Controlling a Wheelchair Indoors Using Thought

Brice Rebsamen, Chee Leong Teo, Qiang Zeng, and Marcelo H. Ang Jr., *National University of Singapore*

Etienne Burdet, *Imperial College London*

Cuntai Guan and Haihong Zhang, *Institute for Infocomm Research*

Christian Laugier, *INRIA Rhône-Alpes*

This brain-controlled wheelchair prototype uses a P300 EEG signal and a motion guidance strategy to navigate in a building safely and efficiently without complex sensors or sensor processing.

Amyotrophic lateral sclerosis, or ALS, is a degenerative disease of the motor neurons that eventually leads to complete paralysis. We're developing a wheelchair system that can help ALS patients, and others who can't use physical interfaces such as joysticks or gaze tracking, regain some autonomy. The system must be usable in hospitals and homes with minimal infrastructure modification. It must be safe and relatively low cost and must provide optimal interaction between the user and the wheelchair within the constraints of the brain-computer interface. To this end, we've built the first working prototype of a brain-controlled wheelchair that can navigate inside a typical office or hospital environment (see figure 1).[1]

This article describes the BCW, our control strategy, and the system's performance in a typical building environment.

Background

Research on brain-controlled interfaces has flourished in recent years. For instance, BCI researchers have placed neural implants in the brains of animals and humans to control simple mechanisms.[2] Because such invasive techniques are still risky, human BCI research has focused mainly on noninvasive methods for monitoring brain activity, such as electroencephalography (EEG), magnetoencephalography, near-infrared spectroscopy, and functional magnetic resonance imaging.

Our wheelchair uses an EEG-based BCI—a simple, portable system providing continuous measurement of brain activity. EEG is the most common approach to building BCIs, and researchers have developed various techniques to use EEG for communication between a human and a machine.[3]

As with other BCIs, EEG yields a low information transfer rate: either the waiting time between consecutive commands is long, typically several seconds, or uncertainty about the command is high. The difficulty is figuring out how to use such a poor signal to control a wheelchair that requires real-time specification of its position within the 3D space of planar motion. One solution is to give the system some autonomy, such that the user must provide the wheelchair with directives only from time to time.

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

Figure 1. The brain-controlled wheelchair uses a compact acquisition device and an embedded computer.

For example, in the work of José del R. Millan and his colleagues, an EEG BCI based on recognizing three mental states interacts continuously with a mobile robot's automatic behavior to successfully maneuver in a simple environment.[4] However, the motion depends on sensor processing, which isn't foolproof. So, this type of control might be too risky for a wheelchair. Besides, this approach requires the user to be constantly alert, which will cause stress. Similarly, using an EEG BCI to directly choose the wheelchair's next move[5] necessitates a series of decisions to complete even a simple movement, thus possibly exhausting the subject.

Our control strategy relies on a slow but safe and accurate P300 EEG BCI that lets the user select a destination item on a menu. The wheelchair then moves to the corresponding target on a predefined path. This strategy requires minimal effort from the user. The paths are software defined and not hard coded, so they can easily be modified if the environment changes.

Properties of the P300 BCI

An EEG-based BCI is particularly well suited for our wheelchair system because it can deliver a continuous time signal and the necessary hardware is portable. A set of electrodes on a cap (see figure 1) is wired to an amplifying, filtering, digitizing device, which transfers the signals (such as the ones shown in figure 2a) to a computer for analysis. The associated electronic equipment is smaller than a laptop and weighs less than one kilogram.

The *P300 evoked potential* is a well-studied, stable brain signal. This natural, involuntary response of the brain to infrequent stimuli can provide a BCI with an *oddball paradigm*. In this paradigm, a random sequence of stimuli is presented, only one of which interests the subject. Around 300 milliseconds after the target flashes, there is a positive potential peak in the EEG signal. When the system detects a P300 signal (P for positive, 300 for the 300-millisecond delay), it determines that the target stimulus occurred 300 ms earlier.

Figure 2a shows the raw EEG signal from 10 electrodes. The vertical red and green lines mark the times of target and nontarget stimuli, respectively. The system can't distinguish the P300 signal from background activity, but averaging several samples attenuates the uncorrelated activity, noise, and signal variability.

Figure 2b shows the signal that results from averaging hundreds of samples corresponding to target events (red curve) and nontarget events (green curve). The P300-based BCI requires no user training and only a few minutes to calibrate the detection algorithm's parameters. This is noteworthy because other BCI techniques require a long training phase, up to several months in the case of slow cortical-potential devices.[6]

We use a visual oddball paradigm. Our system displays items to be selected on a screen, randomly flashing them one by one (see figure 3). To select an item, the user focuses on it; a simple way to focus is to count the number of times the target flashes. The user may gaze at a location other than the target because the P300 measures surprise, not a direct visual signal.

Main features

Our BCW uses the asynchronous P300 system.[7] The system must be asynchronous so that the user can intervene at any time. First, the system filters and cleans the signals from 15 EEG electrodes placed on the top of the head. It then segments the signals to associate each button on the GUI with a sample corresponding to data 150 to 500 ms after that button flashes. These samples are fed to a support vector machine, which computes a score expressing the likelihood that the sample contains a P300.

After each *epoch* (the period during which all buttons are flashed once), the support vector machine outputs new scores for all buttons. When one or more scores are higher than a decision threshold, the system designates the button with the maximum score as the target.

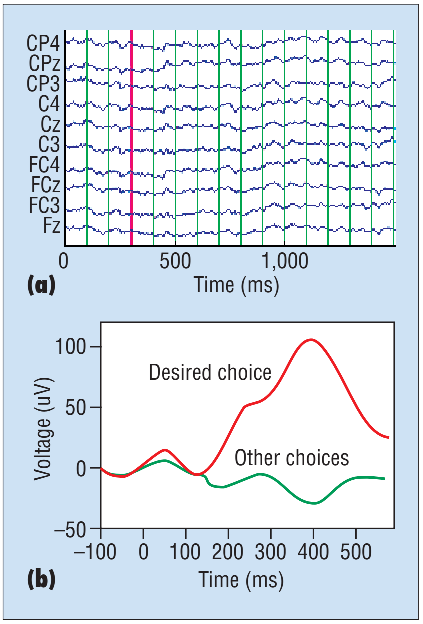

Figure 2. (a) The raw EEG signal from 10 electrodes. Vertical lines mark the times of stimuli, the red line corresponding to a target stimulus. (b) Averaging several epochs cancels out uncorrelated noise. The EEG signal shows a potential peak 300 ms after the target is presented (red curve), whereas it remains relatively flat at other times or when other items are presented (green curve).

Three factors are critical for the P300 BCI:

- Error rate (Err): the ratio of wrongly se-

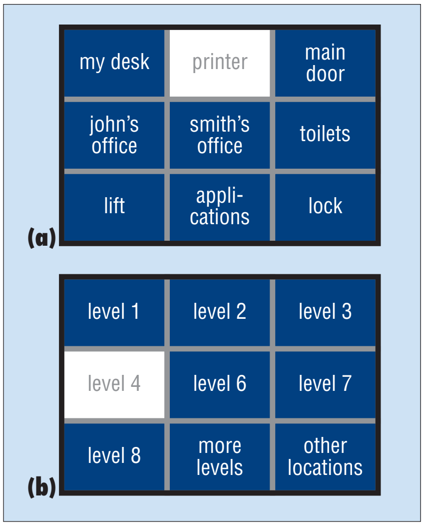

Figure 3. Context-dependent menus of commands displayed to the subject. The items flash in a random sequence onscreen, and the P300 BCI selects the item on which the subject is focusing. (a) This menu corresponds to the wheelchair navigating on one floor. (b) This menu corresponds to the wheelchair that is currently at the elevator's fifth floor, which is thus not displayed. Note that the number of commands isn't limited to nine.

lected targets (substitution) divided by the total number of selections during an experiment. We keep this error rate low by using the moving average of scores over the last eight epochs to select the target.

- Response time (RT): the time before a button's (averaged) score reaches the decision threshold.
- False-acceptance rate (FA): the number of times per 100 seconds that the system wrongly detects a P300 signal.

These features depend on the decision threshold's value. A low threshold leads to a fast selection but might produce a lot of errors and a high FA. Conversely, a high threshold leads to a long RT (possibly no response at all) and a low FA.

Selecting a decision threshold

To determine a suitable threshold for our application, we measured RT, FA, and Err as a function of the threshold. We recorded the EEG signals from five young, healthy subjects while they were selecting buttons on the interface and while they performed other mental tasks such as reading or relaxing.

Figure 4 shows the experiment's results. For each subject, both the P300 and non-P300 scores are approximately normally distributed, and the mean of the P300 scores is larger than the non-P300 mean. The RT curve is larger than 8, the averaging window's width. RT increases with the threshold because fewer samples have a high score. Conversely, FA is close to 100 percent for threshold values lower than the score distribution's center and tends to zero for high threshold values. Err is below 10 percent and decreases for large threshold values.

Which threshold value to choose depends on the application and desired performance. For a speller, where backspacing is possible, choosing a threshold value that yields short response times is desirable, even if the interface must commit some errors.

To control the wheelchair, we focus more on reliability and thus choose a relatively high threshold value, yielding a low Err. Our results demonstrate that for a threshold that keeps FA around 2.5 percent, RT is approximately 20 seconds, which is completely acceptable. For example, the waiting time in elevators or for a green light on the street is on the order of tens of seconds or minutes.

Simple collaborative control

Standard wheelchair control requires the user to provide commands continuously, and even a slight error can cause an accident. However, faultless continuous brain control isn't feasible because current BCIs aren't fast or reliable enough. So, we must provide sufficient autonomy to navigate the wheelchair between two commands or to correct an erroneous interpretation of the brain signal.

A conventional approach to autonomy is to equip the vehicle with sensors to perform obstacle detection and localization (that is, determine its own location relative to some reference coordinate system). The robot must have sufficient artificial intelligence to generate a suitable trajectory to reach a destination while ensuring safety during motion. However, this conventional strategy has a heavy cost (both financial and computational), and the system's decision might seem awkward to a human observer. For example, autonomous vehicles sometimes have refused to move forward because of obstacles that a human driver would easily be able to navigate around.

Furthermore, a robotic wheelchair's mission is to assist a person, who might decide to change course at any time—for instance, to stop on the way to the mail room, pick up a book in the library, or go to the toilet. The quality of interaction between the user and the robotic wheelchair might well determine whether a person will adopt it.

For all these reasons, we decided to develop a safe motion-control strategy using only simple sensors, relying on collaboration with the user to solve complex situations that will arise from time to time.

Motion guidance

The wheelchair user can't continuously issue commands, so we simplify motion control by using a set of predefined paths between different relevant locations in the user's daily environment. Once the user selects a path, the system drives the wheelchair along it using a dedicated path-following controller.[8]

If a building plan is available, the wheelchair system can create a collection of guiding paths (as shown on the map in figure 5) automatically. Alternatively, we can create a map using *walkthrough programming*[8]: the onboard computer records the trajectory while a helper pushes the wheelchair between two locations. A cubic B-spline is least-squared fit to this path and serves as a guide for subsequent movements. B-spline is a piecewise polynomial function, locally simple, yet smooth and globally flexible. A few control points (four for a cubic B-spline) act as attraction points to the curve, so they have an intuitively geometrical meaning and can be used to modify a guiding path.

You can easily extend this map to include paths to neighboring offices or spaces. We developed a tool-equipped path editor that lets wheelchair users or helpers easily modify the guiding paths to adapt to environmental modifications such as changes in furniture locations or obstacles.[8]

Context-dependent menus

Navigating with the BCW is straightforward: from the current location, the computer scans the locations linked by guiding paths and displays the list on the P300 interface, as illustrated in figure 3a. When the user chooses a destination, the wheelchair heads toward it, following the appropriate guiding path.

The menu on the interface is context dependent. When the BCW stops in front of an obstacle, it displays options to solve the situation. When it's at an elevator, the GUI shows a list of the floors (see figure 3b). We assume a smart elevator able to communicate wirelessly with the wheelchair computer and equipped to control the wheelchair's entrance and exit. The P300 BCI we're using can select 20 or more buttons, so the user can select the elevator level in one step. So, the strategy seems particularly adapted to motion planning in an office-like building or in a hospital. In contrast, a BCI based on two or three states (such as other researchers have used[4,5]) would require several successive selections.

Adapted control hierarchy

To enable the user to modify a command after issuing it, we devised a faster P300 paradigm that issues a stop command in a few seconds. Because the most relevant command during motion is "stop," the GUI presents one stop button and eight dummy buttons (to comply with the oddball paradigm).

However, because the response time is still on the order of several seconds, and to provide redundant safety, we implemented a stop reflex based on simple sensors on the front of the wheelchair. We used a low-cost module of three infrared proximity sensors with a range of 50 cm and a combined angular range of 180 degrees.

Once the wheelchair is stopped in front of a detected obstacle, the GUI asks the user what action to perform:

- resume movement along the path if the obstacle has disappeared (for example, a human blocking the way left),
- avoid the obstacle by applying an elastic deformation to the path[8] to the left or right, or
- call for assistance.

Because of the variety of possible situations, we rely on the user's cognitive abilities rather than let the artificial system compute a solution on the basis of its generally poorer inference and sensory capabilities.

In summary, we adapted the control scheme to the P300 BCI's properties to minimize the user's input while letting him or her remain in charge of major decisions. Long-term commands such as choosing a destination or an elevator floor are selected by the user with a high decision threshold, yielding a mean response time of 20 seconds and an Err of 2.5 percent. While the BCW is moving, we use a lower threshold setting (faster but with false alarms) to issue a stop command within a couple of seconds. Simple reflex proximity sensors guarantee safety. This collaborative control strategy lets the user operate the wheelchair safely and efficiently while requiring little intervention, and it can be used with various types of robotic wheelchairs.

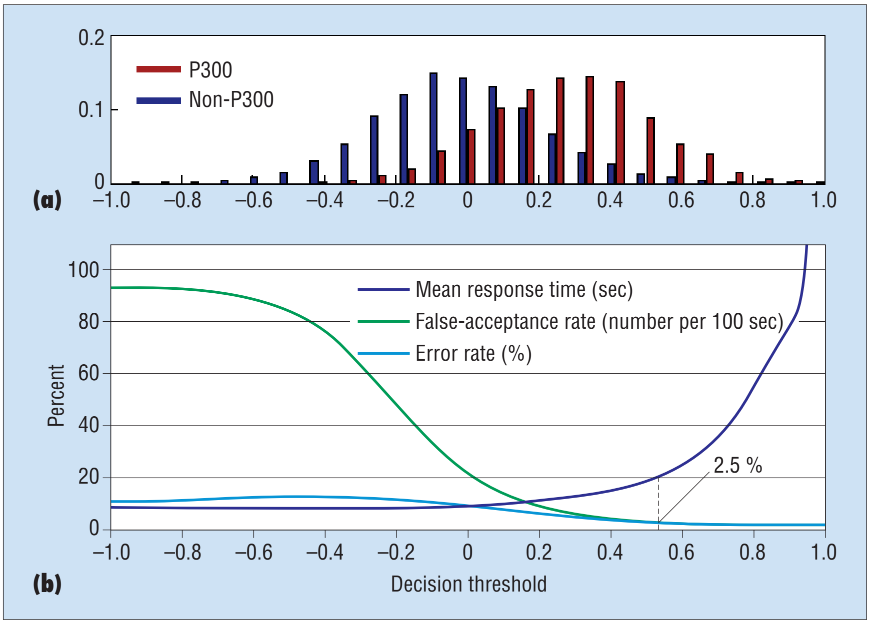

Figure 4. (a) Distribution of a typical subject's scores for the samples that contain a P300 (corresponding to the highest score) and those that don't. (b) Characteristics of the BCI vary with the threshold. The mean response time, error rate, and false-acceptance rate averaged over the five subjects. For a 20-second response time, the error and false-acceptance rates are as low as 2.5 percent.

Implementation

We built the BCW prototype[8] on a Yamaha JW-I power wheelchair. The real-time control program is written in C and runs on a Toshiba M100 laptop with a Pentium 1.2 GHz processor operated by Ubuntu Linux 6.06 with a 2.6.15 kernel patched with Real-Time Application Interface v3.3 for real-time capabilities. We limit the sensors to two optical rotary encoders attached to specially designed glidewheels for odometry and a bar code scanner (Symbol M2004 Cyclone) for global positioning.

This scanner, similar to models used in supermarkets to read price codes, is mounted below the seat. Bar code patterns are placed at critical locations such as doors. Each set of bar code patterns has a unique code corresponding to global coordinates that we've entered into the memory. Combining information from these two simple sensors, the system provides sufficiently accurate wheelchair positions and orientations at speeds up to 0.6 meters per second.[8]

For EEG acquisition, we use Neuroscan's NuAmps, a high-quality, inexpensive 40-channel digital EEG amplifier that's capable of 22-bit sampling at 1,000 Hz, measuring signals from DC to 260 Hz.

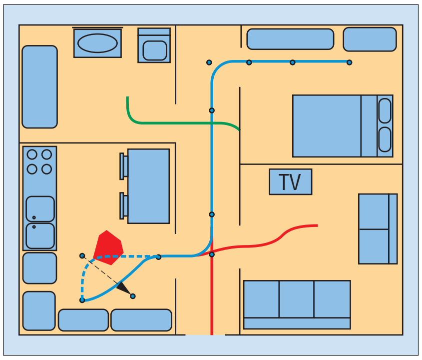

Figure 5. Sample map with a guiding path in a home environment. The control points can be used to modify the path in the kitchen to avoid a large obstacle.

Locking scheme for greater reliability

The wheelchair user might want to stay in a particular location and perform some activity—for example, work with a computer. To prevent falsely accepted commands to set the wheelchair in motion, we implemented

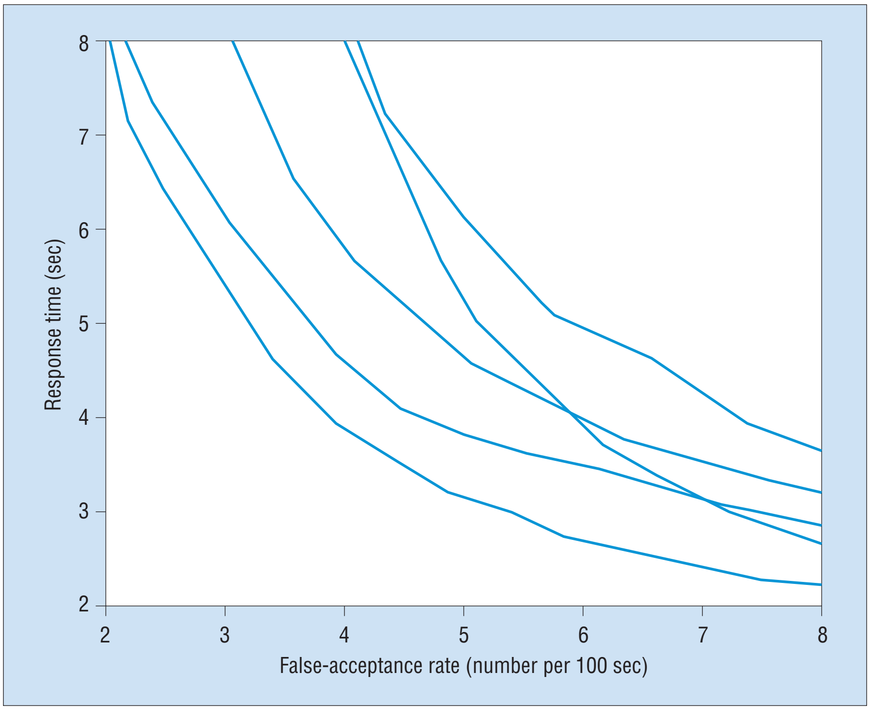

Figure 6. Response time (RT) as a function of the false-acceptance rate (FA) for five subjects, measured with the stop interface used during movement.

an interface-locking scheme similar to the keyboard-locking facility on cellular phones.

The user locks the interface using one of the nine buttons on the menu. Once it's locked, the user can't issue a command before entering a sequence of keys. The number of false selections of the unlocking key is FA/9. For three keys in the unlocking sequence, we can estimate the number of false instances of unlocking as $(FA/9)^3$. This results in less than 1 accidental unlocking in 100 hours.

A faster interface for stopping

Using only one button to stop the wheelchair during movement has two advantages. First, the false-acceptance rate is divided by nine because false P300s are distributed evenly among the nine buttons, only one of which triggers an action. So, using a lower threshold to reduce the response time is possible. Second, because the user can select only one button, no error is possible, and averaging the scores over eight epochs isn't necessary.

Figure 6 shows RT as a function of FA for the stop interface, measured for five subjects. We see that it's possible to achieve a response time of less than six seconds while keeping FA below six occurrences per 100 seconds.

Although a response time of six seconds (corresponding to three meters at the maximal speed of 0.5 meters per second) is acceptable, five false alarms in 100 seconds is relatively high. After the user has selected stop via the interface, the same menu remains, and the wheelchair will pursue the movement if the user doesn't confirm by selecting the stop button a second time. This avoids disruption, particularly in long movements.

Navigation with the wheelchair

We tested the BCW in an office environment with healthy subjects. This environment included several floors connected by an elevator. At each floor, we created four destinations interconnected by six guiding paths. We designed the paths before the experiment using the walkthrough programming method we explained earlier.

We asked subjects to move from one location to another on a different floor. We manually operated the elevator as well as the wheelchair's entrance into and exit from it. Subjects reported that activating the commands was easy. All subjects succeeded in their first attempt to reach the desired locations, taking approximately 15 seconds to issue a command. A video demonstrating these capabilities is available.[9]

The human-machine collaboration inherent in our system was designed to use both the wheelchair system and the user to the best of their abilities. Context-dependent menus enable them to communicate, letting the user select the necessary commands corresponding to the current situation. This lets the user decide the next action, relying on his or her superior sensing and inference capabilities, while the BCW is in charge of executing them safely and reliably, thus compensating for the user's motor disability.

Motion guidance provides efficient control while requiring little input, so it's adapted to the BCI's slow information-transfer rate. This also avoids costly and potentially unsafe complex sensor processing. If the user changes his or her mind on the way, he or she can stop the wheelchair in a few seconds using a fast P300 stop button. IR proximity sensors stop the wheelchair immediately if it encounters a physical obstacle.

The system is easy to set up: it doesn't require environmental modifications, and a map of guiding paths can be designed using simple tools. Our experimental results showed that the system is also easy to use. Because the user only has to select the destination and deal with unexpected situations, the system requires minimal input and concentration. Besides, because the BCW repeats movements along the same paths over time, its motion is predictable, so the user can relax during the movement.

Our current BCI might appear to suffer from a slow RT. However, assess this in the context of potential users. The system is intended for people who can't move at all and are normally stuck in bed; their notion of time differs from ours, and being able to move independently within their environment represents a much-improved quality of life, whether it takes time or not. In this context, safety and reliability are much more important than speed.

Nevertheless, the algorithms to detect P300 signals leave room for improvement. In particular, we're working on using signals corresponding to buttons neighboring the target to reduce the time to issue a command and the number of involuntary commands.

Finally, conducting experiments with disabled users who really need a BCW is essential, because they might respond differently from the healthy individuals we've worked with so far. Our simple system is well suited to performing such experiments, and their results will influence future development.

The Authors

Brice Rebsamen is a doctoral candidate at the National University of Singapore. His research focuses on rehabilitation robotics. He received his engineering diploma in electronics from ENSEIRB, France, and a master's in cognitive sciences from the Grenoble Institute of Technology. Contact him at the Dept. of Mechanical Eng., Faculty of Eng., 9 Eng. Dr. 1, Singapore 117576; brice@nus.edu.sg; http://guppy.mpe.nus.edu.sg/~rebsamen.

Etienne Burdet is a senior lecturer in human robotics at Imperial College London. His main research interest is human-machine interaction. He received his PhD in robotics from ETH Zurich. Contact him at the Dept. of Bioengineering, Imperial College London, SW7 2AZ London, UK; e.burdet@imperial.ac.uk; www.bg.ic.ac.uk/staff/burdet.

Cuntai Guan is a senior scientist at the Institute for Infocomm Research, Singapore, where he built the Neural Signal Processing group. His research focuses on brain-computer interfaces, neural rehabilitation, machine learning, pattern classification, statistical signal processing, and interactive and digital media. He received his PhD in electrical and electronic engineering from Southeast University. He's a senior member of the IEEE. Contact him at the Inst. for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613; ctguan@i2r.a-star.edu.sg.

Haihong Zhang is a research fellow in the Neural Signal Processing group at the Institute for Infocomm Research, Singapore. His research focuses on reliable and efficient brain-computer interfaces, including machine learning and pattern recognition in brain signals. He won an award in the international BCI Competition and a Tan Kah Kee Young Inventor's Award (Merit). He obtained his PhD from the National University of Singapore. Contact him at the Inst. for Infocomm Research, 21 Heng Mui Keng Terr., Singapore 119613; hhzhang@i2r.a-star.edu.sg.

Chee Leong Teo is an associate professor in the National University of Singapore's Department of Mechanical Engineering and the director of NUS Overseas Colleges. His research areas are mechanical-systems control and human-machine interfaces. He received his PhD in mechanical engineering from the University of California, Berkeley. Contact him at the Nat'l Univ. of Singapore, Dept. of Mechanical Eng., Singapore 119260; mpeteocl@nus.edu.sg.

Qiang Zeng is a doctoral candidate in rehabilitation robotics at the National University of Singapore. His research interests are in human-machine interaction and rehabilitation robotics. He received his master's in technological design in mechatronics from the National University of Singapore and Eindhoven University of Technology. Contact him at the Dept. of Mechanical Eng., Faculty of Eng., 9 Eng. Dr. 1, Singapore 117576; zengqiang@nus.edu.sg; http://guppy.mpe.nus.edu.sg/~zengqiang.

Christian Laugier is a research director at INRIA (the French National Institute for Research in Computer Science and Control) and leader of the robotics project team at INRIA Rhône-Alpes. His research focuses on motion autonomy, intelligent vehicles, decisional processes, and virtual reality. He received the Nakamura Prize for advancing the technology on intelligent robots and systems. He received his PhD and State Doctorate in computer science from Grenoble University. Contact him at the e-Motion Project-Team, INRIA Rhône-Alpes, 655 ave. de l'Europe, Montbonnot, 38334 Saint Ismier Cedex, France; christian.laugier@inrialpes.fr; http://emotion.inrialpes.fr/laugier.

Marcelo H. Ang Jr. is an associate professor in the National University of Singapore's Department of Mechanical Engineering. His research interests include robotics, mechatronics, automation, computer control, and applications of intelligent systems methodologies. He's also the founding chairman of the Singapore Robotic Games and a member of the IEEE. He received his PhD in electrical engineering from the University of Rochester. Contact him at the Dept. of Mechanical Eng., Nat'l Univ. of Singapore, 9 Eng. Dr. 1, Singapore 117576; mpeangh@nus.edu.sg; http://guppy.mpe.nus.edu.sg/~mpeangh.

References

1. B. Rebsamen et al., "A Brain Controlled Wheelchair Based on P300 and Path Guidance," *Proc. 1st IEEE/RAS-EMBS [IEEE Robotics and Automation Soc. and IEEE Eng. in Medicine and Biology Soc.] Int'l Conf. Biomedical Robotics and Biomechatronics* (BIOROB), IEEE Press, 2006, pp. 1001–1006.
2. "Focus on Brain-Machine Interfaces," *Nature*, 13 July 2006, pp. 164–171; www.nature.com/nature/focus/brain.
3. J.R. Wolpaw et al., "Brain-Computer Interface Technology: A Review of the First International Meeting," *IEEE Trans. Rehabilitation Eng.*, vol. 8, no. 2, 2000, pp. 164–173.
4. J.D.R. Millan et al., "Noninvasive Brain-Actuated Control of a Mobile Robot by Human EEG," *IEEE Trans. Biomedical Eng.*, vol. 51, no. 6, 2004, pp. 1026–1033.
5. K. Tanaka, K. Matsunaga, and H.O. Wang, "Electroencephalogram-Based Control of an Electric Wheelchair," *IEEE Trans. Robotics*, vol. 21, no. 4, 2005, pp. 762–766.
6. N. Birbaumer et al., "A Spelling Device for the Paralysed," *Nature*, 25 Mar. 1999, pp. 297–298.
7. H. Zhang, C. Guan, and C. Wang, "A Statistical Model of Brain Signals with Application to Brain-Computer Interface," *Proc. 27th Ann.*

Int'l Conf. IEEE Eng. in Medicine and Biology Soc. (EMBS 05), IEEE Press, 2005, pp. 5388–5391.

8. Q. Zeng et al., "Design of a Collaborative Wheelchair with Path Guidance Assistance," *Proc. 2006 IEEE Int'l Conf. Robotics and Automation* (ICRA 06), IEEE Press, 2006, pp. 877–882.
9. B. Rebsamen et al., "Navigating a Wheelchair by Thought in a Building;" http://guppy.mpe.nus.edu.sg/~brice/videoBCW.

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