An Adaptable Outdoor Robotic Platform: Architecture, Communications, and Control

Matt Briggs, David Baird, Wade Ogg, Stian Aafloy, Aly I. El-Osery, Kevin Wedeward
Department of Electrical Engineering
New Mexico Institute of Mining and Technology
Socorro, NM 87801
isrg@ee.nmt.edu

Abstract – This paper presents a robust, easily upgradeable robotic platform for research along with its accompanying low-bandwidth and long-distance communication system. The proposed solution is a man-portable platform with easily accessible electronics bays and a communication system that uses free software, the Internet, and a cellular network. The design is made with the goal of developing multiple robotic platforms for cooperative robotic research.

Keywords: Robotics, web-based user interface, control and communication architecture.

1 Introduction

The promise of increased human safety has motivated the use of mobile robots in military applications ranging from surveillance and monitoring to assisting soldiers on the battlefield [1-3]. As new innovations in technology are adopted, the capability and performance of these robots improves at the cost of increased system complexity. Even though many issues in robotics such as kinematics, dynamics, and control of manipulators are well-established new challenges have arisen recently related to autonomy and intelligence of mobile robots, uncertainty in environments, and complexity in coordinating robot-to-robot interaction and cooperation. As solutions to these problems are developed an adaptable robotic platform is needed to serve as a test bed for implementation of both new hardware and software components.

Several robotic platforms have been developed in the past with off-the-shelf components [4-6] aimed at providing a robust platform for various applications. Unlike those robots, the platform presented here was mainly developed as a robust, yet flexible outdoor platform with a variety of modular components that include an easily changeable and accessible electronics bay. The designed system has a web-based interface that is operating system independent, free and open-source software, on-board navigation sensors, remote control, and communication redundancy.

This paper is organized as follows. In Section 2, the overall architecture of the robotic platform is provided. Section 3 describes the operator interface followed by a discussion of the server in Section 4. The communication framework is explained in Section 5 and Section 6 presents the robot hardware and control system. Finally, the conclusion is given in Section 7.

2 Overall Architecture

A view of overall system architecture is given in Figure 1. Each of the robot platforms consist of the following systems:

1. Operator station with computer, web-browser user interface, and ability to access and cache aerial photographs.
2. Server comprised of web server and database for queuing commands and logging system data.
3. Connection for communication via 802.11 wireless and/or a cellular network.
4. Robot composed of chassis, sensors, electronics, software, and control algorithms.

Details of these systems will be provided in the following sections.

3 Operator User Interface

One of the major challenges of this project was to develop an easy to use graphical user interface (GUI) that was suited for many to many relationships between robots and operators. One of the most obvious and perhaps the best known existing GUI for this relationship is the GUI used everyday on the Internet. This interface provides portability on both client and server sides of the equation. One of the strongest features of the Internet is its ability to handle many users at once while keeping the computational cost at a minimum. Another selling point for the Internet based solution is very low learning curve for operators.

By incorporating the internet into the system an operator from anywhere in the world will be able to log on to the robot and control it with a keyboard and mouse. Additionally, the user can run a variety of operating systems as long as they support a web standard compliant browser such as Mozilla Firefox.
One of the design challenges was to create a user interface that responded accurately to the user despite the robot’s heavily constrained bandwidth. To facilitate the above, the conventional joystick was replaced with keyboard controls. This was done to insure the operator had adequate manual control of the robot over a low bandwidth cellular connection.

A two mode waypoint based control system was also implemented to handle the low bandwidth connection. In first mode the user sets desired waypoints on a satellite photo/map, see Figure 2, to direct the robot to the desired location. In the second mode the user sets waypoints, on a grid divided into one meter squares, relative to the robot’s current position.

These two modes are also made necessary by the fact that the robot will not always have GPS lock or that GPS localization is not accurate enough for the tasks to be performed.

If the robot is unable to proceed, an image of what the robot is observing is sent over the cell network to the operator for analysis at the same time the robot requests assistance (Figure 3).
4 Server

On the server side, the GUI is supported by the web framework known as Ruby on Rails which provides the user with all the features that a modern database backed website has to offer. Just to note all software used in the GUI barring a few small Linux utilities can be run on Microsoft Windows. One of the features provided by Rails is known as scaffolding for the database where the user at anytime can look at and modify the data which resides in the database. This allows many interesting options that classic standalone application software usually does not provide like the ability to add custom commands or to look at the data in reasonably low level. This component was also essential during the development of the project where no ‘real’ data was available.

The database the web framework used is implemented using SQL schema specifically using SQLite. This standardized database allows the high level GUI to interact with any number of programs querying the database. The programs can be written in any number of languages including Python, C/C++, Ruby, and Java. This flexibility was very desirable as far as future projects are concerned. Another feature of SQL that is heavily utilized is the concept of transaction based data insertions where the data is not inserted into the database until all of the data for the given transaction is collected. This means that the database does not contain any ‘orphaned’ or incomplete data logs.

This database also has many future-looking qualities and research possibilities. One of these that has been considered extensively is the ability for programs to data mine the database and use the collected data for many modeling and simulation algorithms. Taking this data mining idea even further perhaps having an adaptive learning program that can look at the data from multiple robots and learn from both the robots mistakes and the operators’ corrections.

5 Communication System

Figure 4 shows the basic concept for the communications framework of the project. It was necessary to create a communication system that is far-reaching and low-cost and permitted the robot to not be connected all of the time. The framework selected is composed of two types of networks, 802.11b and the cellular network, which connect to the servers over the internet. OpenVPN is used to setup encrypted tunnels for all data sent across the internet, although OpenSSH can also be used. The robot can connect to the internet whenever it needs to check for new commands. The data collected by the robot and status information can be downloaded to the remote server whenever the robot connects.

There are three protocols that are tunneled over the encrypted tunnel: SSH for low-level control and debugging of the robot, ZeroC Ice (Internet Communications Engine) for sending messages between programs, and a custom designed protocol built on top of sockets. The custom protocol is designed to handle network disconnections and latencies. The custom protocol can tolerate sending the same message multiple times and acknowledging it multiple times, which will be required if a disconnection occurs. This protocol is currently the main one used during the robot’s operation.

There are a few drawbacks of the current system. The first is that the server cannot call the robot. The robot is only able to call the server. The next drawback is that the custom sockets protocol is not very ideal for porting between multiple programming languages and operating system platforms. It might be possible to use ZeroC Ice to displace this protocol but for now, this possibility has not been explored.

6 Robot Platform and Control System

The platform used is based on a proven, man-portable, tracked robot named MATILDA and made by Mesa Robotics (see Figure 5a). This platform has seen service in both Afghanistan and Iraq, and divides into three pieces for transportation as shown in Figure 5b where each piece weighs about 20 pounds. An actual picture of the robot is shown in Figure 6.

The electronics, including the sensor suite, were built around a x86 100MHz Single Board Computer (SBC) running a distribution of OpenEmbedded GNU/Linux. Communication to the robot can occur in three ways: radio control that bypasses the SBC for failsafe local control, 802.11b wireless for command/data transfer, and cellular network service for command/data transfer. The navigation sensor suite consists of a bank of eight short-range ultrasonic sensors, compass with tilt-roll, a network camera,
SBAS-enabled GPS system, and encoders on each drive motor. A custom signal and power interface board with a Field Programmable Gate Array (FPGA) is used to provide regulated power and process signals between the SBC, sensors, motor drivers, and communication equipment. Full optical isolation was implemented on all lines between the interface board in the center section and track sections. Power is provided by three Lithium Polymer batteries each of which resides in one of the three robot sections. Run time of over three hours is achievable and easily expandable to six hours with the addition of three more batteries. Due to the batteries’ intolerance of low-voltage operation, disconnect boards were implemented to warn operators and disengage batteries when low-voltage conditions exist. The computer and its relevant electronics are housed in a modular, easily accessible, weather tight container strapped down in the payload section of the platform.

The following is a list of specific parts used and their key features.

**List of parts and key features**

<table>
<thead>
<tr>
<th>Part Description</th>
<th>Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thales DG16 GPS receiver</td>
<td>20Hz data rate, 3m CEP</td>
</tr>
<tr>
<td>Diamond Systems Prometheus</td>
<td>PC/104 form factor, Ethernet, data acquisition, embedded 486-100MHz CPU</td>
</tr>
<tr>
<td>Thunder Power &quot;ProLite&quot; lithium polymer battery packs</td>
<td>4 cells, 14.8V, 8Ah</td>
</tr>
<tr>
<td>Hitec Supreme 3500 receiver and Laser 4 radio control system</td>
<td></td>
</tr>
<tr>
<td>Honeywell HMR3000 digital compass module</td>
<td>heading, pitch and roll output, 0.5° accuracy</td>
</tr>
<tr>
<td>IFI Robotics Victor 885 motor controller</td>
<td></td>
</tr>
<tr>
<td>Nova Engineering Constellation-10K FPGA development board</td>
<td>PC/104 form factor</td>
</tr>
<tr>
<td>Devantech SRF04 ultrasonic ranger</td>
<td></td>
</tr>
<tr>
<td>Maxon 24V DC motor with 43:1 planetary gearhead and digital encoder</td>
<td></td>
</tr>
</tbody>
</table>

Through utilization of the GUI the user can adequately control the robot in order to complete the desired goals. As discussed above the chosen paradigm is to issue very high-level commands that provide the robot GPS locations through which to traverse. Thus, the real intelligence must lie on the robot rather than the more computationally-able base station. Several algorithms were designed to handle the high-level and low-level control of the robot. The interconnections of the different system components are shown in the five layers of Figure 7. Details of each layer follow.

The top level is physical hardware and the next level is their associated drivers. The third layer is a data acquisition system that saves all the data into a circular buffer. The fourth layer consists of real-time processes that execute feedback control loops and gather data to put into the data acquisition system. The control process is the most sophisticated one in this layer and provides basic control of motion, including track velocity and position control. It also has some safety checks such as using the ultrasonic sensors to detect possible collisions, and checking that the motors have a reasonable velocity for a given PWM signal (i.e., to check if the robot is moving).
The final layer is non-realtime processes, and the network manager handles connecting and disconnecting to various networks. OpenVPN is used to establish encrypted tunnels across the networks and an OpenEmbedded GNU/Linux distribution with the Real-Time Application Interface (RTAI) is the SBC operating system. It automatically builds cross compilers and includes a large number of packages that can be cross compiled. In addition, to expedite the development cycle the high-level language Python was utilized. This creates an object-oriented environment which is quite easy to script.

7 Conclusions

Designing a robust and versatile robotic platform for outdoor applications is extremely complex. The platform will constitute a system of systems in which the integration among them has to be carefully designed. Additionally, due to the massive amount of data collected from the various components and sensors, data acquisition and processing has to be maintained in an orderly fashion to prevent race conditions which will occur if old information is being used during an update phase.

At the Electrical Engineering Department at New Mexico Tech, we have developed a highly robust and versatile platform that can be used for outdoor applications. The platform was designed with research in mind, so modularity and modification capability were a goal. The entire system as been developed, implemented on two robots and tested, and is currently ready for the implementation of high-level intelligent control to perform coordinated behavior research.

8 Acknowledgement

Development of the robotic platform was funded by the U.S. Special Operations Command and U.S. Army Soldier Systems Center-Natick under contract W911QY-04-C-0014.

References


