Navigation Experiments with Mobile Outdoor Rovers via Internet

Qinghao Meng, Klaus Schilling and Rajesh Shankar Priya

Abstract-- Mobile robots offer interesting resources for inter-disciplinary engineering education, combing electronics, mechanics, software and control to a functional system. In the context of tele-education for engineering students, rover hard-ware is made accessible via internet such that experiments with real hardware can be performed remotely. This paper addresses the robot infrastructure, the tele-control and the remote sensor data acquisition methods based on JAVA servlets and applets, as well as the educational contents and experiences. Particular emphasis is on navigation techniques, allowing the robots to localize, to avoid obstacles, to control its movements and to reach given target points.

Index Terms-- Tele-education, mobile robots, navigation, sensors, remote control, telematics.

I. Introduction

Laboratory experiments are a special feature of engineering education, which resisted for quite some time to distance education. With the increasing capabilities of telematics (= *tele*communications + infor*matics*) at the end of the 90ies, it became feasible to remotely control equipment via the internet [2]. While industrial tele-service provision was the driving potential for this development [8], it offered also the internet techniques to enable remote control of laboratory experiments in the context of tele-education [1].

Our lab activities emphasize in particular the control of mobile robots, as a key component for

- the flexible flow of materials in industrial production [3], [4],
- for exploration of dangerouos environments [6], [9] and
- training with motivating equipment inter-disciplinary mechatronic system design in engineering education [7].

This paper addresses the use of outdoor rovers in engineering education. For this purpose in chapter 2 the mobile robot base is introduced, while chapter 3 addresses the telematics techniques for remote sensor data acquisition and control of this equipment via internet. Chapter 4 provides details on the content of the learning units and on the experiences of its use in class.

Informatik VII, Bayerische Julius-Maximilians Universität Würzburg, Am Hubland, D-97074 Würzburg, Germany schi@ieee.org

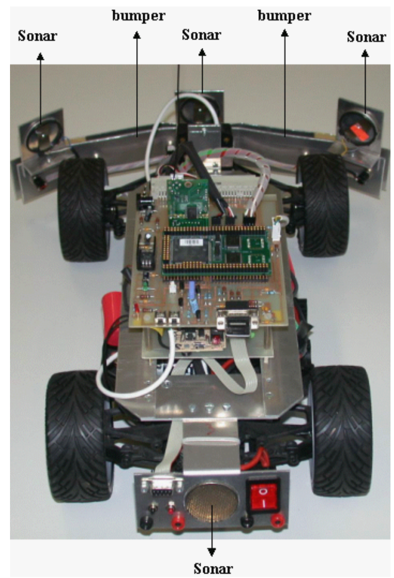

Fig. 1: The MERLIN rover from the rear side

II. The MERLIN Rovers

MERLIN (*M*obile *E*xperimental *R*obot for *L*ocomotion and *I*ntelligent *N*avigation) is a four wheel, car-like mobile robot (cf. Fig.1). It has been designed as vehicle for tests of sensors and mobility aspects in outdoor environments [5]. Due to this initial task, flexible integration of a broad range of sensor types was a key design criterion.

The vehicle is driven by the two rear wheels and steered by the two front wheels. Fig. 2 displays the electrical system and sensors of the robot MERLIN. Controls are realized by puls width modulation (PWM) signals from the micro-controller to the drive and steer motors, respectively. The data tranfer between MERLIN and a computer to record the test data has been realized by radio communication. One Radio Packet Controller (RPC) is mounted on the vehicle's main board; another is mounted on an interface board, which connects to the host PC via the RS232 serial port. The valid communication range for the RPC is 30 m indoors and 120 m outdoors.

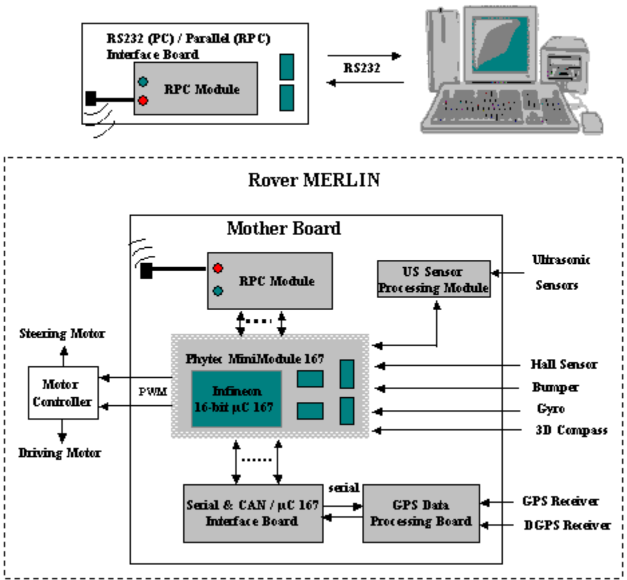

Fig.2 : The electrical system architecture of MERLIN

The mother board hosts the 16 bit microprocessor 80C167, which has the following key features:

- internal CAN-Controller
- 2 serial interfaces
- PWM signals for drive and steering motors
- A/D converter for gyroscope-data
- abundant interrupt system, 56 sources can cause interrupts

Two battery packs power Merlin. One 7.2V battery packs is dedicated to supply the drive and steer motors, another 12V battery pack feeds the micro-controller, the sensors and the other electronic components.

The MERLIN rover includes in its standard configuration a bumper, four ultrasonic ranging sensors, 3D compass, Gyro, and hall sensors as wheel encoders. The bumper is mounted in the front of the robot (cf. Fig.1), which together ultrasonic range sensors is used to avoid collisions. When the bumper reacts, then there is already contact to an obstacle, thus the related interrupt has for the microprocessor the highest priority and initiates a direct stop. Ultrasonic range sensors are applied to detect the distance between the robot and other objects. Three Polaroid ultrasonic ranging sensors are mounted on the bumper, another one in the rear of MERLIN (cf. Fig. 1). The reliably measurable range for these sensors is from 40 centimeters to 14 meters.

The 3D compass is a 3-axis orientation sensor capable of measuring ± 180 degrees of yaw heading, ± 180 degrees of pitch, and ± 70 degrees of roll (cf. Fig. 2), which provides MERLIN in outdoor environments the navigation data to determine its position and orientation. It is composed of a compass to measure the Earth's magnetic field and two inclinometer to detect pitch and roll angles. The 3D compass sends data through the second serial interface to the microcontroller 80C167.

The Gyro is capable of detecting of angular rate when the robot rotates. Thus the orientation of the robot is obtained by integrating the angular rate. The moved distance and the velocity of the wheels can be derived from two Hall sensors and eight small magnets placed in equi-distant way on the rim of each front wheel. The resolution for this odometry measurement is thus 4 cm.

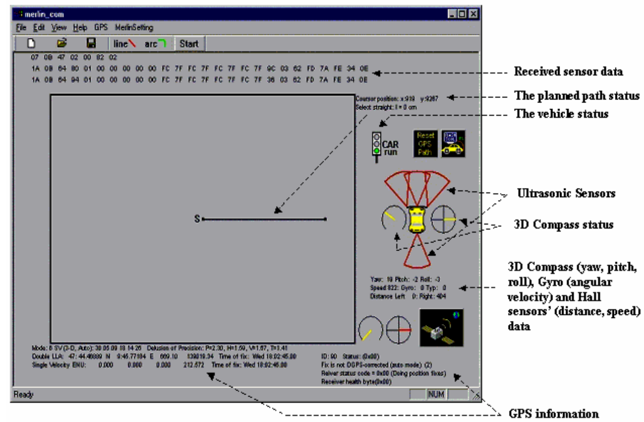

Fig.3 : User interface to control the robot's path and to display sensor measurements

III. THE TELEOPERATIONS INFRASTRUCTURE

Remote clients (students) can access the tele-experiments by using a standard web browser. The task is then executed by a local control system and the results are displayed on the client's browser. The system configuration is shown in Fig. 4.

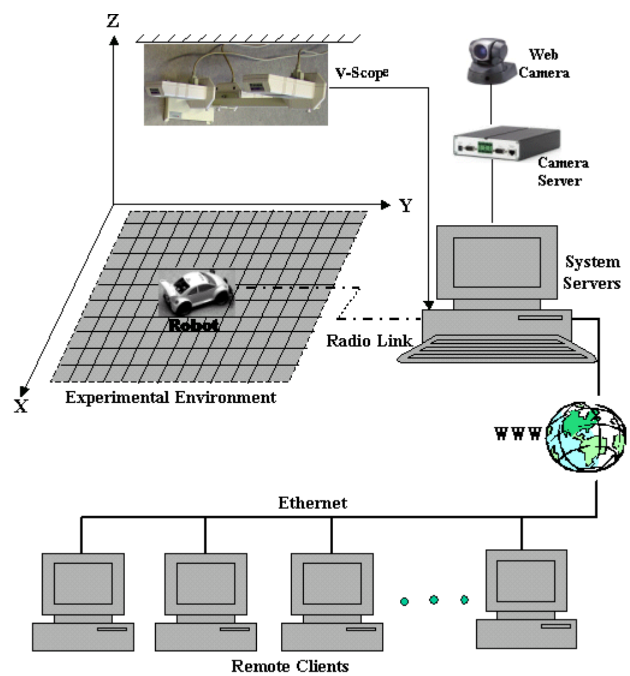

Fig. 4: The Internet based tele-control system configuration

The software system is combined from two parts: One is C-based program running in the micro-controller of the robot, which is responsible for the sensor data acquisition and processing as well as motors (drive and steering) control. The data and command exchange between the robot and the system servers is implemented by using radio link. The other part is Java-based program running mainly in the system servers. The software architecture of the system servers is based on a client/server model utilizing network sockets, where the servers reside on a machine hosting a robot and the client runs as a Java *Applet* in a user's web browser. The user interface for the tele-kinematics experiments is displayed in Fig. 3.

Through this interface, students can perform the kinematics experiment in the following steps:

- the robot is moved to the starting location;
- input steering angle and moving speed profiles;
- after the given moving time, the output of the experiment, including the calculated position and orientation, calculated steering angle and wheelbase, are displayed in the user interface;
- during the whole experiment, the image of the robot is live displayed in the user interface;
- graphs will be displayed to compare the position and orientation calculated by kinematics model and measured by the external motion tracking device V-Scope (or Stereo camera), respectively.

The data stream is controlled by

- the robot's server, the chat server and the html content stored in a Linux machine;
- a *Robot Backup Server*, used to replace the robot's server when it collapses because of equipment failure or hacker attacks, which runs in a Solaris machine;
- an *HTML/Chat Backup Server*, used to substitute the chat server and the html content in the Linux machine when they can not run probably.

The physical separation between the different standby servers makes the system more robust. The information channel Web Camera → Streaming Server ↔ (Internet) ↔ Remote Client provides the live images of the robot MERLIN. The streaming server transforms traditional analogue video into high quality digital images that can be transmitted over the Internet. In fact, the streaming server is hardware as well as software, it a standalone (not based on PC) device and can provide real time image transmission up to 25 frames/sec.

The software architecture is based on a client/server model utilizing network sockets, where the server resides on a machine hosting a robot and the client runs as an applet in a user's web browser. Java was chosen as the programming language of implementation because of its cross-platform characteristics, exception handling, dynamic loading capabilities, and standardized integration with web browsers. The system is built upon two fundamental classes called *ServerNegotiator* and *ClientNegotiator* that reside within the server and client applications, respectively. The main functions of these two classes are network communication and user management, which are the basic requirements for a tele-operated system. Network communication is used to notify developer's software of various important activities by posting events, including communication errors, connection status, robot availability, data availability and so on; user management is applied to coordinate multiple users who want to access the same tele-experiment, a "round robin" approach is used to arrange the access sequence and guarantee that at one time only one user has the right to occupy the experiment sources. Besides these two basic classes, other necessary classes are also developed to communicate with the robots.

Remote students operate the robot via a user interface, different experiment may have different control interface, and Java *Applets* are used to implement such interface.

With its cross-platform characteristics, exception handling, dynamic loading capabilities, and standardized integration with web browsers, Java was chosen as the programming language to implement the Internet-based obstacle avoidance experiment. A client/server model utilizing network sockets is implemented, where the server resides on a machine hosting a robot and the client runs as an applet in a user's web browser.

A special feature is that the remote users can generate C code for specific programme segments which can be transferred to the server, cross-compiled and downloaded to the rover's microprocessor for execution. Thus a remote student can test his software on real rover hardware.

IV. The Laboratory Experiments

The educational unit consists of specific learning modules:

(1) Background knowledge introduction, providing tutorials related to the experiment topic

(2) Interactive questions, providing the student feedback on his progress in learning

(3) Hands-on experiment

For this experiment, the background knowledge include:

- The concepts of holonomic and non-holonomic kinematics constraints
- The concept of dead-reckoning
- Forward kinematics model of the 4-wheel drive car-like mobile robot MERLIN
- Forward kinematics model of the differential driven mobile robot

. The educational objectives pursued with the particular use of the MERLIN robot are as follows:

1) Familiarization with remote code compilation and download onto MERLIN C167 microcontroller,

2) Kinematics Model
    - To identify the wheelbase of robot MERLIN
    - To identify steering angle for the mobile robot MERLIN

- To determine the position and orientation of the robot MERLIN by the kinematics model, and compare the results with the exact ones achieved by V-Scope or Stereo Camera.

3) Dynamics Model
    - To identify the mass of robot MERLIN
    - To identify location of CG (center gravity) of robot MERLIN
4) Motor Controller
    - Constant linear velocity controller design
    - Constant angular velocity controller design
5) Study of obstacle avoidance theory and application
6) Study of localization and docking theory and application

The experiment is performed outdoors in area with obstacles. The sensor information obtained by the robot is transmitted to the system server via radio modems (two radio package controllers provide this wireless link) and sent from the server to remote clients via the Internet. The remote students send in the reverse direction control commands to the robot through the same information channel.

At the begin of the experiment, the robot is guided to the initial pose (position and orientation) by the remote client through the user interface. The robot can move with different trajectories (straight lines or curves) by adjusting the steering angle in the user interface. The moving speed of the robot can also be controlled from the user interface. After a fixed period of time, which is set via the user interface, the robot will stop. The two poses, one calculated by the kinematics model, and the other one measured by the V-Scope during the experiment, are displayed on the same graphics, from which students can compare the calculated, the measured and the real data. By changing steering angle and/or moving speed, different results can be achieved.

V. Conclusions

The MERLIN rovers provide an interesting infrastructure for experiments in mechatronics due to the modular electrical structure and the flexibility with respect to sensor configurations due to the CAN bus interface. These properties are used in the design of tele-education experiments with real, remote hardware by integrating MERLIN into a JAVA based software for distributed systems and remote control developed for tele-servicing purposes. Tele-experiments targeted to navigation and control experiments are implemented, allowing students to access these experiments, more flexible with respect to temporal and local constraints, via Internet.

Acknowledgements

This research has been performed as part of the “LearNet”-project, supported within the program “New Media for Education” of the German ministry of Education and Research.

References

[1] Dormido, S. (ed.), *Proceedings IFAC Workshop on Internet Based Control Education*, Madrid 2001

[2] Goldberg, Siegwart (eds.). Beyond Webcams: An Introduction to On-line Robots, *MIT Press 2001*.

[3] Halme, A. M. Salichs (eds.). Intelligent Autonomous Vehicles, *Pergamon Press* 1998.

[4] Schilling, K., M. Mellado, P. Lopez, J. Remm, Free Navigating Autonomous Transport Robots for Flexible Industrial Production, in : Mårtensson, Mackay, Björgvinsson (eds.), Changing the Ways We Work, *IOS Press 1998*, p. 674 -684.

[5] Schilling, K., Q. Meng, “The MERLIN vehicles for outdoor applications”, *SPIE conference proceedings "Unmanned Ground Vehicle Technology IV"*, Orlando 2002, p. 43 – 49.

[6] Schilling, K., L. Richter, M. Bernasconi, C. Garcia-Marirrodriga The European Development of a Small Planetary Mobile Vehicle *Space Technology* 17 (1998), p. 151 - 162.

[7] Schilling, K., H. Roth, O. Rösch. Mobile Mini-Robots for Engineering Education. Global Journal of Engineering Education, 6, 2002, p.79 - 84.

[8] Schilling, K. H. Roth (eds.). Telematics Applications in Automation and Robotics - *Proceedings IFAC Conference 2001, Pergamon / Elsevier Science 2001*.

[9] Schilling, K., H. Roth, Teleoperated Inspection Robots for Space and Earth Applications, in : G.R. Gerhart, B. Abbott (eds.), Robotic and Semi-Robotic Ground Vehicle Technology, *SPIE Volume 3366* (1998), p. 74 - 82.