Acoustic Modems for Ubiquitous Computing

Sound offers features not available with other short-range, low bandwidth communication technologies, such as radio and infrared, enabling communication among small computing devices and humans in a ubiquitous computing environment.

Cristina Videira Lopes
University of California, Irvine

Pedro M.Q. Aguiar
Technical University of Lisbon

Sound has been largely ignored or dismissed in digital communications. Given that computer systems have been traditionally aimed at long-distance and high-bit-rate communications, this dismissal is understandable. The data rates for sound are relatively low compared to media such as radio, and the sounds can be annoying. Designers of most current digital communication systems assume applications need comminicating devices to exchange large amounts of information, and so they focus on maximizing the transmission bit rate. This is the case, for example, with transferring files over a local area network (LAN) or video over the Internet. As small, mobile computational devices become more pervasive, however, the need for short-range, localized communications is as urgent as the need for long-distance communications.

Ubiquitous computing[1] refines the very notion of computers and computation, envisioning a world in which small devices with embedded computational and communication capabilities interact with other devices and with humans for purposes other than traditional desktop work. Efforts such as IrDA, Bluetooth, and, more recently, impulse radio, or ultrawideband signaling[2,3] target the short-range communications critical in ubicomp. Ubicomp also breaks the assumption that computing applications need high bandwidth. In fact, interconnected devices' communication needs are diverse, ranging from very low to very high bit rates, and from short-range, localized messaging to long-range connections.

Motivated by the specific characteristics of the aerial acoustic communication paradigm used by humans and other animals, we examine the use of device-to-device aerial acoustic communications in ubicomp applications. Unlike radio, sound is contained within walls, which can help solve interference problems at the physical layer. And, unlike infrared, sound doesn't need line of sight and so can go around corners, which can be useful for moving or hidden devices. Unlike both media, sound can also expose the communication to humans when necessary, as well as some location aspects of the communication. Another reason to revisit acoustic modems 40 years after their invention is that CPUs are several orders of magnitude faster than they were in the 1960s. Thus, we can now implement, entirely in software, cheap acoustic modems that use ordinary microphones and speakers.

The Digital Voices project explores device communications using audible sound.[4] (The "Sound in Device Communications" sidebar gives an overview of the use of sound in device communications.) Within the project, we investigate bridges between human and computer commu-

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/03/$17.00

Sound in Device Communications

Ultrasonic remote controls, invented in the 1950s, were one of the earliest attempts to use sound in machine communications and are still used in commercial products. The first commercial ultrasonic remote control, Zenith's Space Command, was a purely mechanical device that used four rods—two for channel up and down, one for power on or off, and one for sound on or off—instead of batteries. Designers cut the rods to lengths that would generate sounds of slightly different frequencies.

We categorize the traditional uses of sound in computer systems into three groups:

- Long-distance point-to-point computer communications over existing telephone networks (phone-line modems)
- Underwater communications
- Speech recognition/synthesis and other nonspeech auditory displays that enhance human-machine interactions

Telephone modems

Modems originally allowed long-distance point-to-point communication using the voice band in ordinary telephone networks. The first modems were acoustically coupled: a user placed the telephone receiver into a modem handset and the modem sent tones to the telephone. Early acoustic modems transmitted at 300 bits per second.

Direct-connect modems, which interface directly with the telephone line, have replaced these acoustic modems. They are less bulky, give a better connection, and avoid the background noise problems of acoustic modems. Modern modems transmit at bit rates of up to 56 Kbps.

Underwater communications

Banned from above-ground computer networks, acoustic modems found a niche in underwater communications. Although water is a poor medium for radio propagation, it's very good for sound propagation. Since World War II, submarine crews have used ultrasound to detect objects, position, and communicate. Recent advances in digital signal processing hardware, driven by the needs of autonomous underwater vehicles, have made acoustic modems for underwater wireless communications possible. These modems use the upper audible and lower ultrasound bands (12 KHz to 30 KHz) and transmit at a maximum bit rate of 19.2 Kbps. A survey of underwater communications is available elsewhere.[1]

Human-computer interaction

Human-computer communications use sound in speech recognition and auditory displays.[2] Speech recognition and synthesis is a well-established field of research and development; we focus on the less examined nonspeech interfaces.

Auditory displays help users monitor and comprehend the data the sounds represent. Any data source—from databases to real-time stock market quotes—can provide information to display. Many systems have used nonspeech sound for auditory displays, either as stand-alone interfaces or graphical user interface enhancements.

Auditory display of computer communications takes many forms. Anyone with root access to a Unix machine, for example, can feed the Ethernet card traffic into the loudspeakers, generating a series of noises that reflect the electric impulses of messages arriving. Several digital artists have created artwork reflecting the sounds of network traffic.

The modems we describe in this article aren't displays for humans to hear; rather, they are displays for computers to read, but they can convey information to human listeners.

Recent work

In recent years, researchers have examined the area of information hiding in audio (a survey by Stefan Katzenbeisser and Fabien Petitcolas, for example, contains extensive references to work in the field[3]). The music industry has been exploring audio watermarking as a way to preserve ownership of music in electronic format. Interest in this kind of communication is growing, and much work remains.

Vadim Gerasimov and Walter Bender used the aerial acoustic channel for device-to-device communications.[4] They evaluated simple modulation techniques according to the data rate, computational overhead, noise tolerance, and disruption level, and report a maximum data rate of 3.4 Kbps using a frequency of 18 KHz.[4]

REFERENCES

1. M. Stojanovic, "Recent Advances in Underwater Acoustic Communications," *IEEE J. Oceanic Eng.*, vol. 21, no. 2, 1996, pp. 125–136.
2. G. Kramer, ed., *Auditory Display*, vol. XVIII of *SFI Studies in the Sciences of Complexity*, Addison-Wesley, 1994.
3. S. Katzenbeisser and F. Petitcolas, eds., *Information Hiding: Techniques for Steganography and Digital Watermarking*, Artech House Publishers, 2000.
4. V. Gerasimov and W. Bender, "Things that Talk: Using Sound for Device-to-Device and Device-to-Human Communication," *IBM Systems J.*, vol. 39, nos. 3–4, 2000, pp. 530–546.

nications and explore the existence of a pervasive infrastructure for sound in the audible band, including desktop computers, palm devices, memo recorders, and televisions, to support no-cost short-range communications in ubicomp applications. Hence our focus on audible sound: We could also use ultrasound, but it requires hardware that few existing audio devices have.

Digital Voices overview

When we first started exploring sound, we analyzed common modulation techniques and the sounds they produce. Some modulation parameters

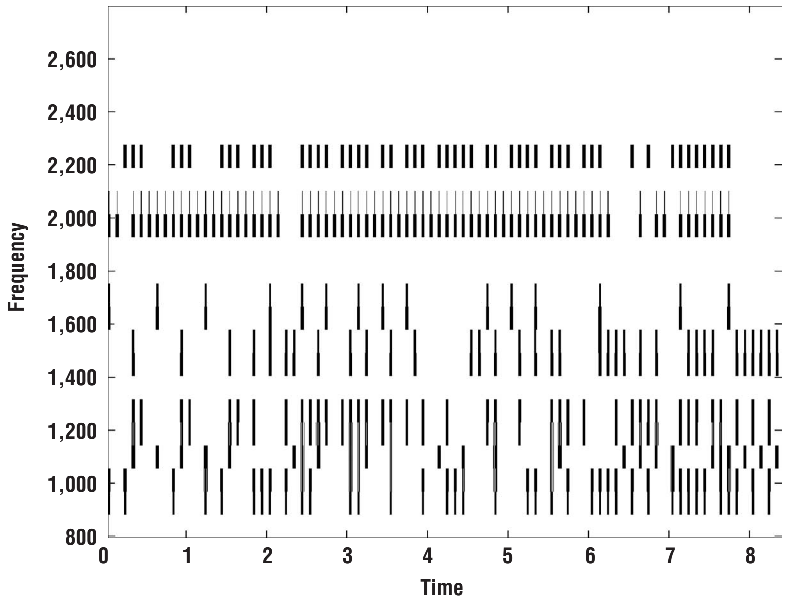

Figure 1. Spectrogram of the sound of a multifrequency B-ASK (binary-amplitude shift key) modulated message with a 100-ms symbol duration. Carrier frequencies are those of the musical notes of the pentatonic scale. The message is a sequence of 7-bit ASCII characters in which the higher-order bit is always 0. This sound resembles music played by soprano flutes.

strongly affect how humans perceive sound quality. Varying those parameters lets us obtain many different types of acoustic messages, ranging from modem noise to music.

In this article, we describe our experience designing audible-range acoustic modems for ubicomp using two criteria:

- The messages of these communication systems must be pleasant to people. They should sound like music or other familiar sounds.
- The systems must be deployed in ordinary hardware and use the existing infrastructure for voice to avoid extra cost. Software should perform the coding and decoding.

Our experiments show that our goals are attainable. This article reports those experiments and lays out the ground work for future research.

Application areas

Ubicomp applications can and should exploit all sorts of short-range communication media, including radio, infrared, and sound. Because these media complement each other, they should be appropriate for each application.

Applications that don't require high bit rates can use sound. A developer might choose sound instead of radio or infrared for an application that

- Can easily be deployed in the existing voice infrastructure, avoiding the expense of extending the hardware with radio or infrared transmitters
- Requires or benefits from human awareness of the communication
- Requires short-range broadcasts enclosed within walls and windows

One application of Digital Voices, developed at Xerox PARC, uses sound to safely transmit preauthentication information in wireless networks.[5] To preauthenticate information using the wireless (radio) network in an office building, a laptop user first goes to the room where the base station is located. Once there, the laptop and base station exchange cryptographic keys via microphones and speakers, with all exchanges remaining secure within the room. When the laptop has the key, as well as other context-dependent information, the user can roam the building using crypto over the "leaky" wireless network.

Another application is related to steganography. We've implemented communication systems that produce acoustic messages using natural sounds, such as bird songs or crickets. To the human ear, these sounds are indistinguishable from those produced by animals, yet they carry coded messages. Agencies conducting secret operations might be interested in such an application, given that these sounds can be broadcast by traditional media and can be picked up anywhere in the world with a laptop or PDA.

A related application area is entertainment. One of our communication systems sounds like R2D2, the robot from the *Star Wars* movies. For demonstration purposes, we edited parts of the movies, replacing the original R2D2 sounds with our sounds containing messages. An R2D2 toy augmented with a microprocessor and a microphone can receive instructions or programs from the movies or from TV. Using sound as the communication medium is not only fun but also easy to implement without having to replace or augment the TV.

Ad hoc networks that relate back to the Internet is a fourth candidate area for Digital Voices. Memo-recording devices (such as most cell phones) could store and exchange URLs in a form that's more reliable and compressed than human speech. For example, a radio program could broadcast the URL for a special discount during a commercial. Listeners could record it in their cell phones and later play it back to the desktop computer.

Human factors

Because this work targets short-range,

low-bit-rate aerial communications that might interest people, why don't we use human speech? In short, human speech is not always the most efficient encoding, and speech recognition engenders a huge computational overhead in the decoding devices. For example, the text www.amazon.com/exec/obidos/ASIN/0262640414/qid=1034295226/sr=2-1/ref=sr_2_1/002-2920555-7785632 is human-readable but not meant for humans to read, much less pronounce. At most, humans might be interested in knowing the Web site to which the URL refers. These sort of short messages are central to several ubicomp applications. Standard techniques in digital communications engineering are simpler and more efficient than human speech.

Early in our experiments, we found that the straightforward application of standard modulation techniques used in nonaudible modems is inappropriate for audible modems, because people associate those sounds with machine communications and tend to get annoyed by them. When designing aerial acoustic communication systems, we must account for human perception and make sounds that are acceptable to people when they need to be aware of them, and imperceptible when they do not.

Human factors affect sound design in two types of acoustic messaging systems:

- Systems using very short duration sounds that are seldom transmitted (say, occasional blips under 1 second)
- Systems using longer duration sounds

In the first case, human perception factors in sound design are important but not crucial; however, in the second case, human perception is critical, both as a limitation and an opportunity for creative artists. Although the Digital Voices project uses both types of sounds, this article focuses on modems for long-duration sounds.

Musically oriented variations of common modulation techniques

We adapt common modulation techniques, such as amplitude shift key (ASK) and frequency shift key (FSK), to support our aerial acoustic communication systems. Samples of the sounds our modems produce are available at www.ics.uci.edu/~lopes/dv/dv.html.

Amplitude shift key

In ASK, information is coded in the amplitude of a sinusoidal carrier. To improve robustness, we use binary (B-ASK) signaling—that is, the signal coding a bit is either zero or a sinusoid segment with a specified fixed-value amplitude. To increase the sound's frequency diversity, we use *multifrequency coding schemes*—that is, the coder transmits several bits in the same time slot.

Pentatonic scale. We use an 8-frequency coding scheme, starting at 1,000 Hz. Frequency selection determines the sound's perceptual quality. Although using major or minor *chords*, or sets of tones, worked well, selecting frequencies according to a pentatonic scale[6] achieved a much more pleasant sound. Symbol duration impacts the sound produced and, obviously, the transmission bit rate. Figure 1 shows the spectrogram of sound of the messages for a 100-ms symbol duration. The sound resembles a piece of music played by several soprano flutes, and the data rate is 80 bps. For a 20-ms symbol duration, the sound is similar to that of grasshoppers, and the data rate is 400 bps.

Harmonic ASK. A second sound design uses 128 frequencies, all 70-Hz harmonics, starting at 700 Hz. For a 100-ms symbol duration, the resulting sound is like an electronic-music drum beat. The large number of carriers results in a higher bit rate—1,280 bps—than the previous example.

Frequency shift key

In FSK, the modem fixes the sinusoidal carrier's amplitude and codes the message in its frequency. The number of bits per symbol determines the number of distinct frequencies allowed for the carrier. Our designs use FSK either alone or with ASK.

Harmonic FSK. To transmit 8-bit symbols, we use 256 20-Hz harmonic frequencies starting at 1,000 Hz. For a 20-ms symbol duration, the sound resembles a single grasshopper with a bit rate of 400 bits per second. When we increase the symbol duration to 100 ms the sound is like a bird, and the data rate drops to 80 bps. Figure 2 shows the resulting spectrogram.

Although the data rates for modems using pentatonic scale and harmonic FSK are the same (because both code 8 bits per symbol), the resulting sounds are quite different, as Figures 1 and 2 indicate. In pentatonic scale, several tones usually occur at each time slot, whereas in harmonic FSK, only one tone sounds at each time slot. The human ear cannot distinguish differences in tone for small symbol duration values, such as 20 ms. Differences only become clear at 100 ms.

Human speech is not always the most efficient encoding, and speech recognition engenders a huge computational overhead in the decoding devices.

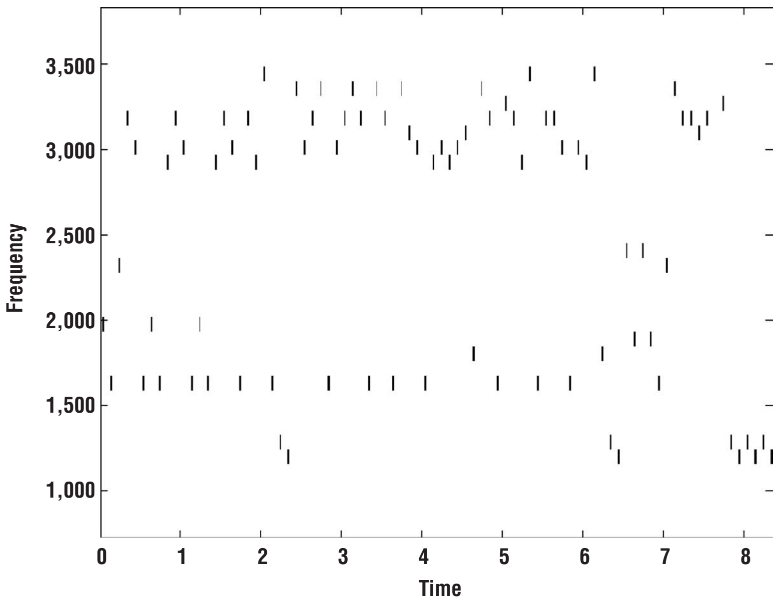

Figure 2. Spectrogram of the sound coding an FSK-modulated 8-bit text string. The sinusoidal carrier frequencies are 20-Hz harmonics and the symbol duration is 100 ms. This acoustic message sounds like a bird.

Chords. In this example, we code 7-bit symbols. We use the seven tones of the A major diatonic scale as the keys to form chords (A is 440 Hz). The chords can be either major or minor (indicated by the presence of the third major or third minor) and can include the seventh tone.

Given a 7-bit value $b_6b_5b_4b_3b_2b_1b_0$, we establish a mapping that simultaneously uses FSK and ASK:

- b_6 determines whether the chord includes the seventh tone
- b_5 determines the mode (major or minor)
- $b_4b_3b_2$ determine the chord's key
- b_1b_0 determine which inversion to use (we assume four possible inversions, the fourth being the same as the first but one octave above)

For 7-bit ASCII characters, this mapping produces sounds that are identifiable by humans as corresponding to numbers versus letters (presence or absence of the seventh tone) and capital (minor) versus lowercase (major) letters. To enhance the sound, we include a silence every seven symbols. The result is a sequence of familiar chords with a 4/4 rhythm that, although in arbitrary sequence, makes the message sound like an ordinary musical composition. With this coding scheme, we get a 35-bps bit rate (for 7-bit ASCII characters, this means 5 characters per second) for a 200-ms symbol duration.

Robustness through redundancy. To increase the robustness of our chord-coding scheme, we use a redundant code. The modem sends the redundant information on the chord key's fourth and sixth harmonics, encoding the same information but in a different form. It encodes bits $b_4b_3b_2$ (the key) in the harmonics' frequency values—for example, if C is the key, the presence of C″ (the fourth harmonic) and G″ (the sixth harmonic) will verify that. The modem encodes the inversion, the mode, and the presence or absence of the seventh harmonic (the remaining four bits) in the time slot at which the two harmonics are played—phase modulation. The result is the same musical composition overlaid with a xylophone-like melodic line that sounds slightly out of tempo.

Frequency hopping

Frequency hopping changes a channel's carrier frequency as a function of time. Researchers have used this mechanism extensively since WWII, both to send messages that enemy receivers cannot detect (unless they know the hopping sequence) and to minimize collisions in multi-user radio channels (such as in cell phone networks). In aerial acoustic communications, frequency hopping is a key factor in producing melodic messages. We've implemented two frequency hopping designs.

Melody. This design's hopping code is a melodic line from the movie, *Close Encounters of the Third Kind*: B′–C#′–A′–A–E–E. The tones last for about one second—that is, the system hops every second—and are included in the final signal. The data is B-FSK-modulated using the hopping code frequency's fourth and eighth harmonics, representing 0 and 1, respectively. The symbol duration is 10 ms, and the bit rate is 100 ms. The result is a relatively slow melodic line with fast temporal variations in the two higher harmonics of each tone.

Bach's piece. The second design uses Johann Sebastian Bach's *Badinerie,* shown in Figure 3, as its hopping code. We use B-ASK on the base notes' first four harmonics. The result is the well-known melody line with fast temporal variations. The duration of each symbol is 32 ms, resulting in a bit rate of 125 bps.

Synthesized musical instruments

Musical instrument synthesis is an active research area in the computer music field, and methods for instrument synthesis have been studied and used for decades.[7] Our approach modulates information in piano, bell, and clarinet sounds. To do this, we synthesize the instruments' sounds and then map the information to musical notes.

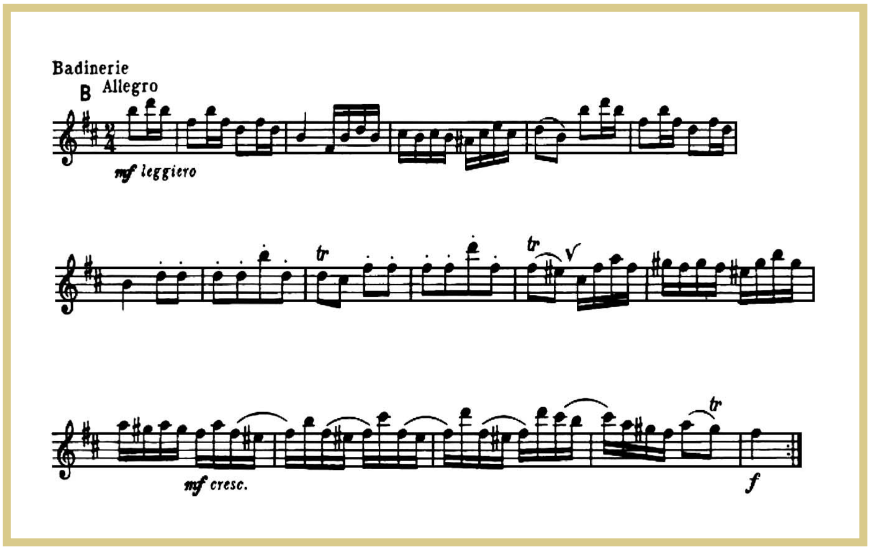

Figure 3. Excerpt from Johann Sebastian Bach's Orchestral Suite No. 2 in B minor (Badinerie), which we used as frequency hopping code.

The challenge of designing air modems is reproducing the instruments' main sound characteristics so the synthesized acoustic waveforms can code the information without requiring specific and complex coding schemes and, consequently, very expensive receivers. As earlier work describes,[8] to achieve this goal we synthesized the musical instruments' audio spectra by either explicitly choosing the sound's spectral components (in the case of the piano) or using frequency modulation techniques (in the case of the bells and clarinet).[9,10]

The modulation scheme assigns each symbol to a synthesized musical note of a given fundamental frequency. The scheme chooses the notes so their spectra do not collide. This way the receiver can perceive the transmitted notes by simply detecting their fundamental frequencies, just like a standard sinusoidal receiver. In addition to its computational simplicity, the receiver doesn't even need to know which instrument the transmitter is synthesizing; the instrument being synthesized can change any time without affecting the transmission. The drawback of this versatility is the low spectral efficiency and consequent very low bit rate.

Bio-inspired modulation techniques

Several sound designs target the emulation of communication systems used by imaginary and real creatures. Examples of these sounds are at http://birds.cornell.edu/brp.

Imaginary creatures

We made R2D2, the robot from the *Star Wars* movies, talk by mapping characters to R2D2-like sounds so the sounds would convey text messages. To achieve this, we first made a time-frequency analysis of R2D2's sounds. Using that analysis, we identified three major sound types: beeps, chirps, and grunts. The R2D2ness of the sounds consists of a mixture of the three sound types, with beeps slightly prevalent. To preserve that nature, we defined the mapping from text symbols to acoustic signals according to Table 1.

With this symbol mapping, our R2D2 modem transmits at an average bit rate of 35 bps. Obviously, the actual bit rate depends on the text message's structure (percentage of punctuation characters and numbers). Figure 4 illustrates the R2D2 modulator by showing the spectrogram of a certain text message.

Animals

We also used sounds from nature, such as bird and cricket sounds. As with the R2D2 modem, we first made a time-frequency analysis of the sounds. We then used the analysis to synthesize sounds resembling the original ones, modulating information in amplitude, frequency, and phase.

Bird. We analyzed a real bird's song obtained from a bioacoustic database by looking for different sound patterns. We synthesized the patterns, embedding information in their amplitude and frequency.

Figure 5 illustrates the process. Figure 5a displays the time evolution of the amplitude of the bird song's acoustic signal. Figure 5b shows three typical segments of that song and the corresponding spectrograms; and Figure 5c shows a synthesized signal emulating the bird song. We modulated the information, represented by a sequence of ASCII characters, in the amplitude and frequency of the song's acoustic units. This coding and modulation scheme yields a 35-bps transmission bit rate.

Cricket. Cricket songs have a particular characteristic structure consisting of sequences of three fast sound impulses, as Figure 6a shows. To produce sounds with this structure, our cricket modulator codes the information, represented by ASCII characters, in amplitude and

TABLE 1
R2D2 symbol mapping scheme.

Symbol	Acoustic signal type	Duration (sec)
A–Z	FSK—that is, beeps of 26 different frequencies	0.1
Space, ?, !,.	Chirps starting and ending at different frequencies	0.25
0–9	Multi-ary (M-ary) ASK using relatively low frequencies	0.2

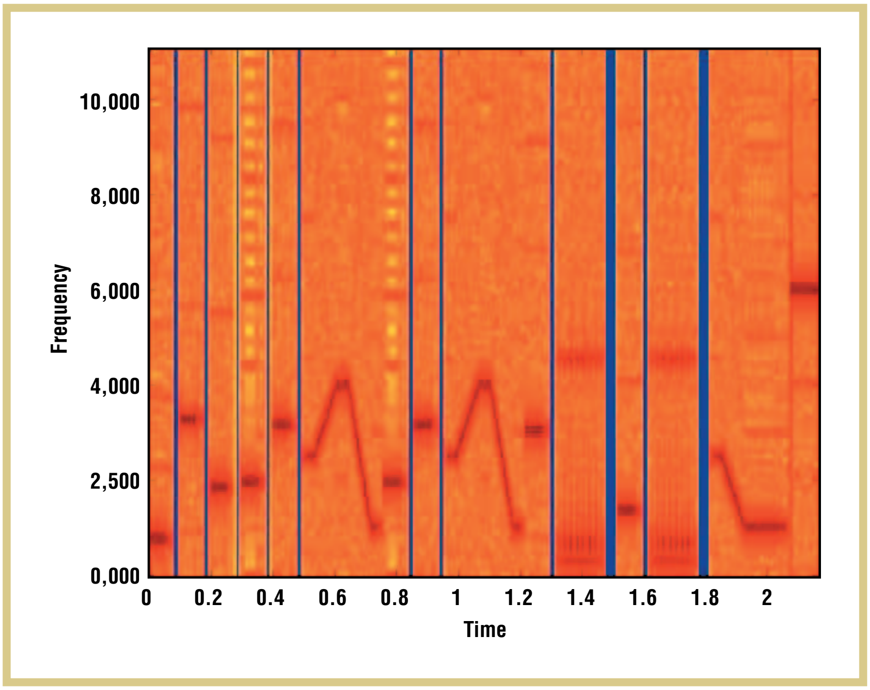

Figure 4. Spectrogram of the sound coding the message, "This is R2D2!" The first-time bin is a hail frequency. Subsequent bins stand for T-H-I-S-space-I-S-.... Notice in particular the longer bins corresponding to the chirps that code the spaces and the ! and the low frequency sounds coding the 2s.

phase. Figure 6b represents the time evolution of the acoustic signal corresponding to a modulated sound. The figure clearly shows the amplitude modulation—each impulse has one of two distinct magnitudes.

In phase modulation, the impulse triples occur once in a specified time period, in this case 0.5 seconds. The modulator divides the time period into four equal slots, letting the information symbols determine in which of the four slots to transmit the impulse triples. This type of phase modulation randomizes the sound, making it more natural. Although subtler, the phase modulation effect is still visible in Figure 6b. This modem encodes information at a bit rate of 22 bps.

Practical matters

Our experiments show that constructing aerial acoustic modems in software and using the existing hardware infrastructure for sound is feasible. In fact, our modems produce sounds that are relatively pleasing to humans over short time periods. We have informally demonstrated the modems to more than 250 people, ranging from audiences of 20 to 50 people to one-on-one demonstrations. In none of these informal demonstrations did anyone associate the sounds with computer communications,

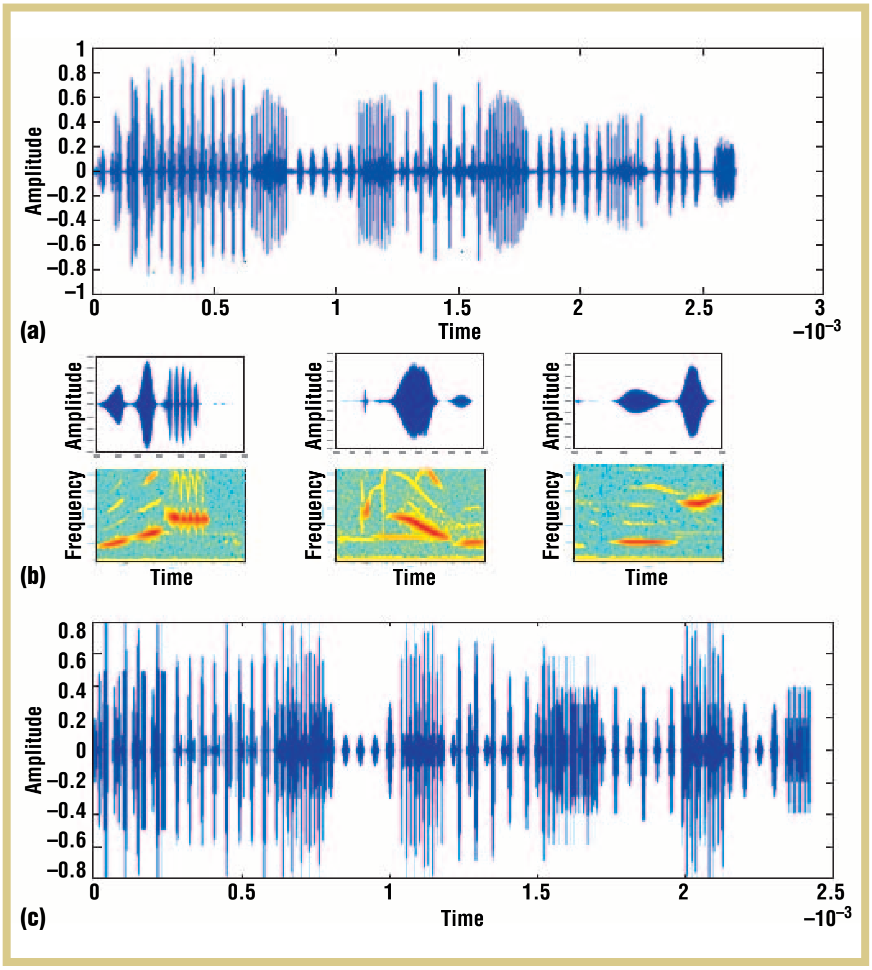

Figure 5. Graphic representations of a bird song: (a) time evolution of the bird's acoustic signal; (b) magnified representative segments with corresponding spectrograms; and (c) time evolution of the synthesized bird sound, resembling the real bird's song.

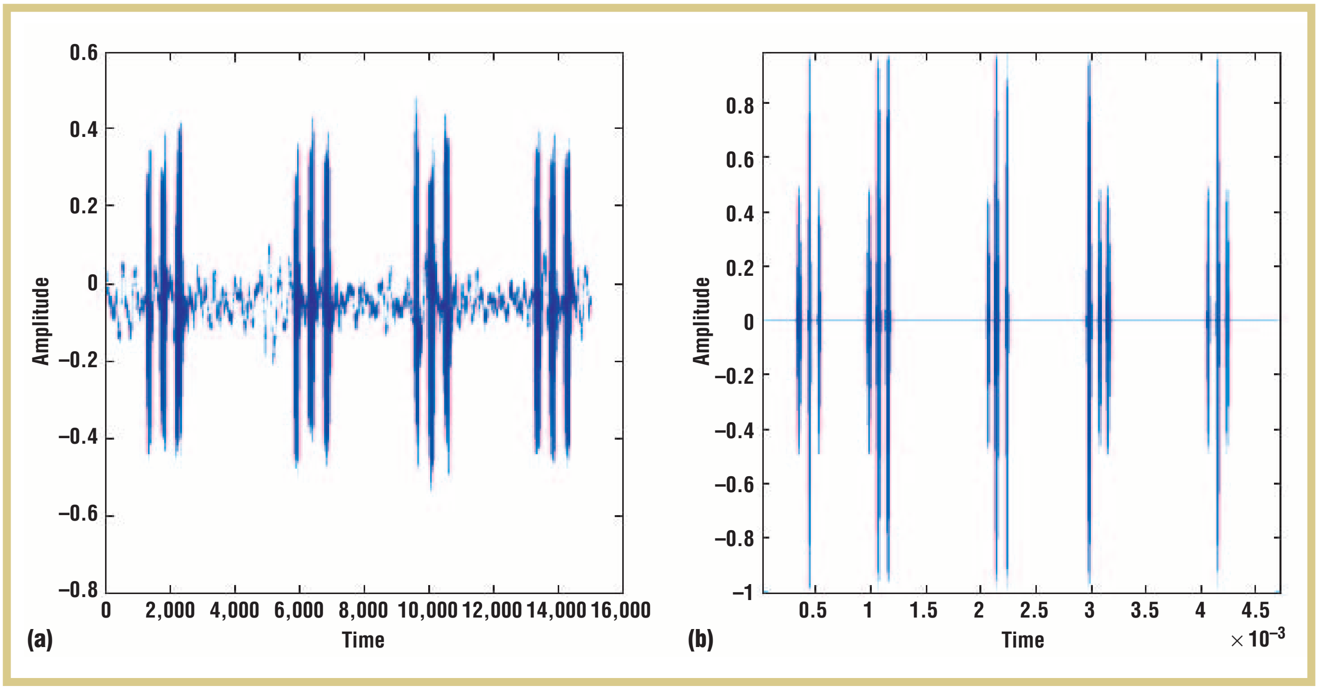

Figure 6. Graphic representations of a cricket's song: (a) acoustic signal of the song, showing sequences of three fast sound impulses, and (b) a cricket modulator.

and most people were surprised when we decoded the sounds into messages. In the one-on-one demonstrations, 90 percent of the people had trouble believing the decodings, and some looked for a hidden trick or infrared and radio links. Some sound designs "fool" human listeners better than others. For example, the bird and the cricket are some of the least suspicious sounds because to the human ear, they sound exactly like their natural counterparts.

Receivers

Earlier work describes the receiver implementation.[4,8] Most of our designs require simple signal-processing techniques. Many of our modulations are based on the presence or absence of certain frequencies, and as such, the receivers simply perform a number of computations of the form

$$R_i = \frac{\left| \sum_{n=0}^{N} r(n) \exp(j2\pi f_i n) \right|}{N}$$

where $r(n)$ is the received signal, f_i the expected frequency, and N the number of samples in a symbol duration. Although an explanation of this technique falls outside this article's scope, this equation illustrates the computational overhead at the receiver for the simple designs.

Experimental environment and error rates

In our experiments, we've been using desktops and laptops with processing speeds above 800 MHz running Windows and have been satisfied with their performance. The decoder for all the coding schemes is based on a simple correlation receiver.[11] For the pentatonic scale and bird song designs, we implement the decoders in Java, and we can decode all messages in real time with practically no errors. We get these results while running several other applications, such as MicroSoft Outlook and Internet Explorer. We've also implemented the pentatonic design in C++ for two versions of the iPAQ—the 36 series (a StrongARM processor at 206 MHz) and the 39 series (an XScale processor at 400 MHz)—both running Windows. In these implementations, the receiver program first records the message and then decodes it. The 36-series iPAQ takes about 6 seconds to decode a 20-character (160-bit) message, whereas the 39 series takes about 2 seconds. Again, the error rate is practically zero.

Error rates, in general, depend on many factors. As expected, error rates are lower for lower bit rates at the source. For example, for modems described in the "Musically oriented variations of common modulation techniques" section, the harmonic ASK design (bit rate of 1,280 bps) produced error rates of around 30 percent, whereas the pentatonic scale and harmonic FSK designs (400-bps bit rates) generated no errors in a relatively quiet office environment. In general, our modems operate with almost no errors at bit rates up to 800 bps.

Limitations

The channel we are dealing with includes the air—the transmission media—as well as the hardware used for producing and capturing the sound. Because we use ordinary speakers, microphones, and sound cards, this channel is far from ideal, and channel characteristics vary from device to device. Most of the existing hardware can handle sounds for humans (voice, music, and so on) with a quality acceptable to the human ear. The human ear is almost deaf compared to artificial decoders, especially those modeled after traditional digital communication receivers. Therefore, channels that target humans can cope with several simplifications and tolerate several imperfections.

For example, even though the human audible sound band extends to 20 KHz, many existing audio channels, such as telephones, TVs, and desktop computers, are band-limited to lower frequencies—typically 4, 8, and 12 KHz, respectively. This occurs because the lower bands can carry speech and music signals with acceptable perceptual quality. We considered these limitations when designing our modems: all of the modems operate below 12 KHz, with almost all of them operating below 8 KHz (exceptions are the harmonic ASK and Bach designs), and several operating below 4 KHz.

Experiments involving video equipment and TVs attest to the feasibility of these modems. We've recorded images with our sounds embedded (pentatonic scale sounds) using an ordinary video camera and later played it on a regular TV set while running the decoder on a laptop. The decoder performed with no errors.

These channels' limitations are not restricted to band. The existing hardware is usually nonuniform with respect to frequency responses and often introduces nonlinearities in the channel. This makes it harder to implement the decoders, because we cannot rely on consistent responses when we vary the devices.

These problems are solvable, however. For example, the transmission power at certain frequencies is not always within the same limits for all hardware. We include calibration sequences in the beginning of the signals and normalize the decoding to those values. This solves two problems at once: the inconsistency of the hardware response and the effect of the distance between speaker and microphone.

In designing aerial acoustic modems for ubicomp, we must accept a tradeoff between data rate and aesthetics.

In designing aerial acoustic modems for ubicomp, we must accept a trade-off between data rate and aesthetics: the faster the transmission, the more likely the sounds will be annoying, and vice-versa. Rather than seeing this as a setback, we see this as a challenge for designing pleasant sounds that are robust enough to survive transmission. We expect that sound designers—from music composers to HCI developers—will affect the kinds of sounds acoustic modems will use.

We envision applications of these acoustic modems in several domains, including security (information warfare), entertainment, ubiquitous Internet, and medical informatics. This will require a lot more work, and we've clearly only started exploring the possibilities.

Future work might examine the effect of other sounds on the data rate. Aerial acoustic channels can be very noisy. The sound of a TV in the living room, for example, competes with the sounds of people talking, children playing, objects falling on the floor, and so on. When transmission frequencies collide, error rates increase. Hotel rooms and offices are usually much quieter than homes, with only occasional bursts of sound as objects move.

Microphone quality can also affect data rate. The more sensitive the microphone, the more noise it picks up. Thus, a cheap microphone closer to the sound source might be better than a high-quality microphone as it will naturally filter most of the ambient noise.

We also need good noise models for these channels. The additive white Gaussian noise model, used widely in communications, misses these channels' most important noise components. We need to study the channels more carefully so we can design sounds that are robust with respect to the most likely ambient noise for each application.

Future work might also explore the role of human perception and its effect on sound design. What kinds of sounds can transmit information to machines while "leaking" some of the information to people? On the other hand, what sounds are easiest to ignore? For this, we can borrow some findings from auditory studies, but we plan to conduct our own studies, especially with respect to the disclosing of auditory information to people.

ACKNOWLEDGMENTS

Many thanks to our group of students who have implemented some of the modems: Ling Bao, Patricio de la Cuadra, Natacha Domingues, João Lacerda, Peter Hebden, and Jon Froehlich. Also thanks to Diana Smetters and the security group at PARC for interesting discussions concerning security applications. Part of this work was performed while Lopes was a PARC research staff member.

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific Am.*, vol. 265, no. 3, Sept. 1991, pp. 94–104.
2. J. Foerster et al., "Ultra-Wideband Technology for Short- or Medium-Range Wireless Communications," *Intel Technology J.*, 2nd quarter 2001, http://intel.com/technology/itj/q22001/articles/art_4.htm.
3. R. Scholtz and M. Win, "Impulse Radio," *Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm.*, IEEE Press, 1997.
4. C. Lopes and P. Aguiar, "Aerial Acoustic Communications," *Proc. IEEE Int'l Workshop Applications of Signal Processing in Audio and Acoustics*, IEEE Press, 2001, pp. 219–222.
5. D. Balfanz et al., "Talking to Strangers: Authentication in Ad Hoc Wireless Networks," *Proc. Network and Distributed Systems Security Symp.*, Internet Soc., 2002; http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/balfan.pdf.
6. G. Jones, *Music Theory*, Harper & Row, 1974.
7. J. Moorer, "Signal Processing Aspects of Computer Music—A Survey," *Computer Music J.*, vol. 1, no. 1, 1977, pp. 4–37.
8. N. Domingues et al., "Aerial Communications Using Piano, Clarinet, and Bells," *Proc. IEEE Int'l Workshop Multimedia Signal Processing*, IEEE Press, 2002, pp. 460–463.
9. J. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *J. Audio Eng. Soc.*, vol. 21, no. 7, 1973, pp. 526–534.
10. J. McClellan, R. Schafer, and M. Yoder, *DSPfirst: A Multimedia Approach*, Prentice-Hall, 1999.
11. B. Sklar, *Digital Comm.*, Prentice-Hall, 1988.

For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

the AUTHORS

Cristina Videira Lopes is an assistant professor in the School of Information and Computer Science, University of California, Irvine. She conducts research in ubiquitous computing and in programming languages. She has a PhD in computer science from Northeastern University. She is a member of the IEEE and the ACM. Contact her at the School of Information and Computer Science, Univ. of Calif., Irvine, CA 92697; lopes@ics.uci.edu.

Pedro M.Q. Aguiar is a researcher in the Institute for Systems and Robotics, Lisbon, Portugal, and an auxiliary professor at the Instituto Superior Técnico (IST), Technical University of Lisbon. His research interests are in the statistical processing of multimedia signals. Aguiar has a PhD in electrical and computer engineering from IST. He is a member of the IEEE. Contact him at Instituto Superior Técnico / ISR, Av. Rovisco Pais, 1049-001 Lisboa, Portugal; aguiar@isr.ist.utl.pt.