vehicle CAN bus. The navigation algorithm waits for the reports. If the steering commands are executed correctly and in a timely manner, the drive commands are sent as usual. If a timeout report is received, the navigation algorithm calls other function modules to handle this situation, during which the information from the IMU, machine vision, and AGDGPS132 are fused.

ACKNOWLEDGMENT
This work is supported by the National High-Tech Research and Development Program of China (2006AA10A305) and the National Natural Science Foundation of China (50575145).

REFERENCES

AUTHOR INFORMATION
Chengliang Liu (chlliu@sjtu.edu.cn) received the Ph.D. from Southeast University, NanJing, China, in 1999. He is a professor in the School of Mechanical Engineering at Shanghai JiaoTong University, China. His research interests include agricultural robots, remote monitoring and control, intelligent maintenance, and mechatronics.

Mingjun Wang received the B.Tech. from Shanghai Jiaotong University, China, in 2004. He is currently a Ph.D. candidate in the Department of Mechanical Engineering. His research interests include computer vision, motion control, and robotics.

Jun Zhou received the Ph.D. from NanJing Agricultural University in 2003. He is currently a professor in the department of agricultural machinery in NanJing Agricultural University. His research interests include image processing and computer vision.

Robo Armwrestler
CHUL-GOO KANG

An arm-wrestling robot called Robo Armwrestler (Figure 1) has recently been developed in our Intelligent Control and Robotics Laboratory to benefit the health care of senior citizens. The motivation for this project, which is supported by the Korean Government, is to reduce social welfare costs and to improve the quality of life of the elderly population by meeting their physical and mental needs. In recent decades, Korea’s aging population has increased by 35%, well over the standard 30% for an aging society. Our vision is to realize humanoid robots that have entertaining functions such as arm wrestling and chess playing, as well as service functions such as errands. Over time, we hope the robot will help both the elderly and the disabled.

Several years ago, Y. Bar-Cohen issued a challenge to build a robot using muscles of electrically activated polymers that could arm wrestle a human [1]. As a result of this challenge, a few arm-wrestling robots were built using electroactive polymers (EAPs) [2]. The primary object of these arm-wrestling robots is to demonstrate the potential of EAP technology. These robots, however, do not have a broad range of functions related to arm-wrestling skills. Another effort relating to a humanoid robot arm has been in the field of prosthetic devices, such as the Utah Artificial Arm [3]. However, it does not appear that prosthetic devices are suitable for arm wrestling, in which strong arm force is required.

Several practical arm-wrestling devices have been patented as amusement units or as units for developing and strengthening wrist and arm muscles [4]–[7]. These devices are classified roughly into three types according to

FIGURE 1 The arm-wrestling robot Robo Armwrestler. This robot, which is intended for use in elderly health care, was developed in the Intelligent Control and Robotics Laboratory at Konkuk University in Korea.

Digital Object Identifier 10.1109/MCS.2008.927963
the methods of generating reaction force against a player’s arm force.

The first type of arm wrestling device uses a spring force; a typical example is U.S. Patent 3,947,025 [4], in which the device is comprised of a helical coiled spring that has adjustable stiffness. The second type of device uses pneumatic or hydraulic cylinders, which is better than the spring type from the viewpoint of force manipulability. However, a disadvantage is that the system becomes complicated and bulky because of the need for a supplementary device for pneumatic or hydraulic pressure generation. An example of the hydraulic device is U.S. Patent 5,842,958 [5]. The third type of device uses electric motors instead of springs or pneumatic/hydraulic cylinders in order to generate resistive force against the user. Examples include Japan patent 6-315544 [6] and Japan patent 2002017891 [7], in which torque motors are used to generate arm force, and sensor plates and photosensors are used to detect arm speed to prevent a throw fracture of the player.

These patented devices are invented for playing simple arm-wrestling games or for practicing strength training, in which they usually generate fixed force levels. If the player generates a stronger force than the arm wrestling device, then he will win; otherwise he will lose. Unfortunately, players become easily bored after a few trials.

Robo Armwrestler has features that remedy the above deficiencies. In particular, Robo Armwrestler automatically generates the force level appropriate to each person after sensing the human arm force at the early stage of the match, and therefore any person with either a strong or weak arm force can enjoy arm wrestling without any parameter changes. Furthermore, Robo Armwrestler’s generated force profile varies with each match, so each person can enjoy arm wrestling with the robot for a long time without becoming bored. The winning average of the robot is determined randomly at the start of the match, but the robot measures the person’s will to win during the match, and this result influences the winning average of the robot. For example, the robot’s arm force becomes weaker and the winning probability of the human increases if the human tries to win the match over a period of time.

The robot recognizes a human’s approach and begins to talk, saying, for instance, “Hello, welcome to Robo Armwrestler! If you want to try arm wrestling, please sit down on the chair.” The device also recognizes the person sitting on the chair and guides them on how to play and enjoy arm wrestling. The tone of the voice varies occasionally so as not to be tedious. The facial expression of the avatar changes synchronously according to arm-wrestling situations.

**HARDWARE DESIGN**

Robo Armwrestler can detect the maximum arm force of the user in the early stage of a match and can automatically and randomly generate a different arm-wrestling scene each time so that the user cannot predict a force pattern in advance. The robot then executes force feedback control to implement the scenario using signals related to the motion of the mechanical arm and to the torque acting on the mechanical arm. The arm-wrestling robot is thus composed of an arm-force generation mechanism, a control system, and an inference engine.

The arm-force generation mechanism comprises a servomotor, a position/velocity sensor, a speed reducer, a torque sensor, three inclinometers, a mechanical arm, and an adapter with a mechanical stopper [8]. The electric servomotor provides necessary torque according to motor control input signals, while the position/velocity sensor detects angular position and angular velocity of the motor. The speed reducer decreases the speed and increases the torque of the motor. The torque sensor detects the torque acting on the mechanical arm, and the adapter with a mechanical stopper is utilized to restrict the angular range of motion of the mechanical arm to guarantee the safety of the user. The inclinometers are used to initialize the absolute angle of the robot arm. The arm-force-generation mechanism makes the mechanical arm rotate in either a clockwise or a counterclockwise direction.

An incremental encoder is used as the position/velocity sensor for high resolution. A harmonic drive is selected as the speed reducer instead of conventional gears, which have large backlash and thus limit torque control performance. The adaptor with the mechanical stopper is further utilized to set the initial absolute angle of the mechanical arm by means of low-speed control of the motor. The initial setting of the absolute angle of the arm can also be achieved redundantly by using multiple inclinometers.

The torque sensor installed between the speed reducer and the mechanical arm must have reasonable resolution to obtain reasonable force-control performance. Two ultrasonic sensors, which are attached at the right and left sides on the front of the table, detect a human’s approach within a prescribed range of angles near the arm-wrestling robot. Ultrasonic sensors have the advantage of high noise immunity compared to other types of sensors and can easily measure the distance of an approaching human under any circumstances. One photoelectric sensor using infrared rays detects the human sitting on a chair. To guide the player, the display monitor and speakers are visibly positioned on the table.

The arm-force generation mechanism is shown in Figure 2, while the schematic of Robo Armwrestler is shown in Figure 3.

**FORCE CONTROL SCHEME**

To make the arm-wrestling game attractive, we add force control functions to the system in such a way that force command is generated intelligently and so that real-time tracking control can follow the given force command. Linux and RTAI (real-time kernel) are adopted for the operating system of the robot. When Linux and RTAI are
implemented on a desktop PC with a 2.8-GHz Pentium IV, timing errors less than ±0.02 ms occur in generating 5-ms timer interrupts. However, Windows XP has roughly ±1-ms latency when the same program is executed with the same PC platform.

A block diagram of the force feedback control scheme is shown in Figure 4. The control system receives feedback signals of actual position, velocity, and torque and calculates torque command using feedback signals and scenarios. The system then controls the mechanical arm by generating the motor control input using force control logic. The force control logic is basically PID type but uses velocity and position information together with force information.

Force control performance depends mainly on the accuracy of the feedback signals and real-time control capability, including the accuracy of the sampling time, as well as the force feedback control logic itself. Force feedback control plays a key role in arm wrestling the robot, but position feedback control is also necessary for rotating the mechanical arm to a starting position and setting the initial absolute angle of the mechanical arm before a match. For this servo control, the motor driver supplies analog voltage signals corresponding to angular speed as well as quadrature encoder signals (A, B, and Z). We generate digital position and velocity signals using the quadrature signals, since the generated digital signals are more accurate than the analog voltage signals.

As shown in Figure 5, the control system of the robot is comprised of a CPU, memory, amplifier, logic circuit, pulse-generation circuit, and output ports. The CPU produces a motor control input using the control program and the feedback signals and produces voice and image signals. The memory stores a control program including control logic and scenarios. The amplifier amplifies the low-level voltage signal coming from the torque sensor and achieves signal conditioning. The logic circuit conditions the feedback signal from the position/velocity sensor, while the pulse-generation circuit produces a pulse signal for the ultrasonic sensors. Voice speakers and a display monitor are driven by the CPU through output ports.

The torque sensor, inclinometer, photoelectric sensor, and ultrasonic sensor signals are converted to digital signals through analog-to-digital (A/D) converters and transmitted to the CPU. Encoder signals are transmitted to the CPU directly through digital input channels. The motor driving signals are converted to analog voltages through a digital-to-analog (D/A) converter and transmitted to the motor driver.

When the CPU is down, the D/A converter can still output the last signal of the motor control input, and thus a dangerous situation can occur if electric power is applied to the motor. To resolve this problem, the
CPU transmits an initialization completion signal to the motor power control through a D/A converter and sends a zero value to the motor drive through the D/A converter when the initialization procedure is completed, where the initialization procedure starts when the main switch is pressed. The motor power controller turns on the mechanical relay (MR) to supply electric power to the motor according to the output signal of the solid-state relay (SSR), which in turn is actuated by the initialization completion signal.

User safety is thus guaranteed even if the motor power switch is turned on before completing the initialization procedure or at abnormal conditions of the control system since electric power is not transmitted to the motor.

When using the incremental encoder as the position/velocity sensor, we initially set the absolute zero-degree angle of the mechanical arm using the mechanical stopper and velocity feedback control. More specifically, the controller slowly drives the motor clockwise or counterclockwise using position feedback control and measures the torque value by means of the torque sensor. If the measured torque is larger than the threshold value, then the controller sets the present angular position as the absolute angle, since the large measured torque implies that the mechanical stopper has reached the stopper seat block.

Initial setting of the absolute arm angle can also be accomplished redundantly using multiple inclinometers, but in this case the arm-force generation mechanism becomes more complicated, and possibly more expensive.

**INTELLIGENT SCENARIOS**

Arm wrestling is not strictly a strength sport, as people often think, because technique and speed are also very important. Therefore, a key element of an arm-wrestling robot is to create an intelligent game scenario online. In Robo Armwrestler, the inference engine generates an appropriate force profile for each match, which considers a human’s maximum force, force pattern, time duration, will to win, and randomness. Force control logic enables the robot arm to follow the generated force profile as smoothly as possible. In Robo Armwrestler, winning or losing is not predetermined but varies online according to the pattern of the game progression.
For a few seconds in the early stage of the game, the inference engine detects the human’s maximum force according to the procedure shown in Figure 6. The system increases force up to the specified value with a parabolic fashion for a short time, and then the robot measures arm velocity every 0.1 s for the next few seconds. If the velocity is positive, then the robot increases the force until the velocity becomes negative, and records the force value.

To realize unpredictable and intelligent game patterns, we adopt a random number and a value called will point, which quantifies the determination of the arm wrestler to win the match. If the will point is near 100, the user is considered to have a strong desire to win the match. If the will point is near zero, the user is considered to have a weak desire to win the match. The will point is calculated by

\[
\text{will point} = \frac{\text{average arm force during one subscenario}}{\text{maximum arm force of the user}} \times 100.
\]

The game scenario is composed of several subscenarios. Each subscenario has a different rotation angle limit of the robot arm within 150° and is divided into three classes: win, draw, and defeat. If a subscenario is selected by means of random number generation, the robot decreases or increases the force during randomly determined intervals of time. During these intervals, human force is measured and averaged to calculate the will point. As soon as the execution of a subscenario is completed, the next subscenario is immediately prepared. This subscenario can be generated online at that instant or can be selected among many subscenarios prepared in advance.

The win subscenario implies a significant decrease of the torque command value, the defeat subscenario implies a significant increase of torque command value, and the draw subscenario implies a small increase, a small decrease, or no change of torque command value. The grouping of win, draw, and defeat subscenarios depends on the present arm angle, and is achieved using a prescribed rule.

Arm-wrestling progression is affected by the will point and pre-specified probability. For example, if a will point of 86 is obtained, then the class of win, draw, or defeat scenario is determined according to winning probability with 8%, drawing probability with 90%, and defeat probability with 2%. This class determination is conducted using a random number between 0 and 99, that is, the generated random numbers 98 and 99 imply the defeat class, random numbers 8 to 97 imply the drawing class, and random numbers 0–7 imply the winning class.

Using another random number, we select a subscenario randomly within subscenarios of the determined class. If the selected subscenario is a drawing one, then the will point is recalculated after the subscenario ends, and the above procedure is repeated. If the selected subscenario is a win or a defeat, then the win or defeat subscenario is employed, and the human wins or defeats, and the match ends. Finally, the arm-wrestling system is initialized for the next match. Figure 7 shows the flowchart of these arm-wrestling scenarios.

Instead of selecting a subscenario among three classes, we can generate a subscenario online, which is characterized by three parameters, force increment, rise time, and maintenance time, as shown in Figure 8. These parameters are determined using a random

**FIGURE 7** Flowchart showing the progression of robotic arm wrestling. The winning average is determined by random numbers and will points.
FIGURE 8  Force increment pattern. The pattern is composed of a smoothly increasing force interval and an interval of constant force.

FIGURE 9  Beginning procedure of robotic arm wrestling. This procedure includes automatic recognition of a human’s approach to the robot, as well as the human sitting in the chair.

FIGURE 10  Robotic arm-wrestling results of a 72-year-old woman. (a) Arm-wrestling picture, (b) a case where the woman wins the game after 31 s, (c) a case where the woman loses the game after 21 s.
number and the will point. The force increasing or decreasing in the subscenario is achieved by smooth polynomial curves as shown in Figure 8.

During an idle situation, the robot plays music and waits for a person to appear. As a person approaches the robot, the system automatically detects the approach, greets him or her, and encourages the person to arm wrestle. If the person sits down, the robot guides him/her to start the game. This procedure is shown in Figure 9.

DEMONSTRATION

Experiences at the laboratory and the Future Tech Korea Exhibition reveal that Robo Armwrestler generates intelligent scenarios unpredictably and reliably and controls the robot arm force properly so that the human arm wrestler feels as if he or she is arm-wrestling against a human.

Figure 10(a) shows a 72-year-old woman playing against Robo Armwrestler. Figure 10(b) and (c) shows match results of Figure 10(a), in which the torque command, angular velocity, and arm angles are plotted with respect to time. Figure 10(b) corresponds to the case in which the woman loses, while (c) corresponds to the case in which the woman wins. From these graphs, we can see that force patterns generated by the arm-wrestling robot and the elapsed time of arm wrestling are different from match to match even if the same person plays.

Figure 11 shows a result of the match between a strong 25-year-old adult and Robo Armwrestler. Figure 12(a) and (b) illustrates a match between a ten-year-old child and Robo Armwrestler. As soon as the adult finishes the match in Figure 11, the child begins the match shown in Figure 12(a). Although the adult produces roughly 50 N-m and the child roughly 20 N-m, the arm wrestling proceeds smoothly without changing any parameters of Robo Armwrestler. In other words, Robo Armwrestler generates an appropriate force level automatically according to the magnitude of each player’s arm-force.

Table 1 shows the elapsed time and winning/losing probability of each match when one user plays arm wrestling ten times with Robo Armwrestler. We can see that the elapsed time for a match varies each time, and the result of the match also varies each time, so that the enjoyment of arm wrestling can be maintained for a long time.

<table>
<thead>
<tr>
<th>Trials</th>
<th>Elapsed Time</th>
<th>Result</th>
<th>Trials</th>
<th>Elapsed Time</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.0 s</td>
<td>Win</td>
<td>6</td>
<td>17.6 s</td>
<td>Lose</td>
</tr>
<tr>
<td>2</td>
<td>14.0 s</td>
<td>Win</td>
<td>7</td>
<td>14.5 s</td>
<td>Lose</td>
</tr>
<tr>
<td>3</td>
<td>20.0 s</td>
<td>Win</td>
<td>8</td>
<td>7.8 s</td>
<td>Lose</td>
</tr>
<tr>
<td>4</td>
<td>47.0 s</td>
<td>Lose</td>
<td>9</td>
<td>23.4 s</td>
<td>Win</td>
</tr>
<tr>
<td>5</td>
<td>20.0 s</td>
<td>Win</td>
<td>10</td>
<td>11.5 s</td>
<td>Win</td>
</tr>
</tbody>
</table>

FIGURE 11 Outcome of a young adult arm wrestling. In this robotic arm-wrestling contest, the 25-year-old adult loses the game after 28 s.

FIGURE 12 Robotic arm-wrestling result of a ten-year-old boy. (a) Arm-wrestling picture and (b) a case where the boy wins the game after 23 s.
Table 2 summarizes results for two users arm wrestling 26 times each with Robo Armwrestler. The table shows that the first user has a 63% probability of winning, while the second user has a 75% winning probability.

CONCLUSIONS
In this article we have presented the system design, implementation, force feedback control, and generation of intelligent arm-wrestling scenarios of an arm-wrestling robot. Although Robo Armwrestler works as expected with the designed autonomy and reasonable control performance, we plan to further pursue research to provide the device with the ability to recognize facial expressions of a human using a Webcam and thus emotionally communicate with the player. Moreover, we plan to add degrees-of-freedom for more human-like motion and to eventually integrate arm-wrestling functions into a humanoid robot.

ACKNOWLEDGMENTS
The author acknowledges financial support from the Seoul R&BD Program (Project Number 10848). He also thanks Ki-Seon Ryu, Young-Woo Kim, Ik-Xu Son, Eun-Jun Park, Hae-Lyong Jung, Ho-Yean Kim, and Han-Seung Lee for their help.

REFERENCES

AUTHOR INFORMATION
Chul-Goo Kang (cgkang@konkuk.ac.kr) received the B.S. and M.S. in mechanical design and production engineering from Seoul National University, Korea, in 1981 and 1985, respectively, and the Ph.D. in mechanical engineering from the University of California, Berkeley, in 1989. He joined Konkuk University, Seoul, Korea, in 1990, where he is a professor and the director of the Future Robot Research Center and the Intelligent Control and Robotics Laboratory. His research includes intelligent robots, force/torque sensing, and brake systems of high-speed trains. He is an editor of the Journal of the Korea Fluid Power Systems Society, a board member of the Korea Robotics Society, and a Congress Highlight program editor of the IFAC 2008 World Congress.

Global Challenge
The hunting problem was grave, of course, because it threatened the productivity (quantitative and qualitative) of nations. Tens of thousands of engines worldwide were misbehaving, and so the mathematicians, scientists, and engineers of several countries sought a solution to the problem. One promising approach was to design an improved governor that eliminated hunting entirely, instead of simply trying to mitigate the bad behavior, by eliminating the source of hunting. The source, ultimately, was the way in which engine speed depended on load. If a steam engine/governor system could be designed so that engine speed was independent of load, then hunting would not arise. Much effort went into designing such engine/governor systems. The German William Siemens and his brother spent over 30 years designing governors, trying to solve the problem of hunting. He described a related problem associated with the Watt type of governor, that of offset. He said that the Watt governor “… cannot regulate, but only moderates the velocity of the engine, that is, it cannot prevent a permanent change in the velocity of the engine when a permanent change is made to the load upon the engine” (Bennett). An astatic governor, one that yielded constant engine speed whatever the load, would solve this problem as well as hunting. The French physicist M. L. Foucault devoted much thought to the design of astatic governors.