Gestures as Input: Neuroelectric Joysticks and Keyboards

EMG technology helps capture gestures as input for virtual joysticks and keyboards and thus could lead to new applications in flight control, space, and the video game industry.

Kevin R. Wheeler and Charles C. Jorgensen
NASA Ames Research Center

Today's laptops and PDAs have more expansive screen sizes and decreasing physical dimensions, all the while becoming increasingly computational. Computers are now small enough to be worn, and—when combined with display goggles—provide a truly mobile computing experience. Their smaller size has opened an opportunity to explore new methods for inputting data to wearable devices (see the "Input Methods for Wearable Devices" sidebar).

We've developed an approach at the Neuroengineering lab at NASA for designing and using neuroelectric interfaces for controlling virtual devices. This approach uses hand gestures to interface with a computer rather than relying on mechanical devices such as joysticks and keyboards. Our system noninvasively senses electromyogram (EMG) signals from the muscles used to perform these gestures. It then interprets and translates these signals into useful computer commands. We've chosen to demonstrate virtual joysticks and keyboards because the average computer user has had some experience with them.

We're certainly not the first group to suggest that gestures can be used as a form of computer input. Most gesture-recognition systems receive input in one of two forms:

- Through an external camera that requires sophisticated image processing and controlled lighting (such as Intel's Open Source Computer Vision Library system)
- Through a sensing glove on the participant's hand (such as Fifth Dimension Technologies' Data Glove)

Both techniques have fundamental difficulties that we want to avoid. We need to recognize gestures in poor lighting conditions in extreme environments outside of the lab without encumbering our hands or using invasive procedures. Our hands must be free so we can use them for tasks such as grasping handholds and operating equipment other than the computer.

Have virtual joystick, will travel

Because our laboratory has done quite a bit of flight control work, we could readily access several high-fidelity flight simulators. This let us test a virtual joystick's efficacy with simulators having a fidelity (class IV) sufficient for pilot training. We first had to decide how to sense the EMG signals. Several standard methods for noninvasive sensing exist. The typical approach uses medical grade wet electrodes that stick to the skin. Although these give a very good signal-to-noise ratio, they aren't appealing for a regular, long-term user interface. Another approach employs dry metal electrodes. For the joystick work, we fabricated a dry electrode sleeve as shown in Figure 1. We outfitted this sleeve with many different electrode positions to facilitate research, using only four pairs of electrodes.

EMG electrodes work by detecting skin currents

Published by the CS and Communications Society ■ 1536-1268/03/$17.00 © 2003 IEEE

Input Methods for Wearable Devices

Commercially available input methods for wearable devices include a screen for drawing graffiti (such as the Palm OS), very small keyboards (such as SFR's BlackBerry), alternative keyboards (such as Handkey Corporation's Twiddler), and voice recognition software (such as IBM's ViaVoice). Other alternatives allow touch typing through light projection (such as Virtual Devices' VKey, http://virtualdevices.net) or by wearing a motion-sensing device on the palms (such as the Senseboard Technologies' Virtual Keyboard, http://senseboard.com). However, the QWERTY keyboard remains the standard data-entry device. Voice-recognition software can facilitate some data entry, but activities such as programming, numeric entry, and scientific tasks require keyboard entry.

Each technology comes with problems. With interfaces that require memorizing a new syntax (such as graffiti), you might forget the syntax of a rarely used but necessary symbol. You also might have difficulty forming the graffiti strokes rapidly and consistently. Alternative keyboards remove the need to memorize a new writing style but often force users to hold their hands and manipulate their fingers in stressful ways that cause cramping.

Most work that we've seen using EMG electrodes to interface with a system has come out of the prosthesis community. A recent PhD thesis on this topic[1] discusses many of the same issues that we've experienced while trying to use EMG technology in a computer interface. We haven't seen a prosthesis using noninvasive technology that lets users type as if using a computer keyboard.

REFERENCE

1. N. Daisuke, "Studies on Electromyogram to Motion Classifier," doctoral dissertation, Hokkaido Univ., Autonomous Systems Eng. Laboratory, 2001.

resulting from an action potential that travels in a muscle fiber from the innervation point to the end of the muscle. The electrodes must have a low-impedance connection with the skin, which can be disrupted by hair and dry skin. An emerging technology might change this finicky sensing by using new noncontact electrodes that don't require a transfer of current (we'll describe them later).

Methodology

Our methodology consisted of the following steps:

1. Select gestures
2. Apply electrodes (location and number)
3. Acquire signals
4. Filter and digitize data
5. Form features
6. Perform pattern-recognition model training and testing
7. Apply pattern recognition in interactive simulation.

Gesture selection. We used four basic coarse-grained gestures to mimic manipulating a joystick:[1] up, down, left, and right, with varying degrees of force. We simplified gesture recognition by using only four gestures and four pairs of electrodes to provide reasonable separation between gestures.

Electrode application. We placed the electrodes in relationship to the gestures we wished to recognize according to individual physiological differences. We sewed the dry electrodes into a sleeve as Figure 1 shows. This sleeve helped reduce variation in electrode placement. Individual differences in personal physiology proved challenging. Differences in arm lengths and widths made it difficult to place the electrodes at proper positions across people without considerable effort. In addition, EMG signal strengths varied across people and the amount of training they received.

Signal acquisition, filtering, and digitization. We followed standard operating procedures for obtaining EMG data by collecting it through differential pair electrodes (one pair is one EMG channel) with a preamplifier located near each electrode pair. We typically spaced the electrode pairs 2 to 2.5 centimeters apart. We used a Bortec Octopus unit to sample the EMG signal at 2,000 Hz and amplify it by a factor of approximately 2,000. We then used a 16-bit data-acquisition card to digitize it

Figure 1. Dry electrode sleeve for flying a modified F15 aircraft.

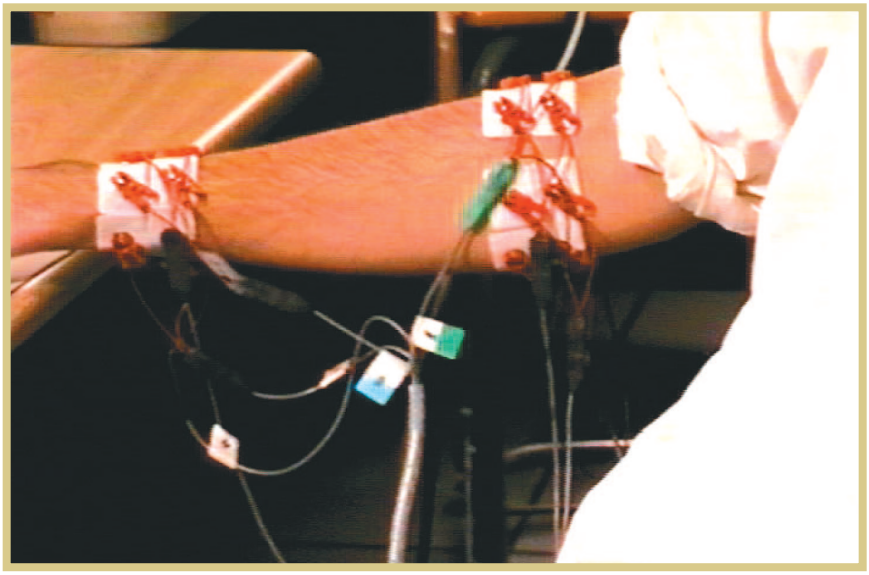

Figure 2. Wet electrodes in two rings about the forearm of a typing participant. Four pairs of standard medical wet EMG electrodes circle the wrist.

and subsequently processed it on an Intel Pentium-based PC. The PC first digitally filtered (using antialiasing) and then transformed and recognized the patterns. It then transmitted the patterns to other machines for simulation display and control.

Feature formation. This step separates the signals enough to let the pattern-recognition module distinguish between gestures. This transformation also creates a space smooth enough to be reliably modeled. We tried many common methods such as short-time Fourier transform, wavelets, moving averages, and autoregression coefficients. In the end, the simplest feature space seemed the best: overlapping the rectified EMG's moving averages.

Pattern recognition. We chose a hidden Markov model (HMM) that the speech-recognition community developed to solve their pattern-recognition time-series problem.[2] The history of speech recognition reveals a process that first attempted to recognize isolated words from a single speaker, then isolated words from multiple speakers, followed by continuous words from a single speaker, and finally, continuous words from multiple speakers. We followed a similar approach, developing isolated gesture recognition for both a single participant and multiple participants. (The work we describe here concerns continuous recognition for the joystick study and isolated recognition for a single typist in the keyboard study.)

We tried to minimize contributions to data variation. For example, electrode placement that drifts from day to day can vary signal statistics. Using a fixed electrode sleeve can help. You probably can't remove day-to-day variations related to natural behavior; in fact, we would benefit from modeling them. For example, the way people gesture can vary slightly from day to day even though they intend to perform the gestures identically. In this case, we need enough data to represent the multimodal statistics as well as a way to daily adapt the system model. Our methodology doesn't vary adaptively yet. Our best remedy is recognizing when day-to-day variation is too great for adequate model generalization. We can then employ less data for training using only data similar to our current day's setup (that is, electrode locations). This method succeeded in the technology's initial demonstrations.

The HMMs we used were continuous, tied-mixture,[3] left-to-right models. We also used standard Baum-Welch training.[2] The models are called *continuous* when they use inputs that can take on a range of floating-point values. The alternative to this is to allow for only discrete values found when quantization has transformed the input. *Tied-mixture* means that a fixed number of Gaussian mixtures are used throughout all states. Thus, any state can use any mixture. In a *left-to-right model,* the HMM doesn't revert to a previous state but remains in a current one or goes on to another.

The joystick work consisted of using nine discrete states with 27 total mixtures. We performed model initialization using k-means clustering, partitioning the states to equalize the variance amount within each state. We segmented the training data sets to ensure that the variance's peak was near the middle of each segment. This centered the bulk of the energy. We sampled segments at 2,000 Hz that contained 3,072 samples per channel, with a maximum of eight channels. We typically varied the HMM parameters: the number of discrete states, Gaussian mixtures, and maximum iterations to train; the method used to arrive at the state partitioning (uniform versus variance based); and the method used to initialize the mixture's parameters (for example, k-means clustering). We performed the real-time recall using the standard Viterbi algorithm.[4]

Experiments

After we fitted the four pairs of dry electrodes within a sleeve attached to the participant's forearm, we asked the participant to pretend moving a joystick left, right, up, and down. The participant performed each gesture 50 times per day. We collected the data over several days. The sleeve position and skin condition naturally varied with each day. We separated the data by gesture and segmented it to have peaks in the center of 3,072 sample segments. We removed artifacts or incomplete gestures from the data sets through manual inspection. We then used these sets to train four HMMs, one for each gesture. These trained models then recognized gestures made on a day excluded from the training set using the same dry electrode sleeve. We used the recognized joystick gestures to perform numerous real-time demonstrations of fly-

ing a simulated 757 transport aircraft to landing.[1] We implemented a more continuous gesture-recognition process by decreasing the segment size and integrating the EMG energy as a relative force measure. Most consumers are familiar with joysticks that translate movement into control (such as pitch and bank-rate control for aircraft). However, some force-oriented commercial command sticks move very little but translate the amount of force into rate control. We combined these two ideas by recognizing the acting pilot's EMG as a gesture with an associated force.

To test online pattern recognition, we trained on a previously acquired day of data. We then used these trained models in our real-time simulation for flying an aircraft.

Results

The live-demonstration performance didn't equal that of batch data sets, which are usually collected under static conditions. Our live demonstrations involved imperfect electrode placement and high stress, where a participant is bombarded with questions and distractions.

Error rates determined from batch test sets don't necessarily indicate real-time performance. In particular, error rates varied across time depending on many factors such as sleeve position (rotation), sweating, dry skin, how long the electrodes were worn, and fatigue (resulting in tremors). By training on only one day's data (with its associated sleeve position and skin condition), we could use the dry electrode sleeve for demonstrations on many different days. We selected the day that gave the best real-time reliability. The recognition was significantly accurate to perform public demonstrations of flying a 757 transport aircraft to landing at a simulation of San Francisco Airport. Accuracy was usually in the 90th percentile range, depending on the pilot's mood and the dry electrodes' condition. The input command rate didn't differ quantifiably from a standard joystick's, but a hardware joystick's resolution and precision is finer than in the EMG-based system. After participants use our system for a couple hours, they can reliably maneuver an aircraft with our virtual stick technology. The aircraft's response to gesture commands also becomes a form of biofeedback the participants can use to modify their behavior. The biggest system limitation was the sensing technology's unreliability, which is currently being resolved (as described later).

Typing on your knee

We wanted to extend the joystick's coarse-grained gestures to the finer-grained motor control necessary for typing. This proved considerably more difficult. To ease the task, we typed on a numeric keypad like the one on workstation keyboards. We abandoned the dry electrode sleeve for wet electrodes to improve reliability and the signal-to-noise ratio. Figure 2 shows how the eight pairs of electrodes were typically placed. The electrode positions varied somewhat because we had to apply the electrodes each morning. This meant the HMMs had to adapt each day to account for position differences. Nonetheless, we demonstrated typing recognition using eight channels of electrodes with 11 HMMs. We trained each HMM for a particular keystroke, in this case "0" through "9," and "Enter." Of course, this typing demonstration only works with touch-typing skills, which raises an open question about how much feedback is necessary for users to successfully navigate new interfaces.

After participants use our system for a couple hours, they can reliably maneuver an aircraft with our virtual stick technology.

We performed the typing-recognition task because people are very familiar with it. Ultimately, as wearable computers evolve, we'll discover interfaces more natural than the QWERTY keyboard mechanical joysticks and mice. Interfaces will evolve to take advantage of gesture-recognition capabilities. This resembles the technology evolution when the computer mouse was introduced to dumb terminals and workstations. The operating systems' front ends evolved to take advantage of the mouse-based cursor control capabilities in terms of graphical windows and icons. Along with the ability to perform device input without a mechanical interface come numerous applications that could help reduce the size of many common devices such as cellular phones and PDAs.

Methodology

We used the same methodology employed for the joystick work except when otherwise noted.

Experiments

The typing experiment used one ring near the wrist and a second near the elbow. The participant touch-typed on a printed picture of a numeric keypad, striking the keys "0" through "9" and "Enter." The participant typed these in order, separated by a one-second rest interval, for a total of 40 strokes on each key. We segmented this data and manually removed the artifacts. We collected data on several different days. We trained 11 HMMs, one for each gesture, consisting of six states with 18 mixtures.

Results

The participant maintained a touch-typist hand position above the simulated number pad. If the hand position varied, distinguishing between hitting the top row of keys and the bottom row became greatly more difficult, requiring electrodes on the upper arm to sense the movement. The participant also had to be careful to maintain the wrist angle to avoid radically changing the sampled signals. Even with careful attention to position and maintenance of electrode placement from day to day, the data tended to vary. However, being able to detect a severe wrist angle that might lead to carpal tunnel injury was

TABLE 1

Multitrial confusion matrix for typing data.

	Recognized keystrokes for keypads 1-9									Correct (%)
Keypad number	1	2	3	4	5	6	7	8	9	
1	46	0	0	4	0	0	1	0	0	90.2
2	0	48	0	0	0	0	0	3	0	94.1
3	0	0	49	0	0	1	1	0	0	96.1
4	11	0	0	38	2	0	0	0	0	74.5
5	1	3	0	5	36	1	3	2	0	70.6
6	0	1	6	0	0	42	0	0	2	82.4
7	0	0	0	0	0	0	51	0	0	100.0
8	0	0	0	0	2	1	3	44	1	86.3
9	0	0	0	0	0	0	0	0	51	100.0

a positive outcome. The gesture interface itself can perform ergonomic assessment and advise the user on better posture.

Multitrial acquisition and testing. The keyboard-replication experiments had greater daily variation in electrode placement than the joystick experiments. We also had reliability difficulties because the participants didn't maintain a consistent hand position from trial to trial. This included ensuring similar wrist angles and that the hand was consistently either not resting on the table during motion or was partly supported by the table (bad form, but consistent bad form). Given all these difficulties, the resulting confusion matrix shown in Table 1 for multiple trials looks pretty good. A perfect result would list the number of keystrokes only along the diagonal. When the keystroke for "1" occurred, the system correctly recognized "1" 46 out of 51 times, then incorrectly recognized it as "4" four times and as "7" once. This resulted in correct classification 90.2 percent of the time. The data variability caused our models to generalize gestures, which caused more confusion. For live demonstrations, we needed to train on the same day that we were using the system and thus employed only one day's data. However, the training took only a few seconds.

Figure 3. Noncontact sensor from Quantum Applied Science and Research Corp. The sensor is 3 × 2.5 centimeters.

System usage testing. We haven't used the virtual keypad very much for data entry. In our trials, it's been convenient to type anywhere, including on one participant's lap. Three major drawbacks delay usage of EMG-based typing as a regular interface. The first, as mentioned earlier, is the sensing technology's reliability and ease of use. The second is processing time. In our implementation on a 400-MHz Pentium PC, the recognition delays are on the order of 700 milliseconds. However, because PCs are now at least six times faster and the technology continues to become more efficient, we believe this is a short-term problem. The final problem is missing the touch feedback that occurs when a typist recognizes a keystroke's completion as the key stops moving at the bottom of the stroke. Missing this feedback is a direct consequence of replicating a keyboard action rather than a more natural gesture interface. Keyboards inherently give the user touch feedback that lets them modify system usage. However, as the system migrates to more natural gestures, this touch feedback won't be necessary. For example, the science fiction movie *Minority Report* showed a glove-based optical system that used natural gestures to manipulate video streams. The operator employed distinct gestures for rewinding, fast-forwarding, copying, pasting, and freezing video frames. We're exploring more natural gestures as an interface to a data-exploration environment we're developing for manipulating data views. Suitable software is unavailable for exploring these interfaces, which inspired us to develop a system.

Challenges

Using wet electrodes caused unintentional misplacement that greatly degraded our recognition performance. Standard EMG dry electrodes incorporated into a sleeve alleviated this problem but then raised significant reliability issues in signal

sensing. We're working with Quantum Applied Science and Research to develop a dry electrode sleeve with a cloth layer between the electrode and skin to let clothing sense EMGs. Figure 3 shows an initial prototype sensor.

A second enhancement includes a model-correcting adaptation now common in the speech-recognition community. This adaptation lets models tune to small variations throughout the day and also to differences between training models and the current day's configuration. We're also working on a calibration stage so that when a participant gestures to issue a certain command, the computer adapts to understand that these signals are that command. This frees us from requiring a participant to learn a fixed set of gestures. The person will perform a natural gesture to accomplish a given task, and the computer will simply map those signals to the correct action.

A less mainstream application we've been considering isan interface to a wearable robotic exoskeleton. DARPA has funded several projects to develop an exoskeleton that lets soldiers carry heavier loads over greater distances. The human-interface issues for such a system are quite extreme. It would be ideal if the exoskeleton's limbs would move in correspondence to the user's limbs. One approach is to use the very signals generating the muscle movement to control the exoskeleton.

A more ambitious idea for reconfigurable airplanes and other transportation machinery is a virtual wearable cockpit or command center. The US Air Force and other military branches increasingly use unmanned vehicles for surveillance missions. One way to control these systems from the field is a wearable cockpit. You could use a wearable computer with a wireless link and display goggles, and then employ EMG-based gestures to manipulate the switches and control sticks necessary for flight. Noncontact EMG sensors sewn into the field uniform could then sense movements as the acting pilot pretended to manipulate control inputs.

A space-based application could let astronauts type into a computer despite being restricted by a spacesuit. If a depressurization accident occurred on a long-term space mission and astronauts needed to access onboard computers, they could use EMG electrodes within their spacesuits to replicate a computer interface.

In human-centered solutions such as a gesture-based interface, the system customarily compensates for individual differences between users to produce a consistent pattern-recognition rate no matter who is using the system. However, in the case of security, you can take advantage of user differences to prevent unauthorized users. You could also do this by monitoring EMG signals corresponding to typical computer command sequences. The EMG signals have different signatures depending on age, muscle development, motor unit paths, skin-fat layer, and gesture style. The external appearances of two peoples' gestures might look identical, but the characteristic EMG signals are different.

In terms of fun applications, the video game industry constantly needs quick, flexible interfaces. New input devices such as the Xbox controller are pushing the limits by increasing the complexity of numerous physical buttons and sticks manipulated simultaneously. We think it is possible to map multiple muscle groups to different actions to distribute this complexity across the body. This would require training for proficiency, but the net result would be a whole new gaming experience.

We're now prototyping an EMG-based mouse and speaker-recognition system. The mouse can act as both a mouse and an aid to monitor and alert users to potential ergonomic injuries. When someone is notified of a potentially harmful movement, the system can substitute a different movement to perform the same computer input simply by changing EMG signal mapping to computer commands.

We've shown how to use EMG technology to replicate traditional joystick and keyboard interfaces. We don't advocate this as the next great interface technology—only that more natural fine-grained gestures can be enlisted to manipulate computer function and cursor position. A video depicting this work is available online at http://ic.arc.nasa.gov/projects/ne/videos/NECD320x240.3.mov. P

the AUTHORS

Kevin R. Wheeler is group lead of the Extension of the Human Senses group in the Computational Sciences Division at NASA Ames Research Center. His research interests include pattern recognition in nonstationary time-series, Bayesian inference, and nonlinear dynamics. Wheeler received a PhD in electrical engineering from the University of Cincinnati. He is a member of the IEEE. Reach him at Kevin.R.Wheeler@nasa.gov.

Charles C. Jorgensen is chief scientist of the Neuroengineering laboratory in the Computational Sciences Division at NASA Ames Research Center. His research interests include neural network damage, adaptive control of high-performance air and spacecraft, EMG/EEG human-interface technologies, and spiking neuron computing architectures. Jorgensen received a PhD in mathematical psychology from the University of Colorado, Boulder. Reach him at cjorgensen@mail.arc.nasa.gov.

REFERENCES

1. C. Jorgensen, K. Wheeler, and S. Stepniewski, "Bioelectric Flight Control of a 757 Class High Fidelity Aircraft Simulator," *Proc. World Automation Congress, 3rd Int'l Symp. Intelligent Automation and Control*, TSI Press, Albuquerque, New Mexico, 2000.
2. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, Feb., 1989, pp. 257–286.
3. J.R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, Dec., 1990, pp. 2,033–2,045.
4. G.D. Forney, Jr., "The Viterbi Algorithm," *Proc. IEEE*, vol. 61, no. 3, Mar. 1973, pp. 268–278.

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