be needed.) The naive algorithm to extract just a small number  $m$  of the possible  $n$  sensor values would require time  $mn$ .

In the time-division multiplexed (TDMA) case, the transmitted carriers are exactly orthogonal to one another since they do not overlap at all. As in the CDMA case, the total measurement time for one unit is proportional to  $n$ . Because the carriers are exactly orthogonal, the time required for a single measurement is constant, rather than proportional to  $n$  as in the CDMA case. The amount of computation required to demodulate TDMA is also proportional to  $n$ . Thus, in the limit in which all possible measured values are actually demodulated, TDMA appears to have an advantage, requiring  $n$  rather than  $n^2$  or  $n \log n$  time. In the limit in which just a few of the possible sensor values are extracted, the algorithms scale similarly since TDMA still requires linear measurement time, and CDMA’s processing time becomes linear instead of quadratic.

In the practical examples discussed in this paper, there were four transmitters and one receiver (rather than measuring the full  $5 \times 5$  capacitance matrix of this five-electrode system),

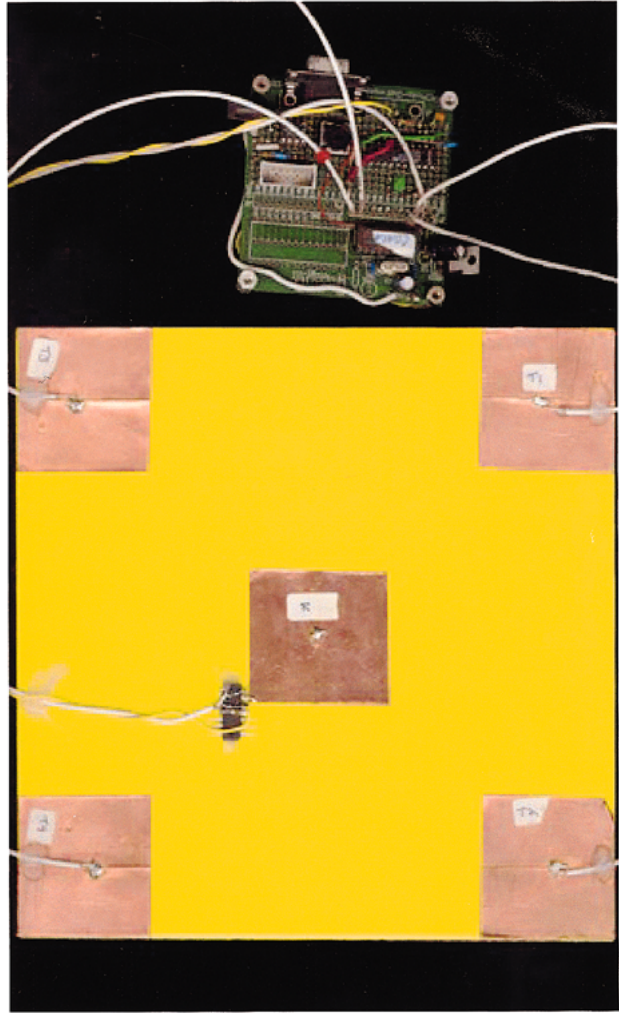


Fig. 2. Hand-wired prototype circuit board used to make measurements, together with electrode array. In the center is a receive electrode, with front-end op-amp attached. In the corners are the transmit electrodes.

so  $m$  was four. The theoretical discussion above considers the measurement and processing time required to achieve the same SNR using software TDMA and CDMA. In the examples, the same total measurement and processing time is allotted to the two schemes, so that a meaningful comparison of the resulting SNR may be made.

The next sections describe the hardware and software used to implement the sensing schemes introduced above.

#### V. HARDWARE

The analog front end consists only of a MAX 474 dual opamp. It is configured as a transimpedance amplifier with a 1M resistor and 22 pF capacitor in its feedback network, followed by an inverting voltage gain stage with a 10K input resistor and 100K feedback resistor, for a gain of 10. Because the opamp uses a single supply (5 V), but the received signals are bipolar (positive and negative), 2.5 V was used as analog ground. The amplified signals are read by the analog-to-digital converter on the PIC16C71 microcontroller. The PIC transmits data through a Maxim MAX233 RS-232 transceiver to a host computer for display and analysis. Fig. 2 shows the

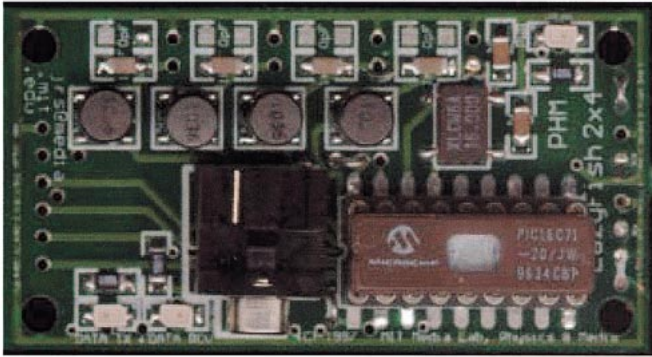


Fig. 3. Later printed circuit board implementation that supports four transmit and two receive channels.

hand-wired prototype board used to make the measurements described in this paper, along with the electrodes. Fig. 3 shows a later printed circuit board implementation that supports four transmit and two receive channels.

## VI. SOFTWARE DEMODULATION

If the gain variable  $G$  from Section III is unity, then the multiplications in the inner product operation described in Section III become additions and subtractions. Performing the demodulation in software requires an accumulation variable in which to store the value of each inner product as it is being calculated. When the first transmitter is high, the processor adds the current ADC value to the first accumulator; if the second transmitter is low, the processor subtracts that same ADC value from the second accumulator. The same ADC value is operated upon differently for each of the demodulation calculations. The addition operation can be thought of as a multiplication by +1 followed by addition to the accumulator; the subtraction is a multiplication by -1 followed by addition to the accumulator. Thus, the operation just described effectively takes the inner product of a vector of samples with a vector representing a transmitted waveform.

An 8-bit LFSR was used, with taps at bits 3, 4, 5, and 7 (counting from zero). This set of taps is known to be maximal [8]. The C code fragment below calculates the four LFSR's, and then sets the states of the transmit pins accordingly.

```
for(k = 0; k < 3; k++) { // k indexes the 4 LFSRs
    low = 0;
    if(lfsr[k]&8) // tap at bit 3
        low++; // each addition performs
                // XOR on low bit of low
    if(lfsr[k]&16) // tap at bit 4 low++;
    if(lfsr[k]&32) // tap at bit 5 low++;
    if(lfsr[k]&128) // tap at bit 7 low++;
    low&= 1; // keep only the low bit
    lfsr[k] <<= 1; // shift register up to make
                  // room for new bit
    lfsr[k] = low; // or new bit in
}
```

```
OUTPUT_BIT(TX0, lfsr[0]&1); // Transmit according
                             // to LFSR states
OUTPUT_BIT(TX1, lfsr[1]&1);
OUTPUT_BIT(TX2, lfsr[2]&1);
OUTPUT_BIT(TX3, lfsr[3]&1);
```

The variable `lfsr` is an array of four bytes. Each byte in the array represents the state of one LFSR. The variable `low` also holds a byte. It is used as temporary storage to calculate the new value of the LFSR's low bit, which must be set to the sum modulo 2 of the current "tap" bits. The value of the least significant bit of `low` contains the appropriate modulo 2 sum after the compare and increment operations above. Its LSB is then masked off and ORed into the low bit of `lfsr`.

The next code fragment performs the demodulation operation. Each value returned by the ADC is operated upon differently for each channel, as specified by the appropriate LFSR.

```
meas = READ_ADC(); // get sample
for(k = 0; k < 3; k++) {
    if(lfsr[k]&1){ // check LFSR state
        next = accumul[k]; // keep backup of low
                          // byte of accum
        accumul[k] += meas; // add measured val to accum
        if(accumul[k] < next) accumul[k]++;
        // if overflow, then carry
    }
    else {
        next = accumul[k];
        accumul[k] -= meas;
        // subtract measured val from accum
        if(accumul[k] > next) accumul[k]--;
        // carry if necessary
    }
}
```

On the host computer, the final sensor value is found by dividing the accumulated value (`accumh*256 + accumul`) by  $s$ , the number of samples taken, as in (1). Because of the division by  $s$ , the fluctuations diminish as  $1/\sqrt{s}$ , but the typical magnitude of the sensed value (that is, the mean of the random variable describing the final sensed value) is independent of  $s$ .

## VII. RESULTS: COMPARISON WITH TIME DIVISION

The performance of the CDMA and TDMA techniques was compared by measuring four channels in 105 ms. The ADC on the current PIC16C71 can sample at about 40 kHz. Calculating the four LFSR's, performing the ADC, and doing the four demodulation computations, the PIC's chip time was 136  $\mu$ s or 7.4 kchip/s. To examine the limit in which the processing time is negligible compared to the ADC time, one compares the performance of the CDMA algorithm with a TDMA algorithm generating a 3.7 kHz square wave (7.4 kchip/s). The total

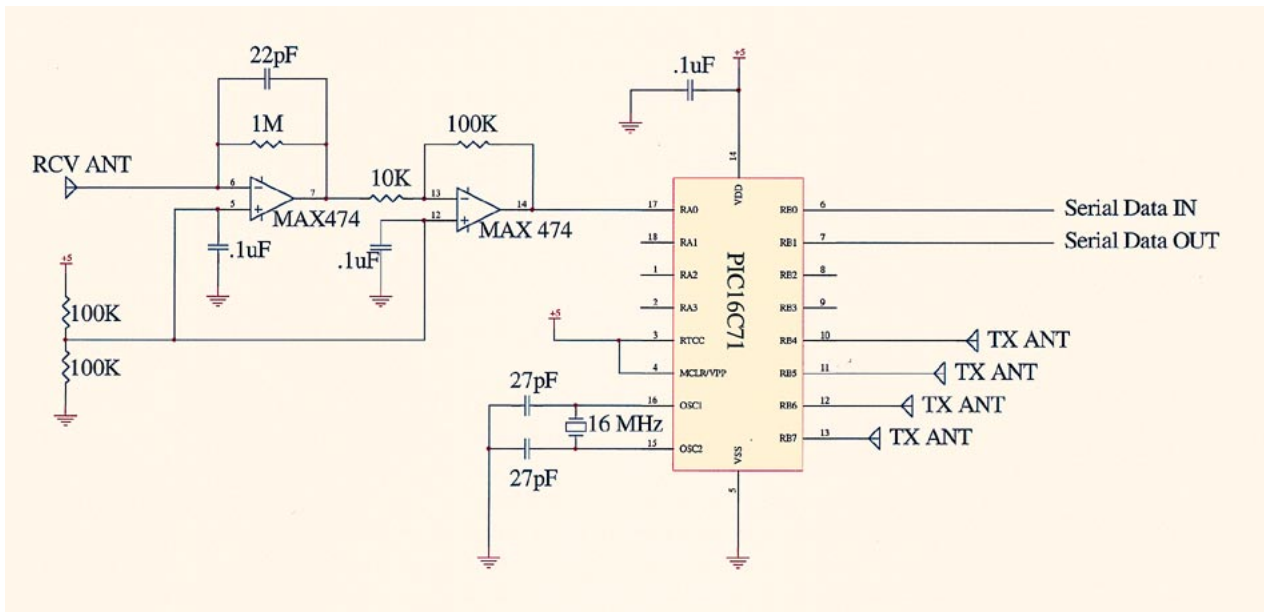


Fig. 4. Analog front end consists of two opamp gain stages. These feed into the ADC built into the PIC. Demodulated values are sent serially to a host computer for display analysis.



Fig. 5. Top trace: received signal induced by four coded waveforms. Middle and bottom traces: two of the four transmit channels.

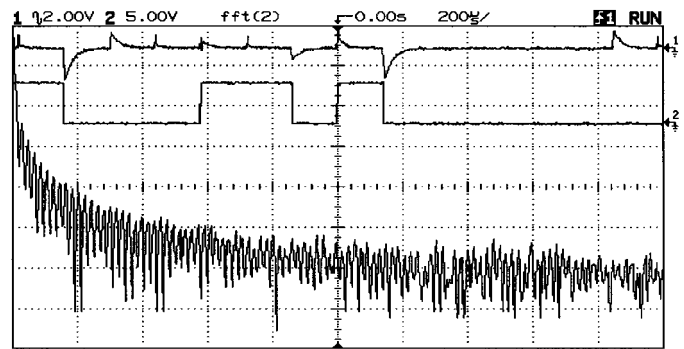


Fig. 7. Top trace: signal received from four coded waveforms. Middle trace: one of the transmit channels. Bottom trace: Fourier transform of one transmit channel (frequency span: 244.1 kHz; 10 dB/division). Notice how much more uniformly the LFSR-generated carrier fills the bandwidth than the square wave of Fig. 6.

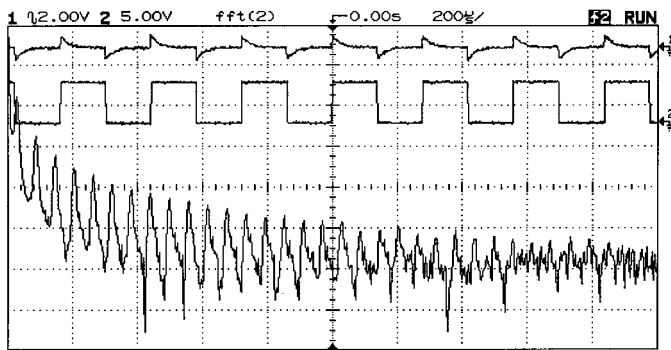


Fig. 6. Top trace: received signal from one transmitted square wave. Middle trace: transmitted square wave. Bottom trace: Fourier transform of transmitted square wave (frequency span: 244.1 kHz; 10 dB/division).

measurement time for the four time multiplexed channels was fixed at 105 ms. An additional comparison was made in which the PIC generated a square wave at top speed (given the time required for analog-to-digital conversion and demodulation calculations), which was 13.5 kHz.

To measure the SNR on a particular channel requires calibration of the maximum and minimum observed values as a hand moved from outside the sensor’s active region (defined to be zero signal) to the sensor’s most sensitive point (maximum signal). With the hand outside the sensitive region, 100 samples were used to calculate the standard deviation. The SNR estimate is an average of five ratios of maximum signal to standard deviation.

When limited to the same chip rate, the SNR for the CDMA and TDMA approaches were as follows: for CDMA, the SNR was 340, or 51 dB, or 8.4 bits; for TDMA, the SNR was 320, or 50 dB, or 8.3 bits. When the TDMA system was operated at the maximum data rate, the SNR increased to 55 dB. This is not surprising. The total output signal power is higher in the higher chip-rate case, and so the SNR should be higher. Thus, in the practical systems of today, the higher computational cost of generating the pseudorandom sequence (which limits the chip rate) diminishes the performance. When the computation

time becomes negligible compared to the ADC time, there is little difference between the performance of time-division and code-division multiplexing.

### VIII. CONCLUSION

There are clear advantages for sensing applications of a software-multiplexed analog front end over hardware multiplexing or multiple analog channels. There are minimal requirements for special-purpose hardware and increased flexibility. Channels may be added without hardware modifications.

Comparing the two software channel-sharing schemes, there are no significant performance differences between CDMA and TDMA. There is a slight practical advantage at present to TDMA. The advantage is that TDMA does not require the computation time to calculate the LFSR's. But even with present technology, this difference could be eliminated by a more efficient algorithm.

Although the mean SNR values were the same for CDMA and TDMA, the variance of the SNR's for TDMA was higher. In the TDMA scheme, each single-channel measurement is based on a smaller number of samples. Thus, even though the average channel properties are the same, there are larger fluctuations in the noise levels in the TDMA channel. In CDMA, the several transmit channels appear as noise to one another, but in this sensing application, with its fixed electrode geometry, the level of that self-induced noise is consistent. Since each TDMA sensor value is formed from a smaller number of samples, the TDMA channels are more "bursty." Although the mean SNR is the same for the two modulation techniques, the distribution of SNR measurements is scattered more broadly about the mean for the TDMA technique than for the CDMA technique. For bulk communications purposes, the increased SNR variance does not matter, but for streamed data or sensor values, this is a distinct disadvantage. The performance of a sensor system is only as good as the worst case SNR, measured in short windows of time. During a bulk communications operation like an FTP download, the short time performance of the channel is irrelevant; the relevant figure of merit is the global performance, calculated over the entire time required to complete the operation.

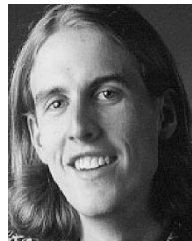
Apart from the subtle SNR variance advantage over TDMA, it is difficult to determine whether the properties of CDMA sensing channels make them significantly more attractive than TDMA channels. The additional flexibility—such as the possibility of determining which sensor channels, or how many, to extract after the raw measurements have been made—might be useful in certain applications. For making multiple measurements with an array of sensors, the more consistent noise levels may make CDMA more attractive than TDMA. In either case, it is useful to process multiple channels of sensor data using a single analog front end, ADC, and DSP software. And while it is common practice to send multiple streams of communications data simultaneously through a single physical channel, it is historically less common (although fundamentally no different) to use a single analog front end and ADC to simultaneously process multiple-sensor channels.

Despite the appeal of CDMA sensing, the most practical implementation of electric field sensing to date, the "Lazy-Fish" board shown in Fig. 3, uses  $LC$  resonant circuits to transform the four 5-V square-wave microcontroller outputs into 80-V sinusoidal carriers. These four transmit channels are time-division multiplexed into two analog front ends. The resonators, which provide a significant increase in SNR, are useless with a spread-spectrum excitation. For the practical system, the SNR advantage of resonant transmission tipped the balance decisively in favor of TDMA.

The practical system benefits from one additional application of software radio techniques. The LazyFish performs quadrature demodulation in software, which enables it to be invariant to phase shifts, such as those caused by cable capacitance. In a hardware implementation, quadrature demodulation would require almost doubling the amount of analog hardware—only the front-end amplification could be shared by the in-phase and quadrature channel associated with a particular receiver. Each receiver would require an additional analog multiplier and low-pass filter for the additional quadrature channel. The LazyFish board fits 16 sensing channels (counting quadrature channels) into a 1 in  $\times$  2 in footprint. Its small size would not have been possible without shifting the division of labor from hardware to software.

### REFERENCES

- [1] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, "Applying electric field sensing to human-computer interfaces," in *CHI'95 Human Factors in Comput. Syst.*, Denver, CO, 1995, pp. 280–287, ACM Press.
- [2] J. R. Smith, "Field mice: Extracting hand geometry from electric field measurements," *IBM Syst. J.*, vol. 35, no. 3 and 4, 1996.
- [3] J. A. Paradiso and N. Gershenfeld, "Musical applications of electric field sensing," *Comput. Music J.*, 1997.
- [4] NEC Automotive Electronics, *Smart Airbags and the Future of Automotive Sensing Technologies from NEC*, World Wide Web page, <http://www.nectech.com/auto> 1997.
- [5] J. R. Smith, "Electric field imaging," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, Feb. 1999.
- [6] R. C. Dixon, *Spread Spectrum Systems with Commercial Applications*. New York: Wiley, 1994.
- [7] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *The Spread Spectrum Communications Handbook*. New York: McGraw-Hill, 1994.
- [8] P. Horowitz and W. Hill, *The Art of Electronics*. Cambridge, England: Cambridge Univ. Press, 1990.



**Joshua R. Smith** received double B.A. degrees in computer science and philosophy from Williams College, the M.S. degree from the M.I.T. Media Lab, the M.A. degree in physics from Cambridge University and the Ph.D. degree from the M.I.T. Media Lab.

He is Director of Escher Laboratories, Cambridge, MA. He holds, along with N. Gershenfeld, several patents on electric field sensing. While at M.I.T., he co-invented the Pixel Tag family of digital watermarking algorithms. He has had visiting positions at the Santa Fe Institute, Los Alamos National Lab, Yale University Department of Computer Science, and NASA's Goddard Institute for Space Studies.



**Christopher Salthouse** is an undergraduate majoring in electrical engineering and computer science at M.I.T. He worked in the Physics and Media Group as an Undergraduate Research Opportunity Student.



**Neil Gershenfeld** (S'80–M'81) received the B.A. degree in physics (with high honors) from Swarthmore College and the Ph.D. degree from Cornell University, studying order–disorder transitions in condensed matter systems.

He is an Associate Professor of Media Arts and Sciences. He leads the Physics and Media Group at the M.I.T. Media Lab, and codirects the Things That Think industrial research consortium with sponsors including HP, Motorola, Microsoft, Fedex, Nike, and Disney. His laboratory investigates the interface between the content of information and its physical representation, from building molecular quantum computers to building musical instruments for collaborations ranging from Yo-Yo Ma to Penn and Teller. Research in his lab has found application in products in markets such as computers, automobiles, and toys. He was a Technician at Bell Labs using lasers for atomic and nuclear physics experiments, and he was a junior fellow of the Harvard Society of Fellows, where he organized an international study hosted by the Santa Fe Institute on forecasting the behavior of complex systems.