Int. J. Engng Ed. Vol. 24, No. 1, pp. 92–100, 2008
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Omni-directional Robotic Wheel—A Mobile Real-Time Control Systems Laboratory*

BHARATH RAMASWAMY,[1] YANGQUAN CHEN[2] and KEVIN L. MOORE[3]
[1]*Qualcomm Incorporated, San Diego, CA 92121 USA. E-mail: bharath.ramaswamy@qualcomm.com*
[2]*Department of Electrical and Computer Engineering, Utah State University, Logan UT 84322 USA. E-mail: yqchen@ece.usu.edu*
[3]*Division of Engineering, Colorado School of Mines, Golden CO 80401 USA. E-mail: kmoore@mines.edu*

A mobile laboratory was developed for students of the ECE5320 Mechatronics and ECE7750 Distributed Control Systems courses at Utah State University. A serial server was connected to the microcontroller of a stand-alone omni-directional robotic wheel assembly. This enabled communication between the wheel and any remote computer, via a wired or wireless Internet connection. A telepresence control system and a prototype networked control system (NCS) were developed and tested. This system was suitably modified to accommodate the needs of the course laboratories, thereby enabling students to design, debug and test their laboratory project in real-time from the comfort of their own locations.

Keywords: mobile laboratory; robotics; NCS; mechatronics

INTRODUCTION

DEVELOPMENT OF THE INTERNET and advances in mobile communications technology have led to rapid progress in the use of telepresence systems for distance learning and remote experimentation purposes. The advantages offered by remote laboratories over conventional and virtual laboratories include cost-reduction and more efficient usage of laboratory equipment, reduced maintenance and flexible and self-paced learning, but with use of versatile, real-world equipment. These advances have also greatly benefited the field of control systems education. For example, a typical use of a commercial platform like National Instruments' LabVIEW^{TM} environment to develop a remote teaching laboratory can be seen in [1].

To provide a satisfying, real-world experience for the student, a remote laboratory should incorporate several features such as data-collection facility, live video and audio streaming, safety and stability measures, regulation of access and collaboration support. In view of this, a remote learning application called 'Second Best to Being There' (SBBT) was developed at Oregon State University [2].

The Department of Electrical & Computer Engineering at Utah State University offers a graduate-level course in Distributed Control Systems [3]. This course deals with the design, implementation and stability issues in networked control systems, wireless sensor networks, and distributed parameter systems. The department also offers a course in Mechatronics [4], which deals with the principles, interfacing, and signal-conditioning of motion sensors and actuators, the modelling, analysis and identification of discrete-time systems and digital controller design methods.

Note that work described in this paper is an extension of that previously submitted for presentation at a conference [5]. Here we present more technical details, including a discussion of how the system is used in courses.

PLANT BACKGROUND INFORMATION

The Center for Self-Organizing and Intelligent Systems (CSOIS) at Utah State University has designed and developed several prototype robotic vehicles based on a key enabling concept called the 'Smart Wheel'. It is a self-contained robotic wheel module with three independent axes, namely the steering axis, the drive axis and the z-axis. When multiple smart wheels are attached to a chassis, the resulting vehicle is called an omni-directional vehicle, which is capable of independent orientation and motion, and can even climb stairs [6].

CSOIS has a stand-alone smart-wheel assembly (shown in Fig. 1) equipped with steering and drive motors, a linear actuator for z-axis movement, drive circuitry for the motors and actuator, encoders for drive and steering feedback, a microcontroller and a power distribution unit.

The smart-wheel assembly was modified and augmented for use in the two courses mentioned.

---

* Accepted 5 November 2007.

Fig. 1. Smart-wheel assembly on mobile rig.

The intended outcome was to enable students to design and test their own controllers for the steering and drive axes of the wheel, from any location with Internet access. The assembly also serves as a demonstration system [7] to showcase the capabilities of the smart wheel to visitors in situ and to visitors of the CSOIS website. In addition, the smart-wheel assembly also serves as a research test-bed for faculty members and graduate students.

SYSTEM OVERVIEW

Networked control systems: an introduction

A networked control system is a feedback control system in which the control loop is closed through a communication network [8]. Such a system lends itself naturally to remote experimentation with the plant and feedback sensor situated inside the university's research centre, the controller located on the student's remote computer and the communication network being the IP network between the plant/sensor and the controller.

The objective of any networked control system is to use the finite network capacity while maintaining good closed-loop performance, including stability, rise time, overshoot and other design criteria [8]. There are two approaches most commonly-used in the design of networked control systems [9]. One approach involves considering network-induced effects in the controller design for a given plant, set of design criteria and network protocol definition. The other approach involves making modifications to the network protocol for a given plant and controller (in this case, the controller is designed in advance without considering network-induced effects). The course laboratories described in this paper follow the first approach.

Distributed control systems design laboratory

In this course, the student is expected to design, test and demonstrate a working networked control system for the steering axis of the smart wheel.

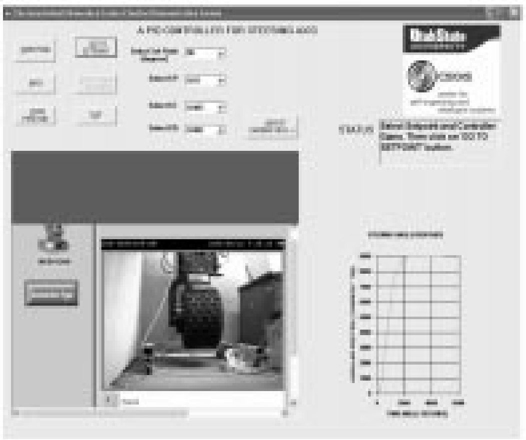

Fig. 2. Screen-shot of the prototype networked control system application.

The serial port (RS232) on the smart wheel's embedded controller was connected to an Ethernet Hub via a Serial Server, which translates messages between Ethernet and RS232 formats. The microcontroller on the wheel was programmed to poll the position encoder and transmit these values through its serial port. It also accepts velocity values via the serial port and converts them into pulse width modulated (PWM) signals to drive the motor.

The student with the Internet-enabled computer at any remote location will install a virtual COM driver, in order to communicate with the smart wheel's serial server. He/she can develop his/her control algorithm on the computer, which will transmit velocity values to the smart wheel, calculated based on the encoder readings received from the smart wheel. The Internet introduces random time delays and perturbations. These network-induced effects need to be taken into consideration while designing the controller, through the use of suitable techniques such as network prediction and delay-compensation. The closed-loop performance of the control system can be examined, by plotting the encoder data on the remote computer. Any windows-based environment may be used for development.

Mechatronics design project laboratory

In this undergraduate-level course, the students will design and test simple PID networked controllers for the steering and drive axes on their Internet-enabled computers, but they will not be expected to consider network-induced effects in their design.

Figure 2 shows the user-interface for a prototype networked control system implemented for the steering axis of the smart wheel developed in Visual Basic. The set-point (steering angle) and controller gains (PID controller) are user-selectable and the user can view the system performance by observing the angular position plot that appears in real-time on the graphical display. An Internet camera streaming live video enables the student to

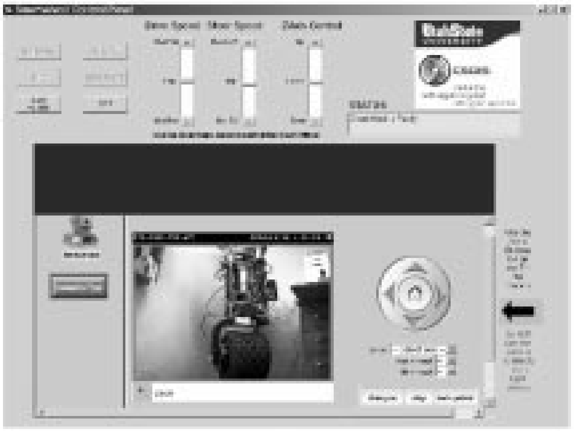

Fig. 3. Screen-shot of telepresence control demonstration system.

view the wheel motion in real-time on the control panel.

Demonstration system

When the wheel is not being used for the course laboratories, the capabilities of the wheel can be showcased to local visitors and remote visitors to the CSOIS website. In this case, the microcontroller accepts set-points for the steering, drive and z-axes from a pair of joysticks or the remote computer, and controls the wheel according to its on-board control program for each of the three axes. Figure 3 shows the user-interface for the smart wheel's telepresence control demonstration system.

To coordinate between the joystick, telepresence and networked control modes, and ensure predictable behaviour when multiple users attempt to access the system, suitable handshaking, control and arbitration protocols have been implemented.

SYSTEM DESCRIPTION

The architecture of the stand-alone smart-wheel demonstration system [7] is shown in Fig. 4.

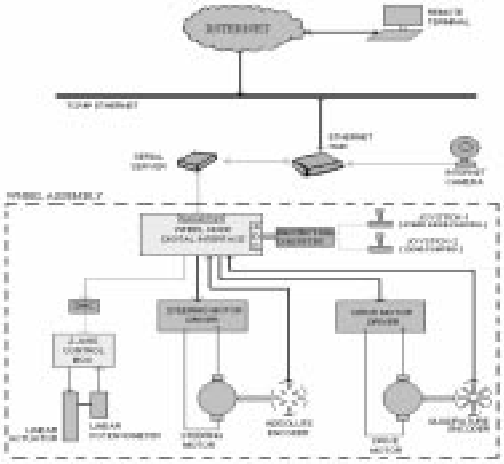

Fig. 4. Smart wheel system overview.

Overall plant description

The steering motor, drive motor, and linear actuator control the steering axis, drive axis, and z-axis of the smart wheel, respectively. The steering and drive motors are controlled by PWM and direction signals from their respective motor drivers. The steering motor is coupled to an absolute encoder, which measures the absolute angular position of the wheel. The drive motor is attached to a quadrature encoder that measures the relative motion of the wheel. The linear actuator moves the entire steering column up or down along the linear slide. The linear potentiometer provides feedback about the z-position of the steering column. The z-axis control box commands the linear actuator based on the voltage value from the digital-to-analogue converter (DAC) and the feedback from the linear potentiometer.

Mechanical hardware

A cross-sectional view of the smart-wheel assembly (without z-axis actuator features) is shown in Fig. 5. The steering suspension supports the entire steering mechanism and wheel electronic systems. The steering spindle is directly coupled to the steering motor. The yoke fastens the drive assembly to the steering spindle. The yoke also houses the power and communications cables, which go down to the wheel. A custom slip ring allows infinite rotation of the wheel about the steering axis. The entire drive assembly including the drive spindle, motor, failsafe brake, and encoder is enclosed inside the wheel shell. The drive motor is a Kollmorgen model QT-6407 frameless torque motor and the steering motor is a MicroMo series GNM 5440 PM DC motor. A CP-560 quadrature encoder from Computer Optical Products is used to measure relative wheel position. The absolute encoder used to measure absolute angular position of the steering motor in body-fixed coordinates is a Model 40H from Sequential Electronic Systems.

Z-axis actuation capability was added by mounting the assembly shown in Fig. 5 on an Electrak linear actuator from Warner Electric. A linear potentiometer is mounted adjacent to the linear actuator near the top. The power distribu-

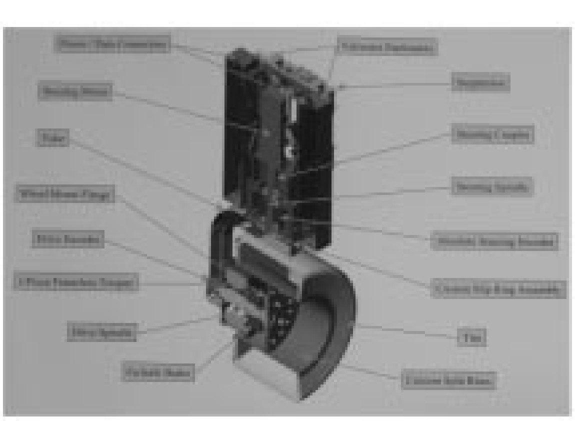

Fig. 5. Smart-wheel mechanical assembly.

tion units are enclosed behind the damper. A control box for the z-axis movement is mounted on one side of the power distribution unit housing. The wheel electronics and motor drivers are mounted on either side of the steering column. The entire assembly is mounted on a wheeled stand. Refer to Fig. 1 again to see a complete picture of the system.

Embedded control hardware

The wheel node digital interface board consists of the Tattletale Model 8 (TT8), a microcontroller by Onset Computer Corporation [10] based on Motorola's MC68332 microprocessor. It also provides interfaces for serial communication, motor drivers, D/A and A/D converters, absolute and quadrature encoders, circuits for power supply and regulation, level-shifting, optical isolation and the hardware watchdog unit.

The steering and drive motors are each driven by a 50A8DD Series PWM servo amplifier (or motor driver) from Advanced Motion Controls. The duty cycle of the pulse width modulated (PWM) signal determines the speed of the motor and the logic level of the direction signal determines the direction of rotation.

A pair of joysticks interfaced to the A/D converter of the TT8, enable visitors to control the 3-axes of the wheel locally. One of them (Joystick-1) is a 3-axis joystick wherein the x-axis is used to set the drive speed and the twist axis is used to set the steering speed. Another single-axis joystick (Joystick-2) is used to set the z-axis position.

The plant components together with the embedded control hardware comprise the smart-wheel assembly.

Telepresence control hardware

The smart-wheel assembly is connected to the IP network via a serial server and an Ethernet switch. The serial server includes the hardware and embedded software for converting messages between RS232 and Ethernet formats. The DE-311 serial server (Fig. 6) from Moxa Technologies was used in the system [11] because of its low-cost, multi-OS portability (for future use), compatibility with 10BaseT and 100BaseT Ethernet, and ease-of-use.

Fig. 6. DE-311 Serial server (Image courtesy Moxa Technologies).

A DCS5300 Internet camera from DLink (shown in Fig. 7) located near the smart-wheel assembly directly sends streaming video and audio [12] of the smart wheel motion to the remote operator independent of the controller data (which uses the serial server). Like all Internet cameras, it has a built-in web-server which captures and transmits live audio and video directly over the IP network, without the need for a direct connection to a PC. The camera's features which include live video (30 fps at 640*480 resolution) and audio streaming (via its built-in microphone), as well as pan, tilt and zoom control capability, and ease of access made it an attractive choice. This off-the-shelf solution though not intended for this purpose, provides an immersive experience for the remote student, and also enables him/her to monitor the proper functioning and safety of the plant.

However, because of the bandwidth intensive nature of the stream and the lack of priority control between the audio/video data and controller data, control system performance may be affected to varying extents depending on the remote computer's processing power.

Software description

The software architecture of the stand-alone smart-wheel demonstration system is shown in Fig. 8 and consists of the following components:

- Embedded software that resides on the TT8 microcontroller board.
- Video and audio buffering and streaming software on the Internet camera's web server.
- Embedded software on the serial server to convert messages between Ethernet and RS232 formats.

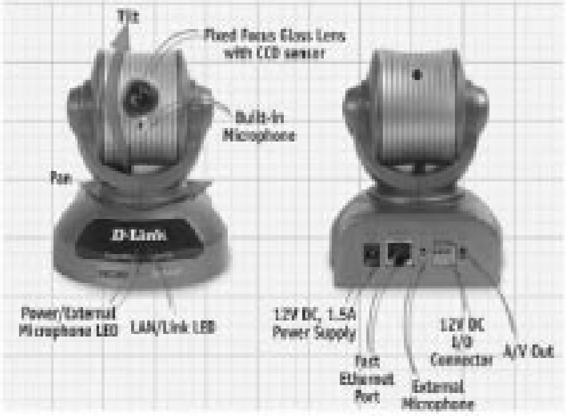

Fig. 7. DLink Securicam DCS 5300 internet camera (Image courtesy DLink Corporation).

- Virtual COM driver that allows communication between the student's remote computer and the serial server that instals on the student's remote computer.
- Application software for telepresence and networked control of the 3-axes of the smart wheel that runs on the student's remote computer.

Of the above, the Virtual COM driver (Moxa tech.), the serial server embedded software (Moxa tech.) and the Internet camera webserver software (DLink corp.), have been provided by their respective manufacturers. The embedded software on the smart wheel's TT8 microcontroller board was developed at CSOIS. For demonstration purposes, an integrated application with a telepresence controller and a prototype networked controller was developed at CSOIS.

The software on the TT8 microcontroller [7] runs the system in one of four modes: the idle mode, the joystick mode, the telepresence mode and the networked control mode. In idle mode, the system silently polls the joystick inputs and the serial ports for potential access requests. When a local user requests access (by pressing a button on the joystick panel), the system goes into joystick mode. In this mode, the on-board control algorithm commands the actuators based only on inputs from the joystick. When a remote user requests telepresence control access (by pressing a button on the GUI), the system transitions to telepresence mode. In this mode, the on-board control algorithm controls the wheel axes based on set-points received from the remote computer (by moving the 3-slider bars on the GUI). When a remote student requests networked laboratory access (by sending the appropriate access request command), the system goes into networked mode. In this mode, the on-board program sends sampled encoder readings along with the time-stamp and converts motor velocity values received from the remote computer into PWM signals for the motor driver hardware. The built-in arbitration mechanism sends appropriate rejection messages to remote requests when the smart wheel assembly is already being controlled by a local or another remote operator. For resolving multiple requests to access the plant, the access to the system is on first-come first-served (FCFS) basis, so as to reduce idling and increase resource utilization.

LABORATORY STRUCTURE AND SET-UP

Figure 9 depicts the laboratory setup for the networked control system design for the steering axis of the smart wheel. The project handout is shown in Appendix-A.

The student is required to install the virtual COM driver on his/her computer. This makes the smart wheel's serial server appear to the computer as just another serial port. The only requirement is that the computer must have an active Internet connection, which could be wired or wireless. The suggested minimum system configuration is a Windows 98 OS, Internet Explorer 5.x, Pentium-III 800 Mhz, 128Mb RAM, with a VGA Card 800*600.

The protocol and packet structure have been provided to the students as part of the laboratory handouts [13, 14]. The procedure is summarized as follows:

1. Remote terminal sends *ncs_request* command.
2. Smart wheel sends *ncs_accept* command (if it was in idle mode) and goes into networked mode and resets its on-board timer.
3. Smart wheel sends *ncs_reject* command if it was in already in joystick, telepresence or networked mode.
4. If remote terminal reads an *ncs_accept* command, it initializes its controller and gets ready for incoming encoder feedback data packets.
5. When smart wheel is in networked mode:
   (a) It gets feedback values from encoder device driver and packs it in the format *p<encoder>q* where <encoder> is any value between −180 and −180 degrees.
   (b) It gets timer values from the timer device driver and packs it in the format *t<time>u* where <time> represents time in milliseconds relative to the reset performed in step (2).
6. Smart wheel then sends the packed encoder value and corresponding time-stamp via the serial port.
7. Remote terminal reads the above values from

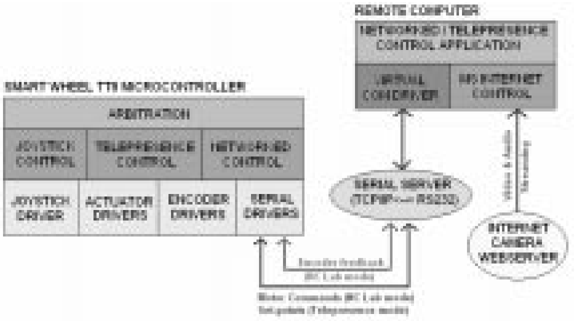

Fig. 8. Smart-wheel software architecture.

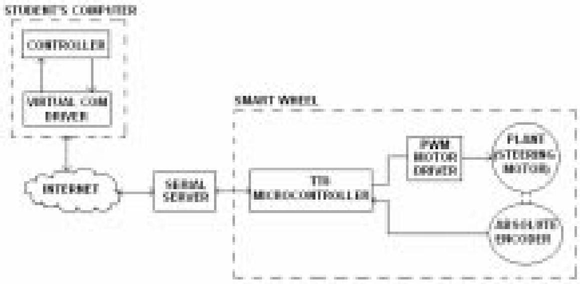

Fig. 9. Networked control system laboratory.

the virtual COM driver and unpacks the results and passes it to the control program.

8. The control program calculates a new velocity depending on the error and the networked control algorithm used.
9. The new velocity is mapped to an appropriate ASCII character as follows:
   (a) 70 to 100 for clockwise velocities (70—stopped, 90—maximum velocity).
   (b) 70 to 40 for counter-clockwise velocities (70—stopped, 50—maximum velocity).
10. The above value is transmitted to the smart wheel via the virtual COM driver and the serial server.
11. Smart wheel reads the remote velocity, converts it to a duty-cycle (-1.0 and 1.0) and transmits it to the motor driver via the on-board timer peripheral.
12. When the smart wheel receives an *ncs_complete* command from the remote terminal, it halts the motor, and goes back into idle mode.

An emergency stop mechanism enables the student to halt the wheel if the student notices any abnormal behaviour through the live video. the system enforces a time-out if there is no communication for a specified period of time. This ensures stability of the control loop and safety of equipment in case of a power outage or network failure.

The system response can be plotted on the student's computer for performance-evaluation purposes. This enables the motion of the wheel axis to be viewed in real-time during development and testing. Up to twenty users can simultaneously view the wheel's motion via the Internet camera's web-server.

Since the protocol and packet format are simple and use only keyboard characters, the student can use a terminal emulator (like HyperTerminal) to communicate with the smart wheel's embedded controller and get familiarized with the format of transmit and receive information, before getting started with controller design.

This setup will also be used in the ECE5320 Mechatronics course during Spring 2006 [14]. The students can use the same system to design and test simple PID controllers, but they need not consider network-induced effects in their design. One suggested benchmarking exercise for the students will be to compare the system responses for a given controller on different remote computers, different types of Internet connections and different locations.

LABORATORY BENEFITS

During the Spring 2005 offering of the Distributed Control Systems course at Utah State University, the students were able to successfully develop and test their networked control system designs from any wireless or wired internet-enabled computer and gain a better understanding of networked control-related design issues. We intend to extend the use of the smart wheel assembly for the Mechatronics course in the year 2006.

For much of the modelling, analysis, design and programming work, the students did not need access to the actual system. It was used only when they tested their design and iteratively modified it towards better closed-loop performance. Since all of this was done at the time and place of their convenience and the system was available for access round-the-clock, resource conflicts were minimal. There was no need to provide student-access to the research centre that houses the equipment and no contentions with the centre's other activities.

With only one equipment station, every student enrolled in the course was able to perform the laboratory assignment. The serial server enabled us to retrofit an existing plant without the need for a dedicated local computer that many of the expensive commercial solutions require. This helped the department control costs of laboratory equipment and maintenance for such a specialized subject without compromising on the quality of the course, thereby fulfilling an important objective of internet-based education [15].

LIMITATIONS AND FUTURE WORK

At present, the embedded controller transmits encoder values at fixed intervals of 10ms and the serial configuration has been set to 9600 bps, 8 data bits, no parity, 1 stop bit, and no flow control. This was done to enable low-speed computers to buffer and process the information without a system failure. A future upgrade with remotely-configurable parameters will enable high-end computers to take advantage of their processing power and obtain improved system response.

As of now, the video/audio streaming of the laboratory plant uses a commercial off-the-self (COTS) solution which was not intended for use in remote laboratories. Requirements for remote experimentation streaming solutions differ from those required for videoconferencing and other similar applications in several ways. These include the need to transmit the most recent information at the cost of discarding older data, emphasis on video over audio, etc. Also, since this is a bandwidth-intensive stream, viewing the video on the same remote computer as the networked controller may significantly degrade the performance of the controller, depending on the processing power of the computer used. The data for motor commands, encoder feedback, session control and audio/video streams have different QoS requirements. At present there is no control over the different types of data. When the relative priorities of the different streams are custom-managed, the controller performance degradation problem on account of the video/audio streaming can be resolved [16].

At present, the laboratory can only be run under the Windows environment. The student will be able to work on a Linux-based system as well, if we incorporate the custom-streaming solution (described above) and a TTY driver to communicate with the serial server (instead of the present virtual COM driver).

It is also proposed to extend the NCS laboratory to include velocity control of the drive axis of the smart wheel. This will enable students to implement and experience different scheduling approaches for the steering and drive axis control threads, and deal with issues concerning the polling and commanding of multiple sensors and actuators (multirate control).

CONCLUSION

This paper describes a cost-effective, self-contained and truly 'mobile' embedded remote learning solution. By adding a low-cost serial server, the stand-alone robotic wheel assembly could be used for demonstration purposes, research in networked control systems, as well as control systems education.

REFERENCES

1. D. Gillet, C. Salzmann, R. Longchamp, D. Bonvin, *Telepresence: An Opportunity to Develop Practical Experimentation in Automatic Control Education,* European Control Conference, (1997).
2. Burcin Aktan, Carisa A. Bohus, Lawrence A. Crowl, Molly H. Shor, *Distance Learning Applied to Control Engineering Laboratories*, IEEE Transactions on Education, (1996).
3. YangQuan Chen (January 2005). ECE 7750 Distributed Control Systems, Utah State University, Course website [Online]. Available http://mechatronics.ece.usu.edu/ece7750/
4. YangQuan Chen (January 2005). ECE 5320 Mechatronics, Utah State University, Course website [Online] http://mechatronics.ece.usu.edu/
5. Omni-directional Robotic Wheel—A Mobile Real-Time Control Systems Laboratory, Bharath Ramaswamy, YangQuan Chen, and Kevin L. Moore, *Interactive Paper* to appear in the 2006 American Control Conference, June 2006.
6. Nicholas Flann, Kevin Moore, A Six-wheeled Omnidirectional Autonomous Mobile Robot*, IEEE Control Systems*, 2000.
7. Bharath Ramaswamy, *Embedded and Telepresence Control of a 3-Axis Smart Wheel Assembly,* M.S. thesis, Dept. Electrical and Computer. Eng., Utah State Univ., Logan, Utah USA, (2004).
8. Octavian Beldiman, *Networked Control Systems,* Ph.D. Dissertation, Dept. Electrical and Computer. Eng., Duke Univ., Durham, NC USA, (2001).
9. Gragory Walsh, Linda G. Bushnell, Octavian Beldiman, Hong Ye, Caleb Belai, *Networked Control Systems*, Intelligent Control Engineering Laboratory, University of Maryland at College Park, USA, [Online] http://www.enme.umd.edu/ice_lab/ncs/ncs.html
10. Onset Computer Corporation, *Tattletale Model 8 Installation and Operation Manual* (1999).
11. *NPort DE-311 Serial Device Server Solutions Datasheet,* Moxa Technologies Co. Ltd., Taipei Taiwan.
12. *DLink Securicam DCS 5300 Internet Camera Datasheet,* DLink Systems Inc., Fountain Valley, CA USA, (2003).
13. *Networked Control Systems Design Laboratory, Project-1 for ECE7750 Distributed Control Systems,* Course Laboratory Handout, Dept. Elect. Comp. Eng., Utah State Univ., Logan USA, (2005).
14. *Networked Control Systems Design Laboratory, Project for ECE5320 Mechatronics,* Course Laboratory Handout, Dept. Elect. Comp. Eng., Utah State Univ., Logan USA, (2006).
15. Sandra E. Poindexter, Bonnie S. Heck, Using the Web in Your Courses: What Can You Do? What Should You Do?, *IEEE Control Systems* , (1999).
16. Ch. Salzmann, H. A. Latchman, D. Gillet, O. D. Crisalle, *Requirements for Real-Time Laboratory Experimentation Over the Internet,* International Conference on Engineering Education, (1998).

APPENDIX

PROJECT: Design and implement a networked control system for a stand-alone smart wheel assembly

Objective

To design and implement a networked control system to control the angular position of the steering axis of the smart wheel.

Introduction to the system

The smart wheel at CSOIS is a self-contained wheel module with separate drive and steering motors as well as a linear actuator, which enables the wheel to be moved along the z-axis. It is equipped with PWM amplifiers for the steering and drive motors and absolute and quadrature encoders to measure the steering and drive axes' positions respectively. A wheel node digital interface board accommodates a Motorola 68332-based Tattletale TT8 controller along with interfaces to the motor drivers, serial communication, absolute and quadrature encoders, circuits for power supply and regulation, level-shifting, and optical isolation, and the hardware watchdog unit.

For this project, only the steering axis will be controlled. The steering motor is controlled by feeding PWM and direction signals to its PWM amplifier. The steering motor is coupled to an absolute encoder, which measures the absolute angular position of the wheel.

You can communicate with the smart wheel through a device known as the serial server, which converts messages from RS232 format into TCP/IP format and vice-versa. You need to install a virtual COM driver on your development terminal in order to communicate with the smart wheel. This will make the serial server appear to your computer as just another serial port.

Command and communication protocol

Your controller will obtain access to the smart wheel upon sending the ncs_request (ASCII: 47) command. The smart wheel will respond with an ncs accept (ASCII: 33) command to indicate that it is ready to receive motor commands and an ncs_reject (ASCII: 35) command if it is busy. The steering motor rotates in one direction when it receives ASCII values 70 to 90 (70—Stopped, 90—Maximum Speed). It rotates in the opposite direction when it receives ASCII values 70 to 50 (70—Stopped, 50—Maximum Speed).

At approximately equal intervals, the TT8 controller samples the absolute encoder and transmits data to the remote terminal in the following format:

t<τ>u

p<θ>q

where:

<τ> represents the time (in ms) of sampling (relative to receipt of the *ncs_request* command).

<θ> represents the wheel orientation in degrees at time <τ>.

The wheel will halt rotation upon receipt of the ncs_complete command (ASCII: 92). Do use this command at the end of your control session and also, implement an Emergency Stop! To resume access, the ncs_request character has to be sent again.

It is a good idea to use a terminal emulator such as HyperTerminal to communicate with the smart wheel before beginning NCS development (Use this configuration: 9600 BPS, 8 data bits, No Parity, 1 Stop bit, No Flow control).

Motion video

An Internet camera located near the smart wheel assembly directly sends streaming video and audio of the smart wheel motion to the remote operator. This will enable you to view the motion of the wheel axis in real-time during development and testing. The video can be accessed at http://129.123.85.37/ Enter 'user1' as user-name and 'guest' as password when prompted. You may be asked to download a 'DLink Audio Control' before you obtain access to the camera's video.

Lab requirements

Implement a networked control system with a PID Controller to control the steering position of the smart wheel using the available hardware. You can use any software platform of your choice. Your software should plot the system response in real-time and preferably be a self-contained application. You must consider network-induced effects and use the techniques learned in the course in your design. Run the controller for different set-points and controller gains and evaluate the system performance.

On the due date (TBD), you will be required to demonstrate your system to the class. You should also turn-in a report which includes your design, software platform used, and controller-performance evaluation. You may work in groups of up to three members.

Before you begin

Download the client software installation files from the below location and follow all the instructions for installation of the Virtual COM Driver and the Networked Control Laboratory demonstration software

http://csois.usu.edu/people/smart wheel/CompleteInfoPage.htm

The laboratory handout for the ECE 5320 Mechatronics course will be similar to the above except that the students are not required to consider network-induced effects in the design. However, students are expected to run the controller for different set-points and controller gains and evaluate the system performance. They also need to compare the system responses for a given controller on remote computers with different hardware configurations, different types of Internet connections and from different locations.

Bharath Ramaswamy received his BE (Hons) and MSc (Hons) degrees from Birla Institute of Technology and Science, Pilani, India and his MS degree in electrical engineering from Utah State University, Logan UT, USA. Formerly, he was Research Assistant at the Center for Self-Organizing and Intelligent Systems, Department of Electrical and Computer Engineering, Utah State University. He is now an Engineer at Qualcomm Incorporated, San Diego, CA 92121 USA. His interests include mechatronics, control systems, real-time embedded systems and 3G wireless systems.

YangQuan Chen received the B.S. degree in industrial automation from the University of Science and Technology of Beijing, Beijing, China, in 1985, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, in 1989 and the Ph.D. degree in control and instrumentation from Nanyang Technological University, Singapore, in 1998. He is currently an Assistant Professor of electrical and computer engineering with Utah State University, Logan, and the Acting Director of the Center for Self-Organizing and Intelligent Systems. His current research interests include robust iterative learning and repetitive control, identification and control of distributed parameter systems with networked movable actuators and sensors, autonomous ground and aerial mobile robots, fractional order dynamic systems and control, computational intelligence, intelligent mechatronic systems, visual servoing/tracking, and most recently, distributed irrigation control and UAV-based remote sensing.

Kevin L. Moore received the B.S. degree in electrical engineering from Louisiana State University, Baton Rouge, in 1982, the M.S. degree from the University of Southern California, Los Angeles, in 1983, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 1989. He is currently the G. A. Dobelman Distinguished Chair and Professor of engineering with the Division of Engineering, Colorado School of Mines, Golden. From 2004 to 2005, he was a Senior Scientist with the Applied Physics Laboratory, Johns Hopkins University, Baltimore, MD, where he worked in the area of unattended air vehicles, cooperative control, and autonomous systems. He was an Associate Professor with Idaho State University, Pocatello, from 1989 to 1998, and a Professor of electrical and computer engineering with Utah State University, Logan, where he was the Director of the Center for Self-Organizing and Intelligent Systems from 1998 to 2004. He was a Member of the Technical Staff with Hughes Aircraft Company, Culver City, CA, for three years. His research interests include iterative learning control, autonomous systems and robotics, and applications of control to industrial and mechatronic systems.