Control System Architecture for a Remotely Operated Unmanned Land Vehicle

Sandor Szabo, Harry A. Scott, Karl N. Murphy and Steven A. Legowik

Systems Integration Group
Robot Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract

The U.S. Army Laboratory Command, as part of its Robotics Initiative, is developing a testbed for cooperative, real-time control of unmanned land vehicles. The system requires the development and integration of many elements which allow the vehicles to perform both autonomous and teleoperated functions. The National Institute of Standards and Technology is supporting this program by developing the vehicle control system using an architecture based on the Real-time Control System (RCS). RCS is a hierarchical, sensory-based control system, initially developed for the control of industrial robots and automated manufacturing systems. In this application, RCS controls all vehicle mobility functions, coordinates the operations of the other subsystems on the vehicle, and communicates between the vehicle and the remote operator control station. This paper reviews the overall control system architecture and the design of the mobility and communication functions.

Introduction

Artificial intelligence, control technology and vision processing capabilities have not reached the point where robotic land vehicles can function autonomously in off-road tactical situations. Therefore, the vehicle control systems must support a mix of capabilities ranging from master-slave teleoperation to autonomous control. The development of such a control system is underway as part of a program initiated by the U.S. Army Laboratory Command. This program, called Techbase Enhancements for Autonomous Machines (TEAM), is a joint effort between several U.S. Army organizations, national laboratories, and commercial contractors. The goal of the program is to develop and demonstrate technology for use on tactical vehicles in the areas of mobility, sensory processing, target detection and tracking, and communications.

In the TEAM scenario, humans remotely operate two Robotic Combat Vehicles (RCVs) from an Operator Control Unit (OCU). Each vehicle contains a remote driving package, an inertial navigation system (INS), a mission package which performs target detection, tracking, and laser designation, and data and video communication links. The OCU contains controls and displays for route planning, driving, operation of the mission package, and control of the communication links.

A typical mission includes a planning phase where the operator plans a route using a digital terrain data base. The operator then drives the vehicle to a desired location and activates the mission package for automatic target detection. When a target is detected, the mission package designates the target with a laser. Afterwards, the vehicle retraces the path back to a previously known location, a process termed as retro-traverse.

The TEAM control architecture follows the Real-time Control System (RCS) architecture developed at NIST [1-3]. The application of RCS technology to the TEAM program was presented in Reference [4]. The RCS architecture is a hierarchy of control modules in which each module controls one or more modules at the next lower level. High level commands are decomposed into simpler commands as they pass down through the levels. Each control module is composed of task decomposition, world modeling, and sensor processing. Task decomposition implements real-time planning, control, and monitoring functions. It decomposes tasks both spatially and temporally. The sensory processing modules detect, filter, and correlate sensory information. The world modeling modules estimate the state of the external world and make predictions and evaluations based on these estimates.

This paper presents the overall TEAM control system. The first section describes the functional modules of the control system and their responsibilities. The emphasis is placed on the modules that support mobility functions. Next, the design sections of this paper present the details of the mobility and the communication subsystems. Finally, the

---

This work is funded by the U.S. Army Laboratory Command. This paper was prepared by U.S. Government employees and is not subject to copyright. Equipment listings do not imply a recommendation by NIST.

implementation section describes the current state of the mobility and communication control systems.

Control Architecture

The RCS architecture for TEAM is a logical tree structure that contains several levels, each with one or more control modules. Figure 1 shows the control modules that reside on the RCVs and Figure 2 shows the modules in the OCU.

The highest control level on each vehicle is the RCV Supervisor. It coordinates three subsystems: mobility, mission package, and communication. The mobility subsystem is responsible for all modes of driving and for providing sensory information back to the operators in the OCU. At the Servo level, the Vehicle Platform Controller drives independent actuators that position the steering wheel, accelerator pedal, brake pedal and other linkages. The Primitive (PRIM) level Mobility Platform Controller generates smooth trajectories in a convenient coordinate frame based on goals from the operator or from predefined paths. The Elemental (EMOVE) level Mobility Subsystem Controller is responsible for computing motion pathways that are clear of obstacles. The current goals for TEAM do not include automatic generation of obstacle free paths, however future developments in this area would occur at this level.

The automatic target acquisition (ATA) control modules in the mission package subsystem automatically detect, locate and laser designate moving tanks and other targets [5]. The ATA functions are divided among control modules for processing imaging sensors, infrared detectors, and nonimaging sensors, as well as modules for controlling leveling and aiming platforms.

The communications subsystem controls data and video communications between the RCVs and the OCU. The data communication modules provide data paths between modules on the vehicles and the OCU using a single radio link. The video modules utilize compression techniques to reduce the radio link bandwidth requirements while maintaining sufficient feedback to the operators for teleoperation.

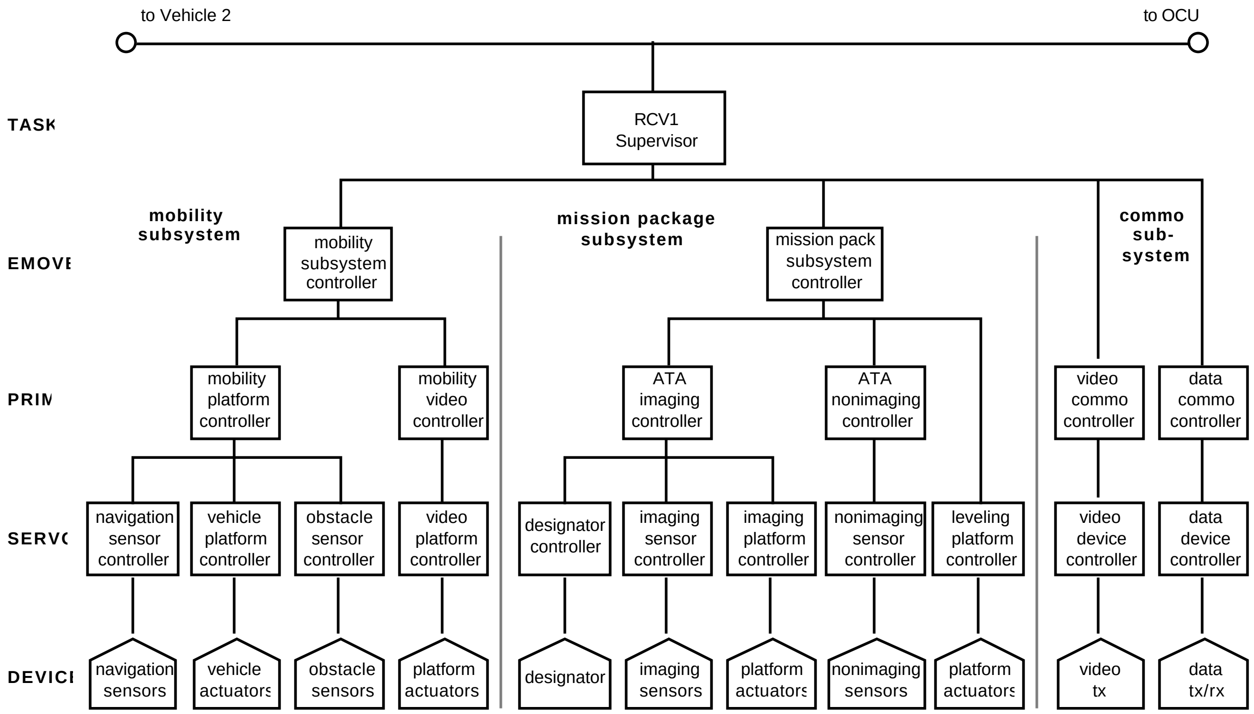

Figure 1. Control system architecture for TEAM Robotic Combat Vehicles.

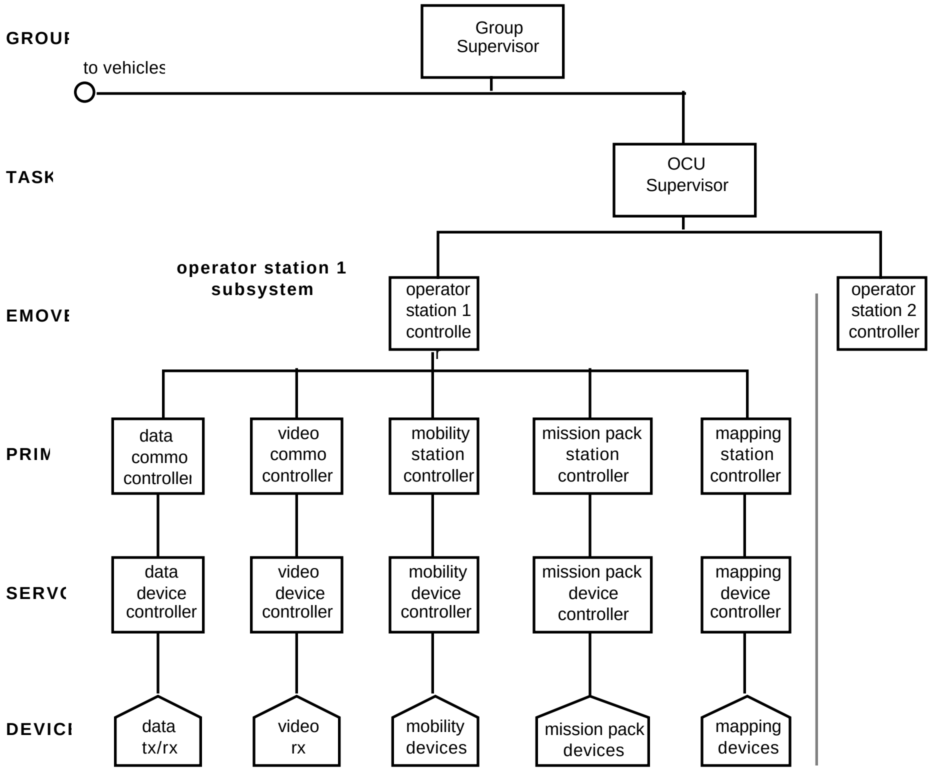

Figure 2. Control system architecture for Operator Control Unit.

The highest level control module in the TEAM architecture, the Group Supervisor, resides in the OCU. This module coordinates activities that involve multiple vehicles such as convoy deployment and cooperative target detection. Beneath the Group Supervisor is the OCU Supervisor which coordinates activities between the two operator station subsystems. Each operator station subsystem contains devices (joysticks, displays, video monitors, etc.) that allow the operator to drive the vehicle and to control the mission package. Included with each operator station subsystem is the communications equipment required to maintain video and data communication with the vehicle being controlled.

The development of the control architecture is an important step prior to performing the detailed design of the subsystems. Initially, the architecture is a tool for analyzing functions of the system and decomposing them into the levels. The architecture described has supported the detailed design efforts for the control modules discussed next.

Mobility Design Details

The TEAM program scenario requires the development of two modes of vehicle mobility: remote control and retro-traverse. In the remote control mode, the operator drives the vehicle from the OCU. In the retro-traverse mode, the vehicle automatically retraces a previously driven path based on data recorded from the inertial navigation system. The modules responsible for these driving modes are the Mobility Platform Controller on the vehicle and the Mobility Station Controller in the OCU.

In an RCS application, control modules use sensor processing (SP), world modeling (WM) and task decomposition (TD) to perform intelligent tasks. In the task decomposition module, the Job Assigner (JA) spatially decomposes tasks between planners. The planner (PL) decomposes tasks temporally in order to achieve goals within a planning horizon. While the planner looks ahead in time, the executor (EX) works in parallel with the planner to achieve immediate goals (For further details on these modules see Reference [3]). Figure 3 shows the roles of these processes within the mobility modules. The arrows between the OCU and the RCV represent information flowing between the two world models. This world model registration is supported by communication modules discussed in the next section. The breakdown of the mobility modules' responsibilities for the various driving modes is discussed next.

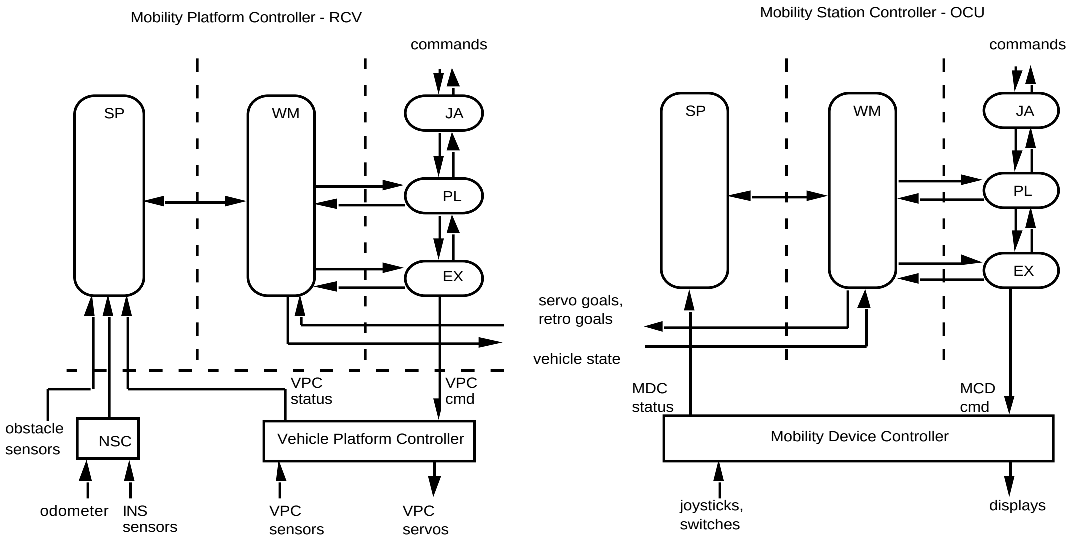

Figure 3. Mobility Prim Level: Command and data paths

Remote Control

In all driving modes, the vehicle Mobility Platform Controller (MPC) generates servo input commands for the Vehicle Platform Controller (VPC). Determining how the servo goals are generated is a task decomposition responsibility. During remote control, the world model contains the current state of the operator master devices (e.g., steering, brake, throttle, etc.). The MPC executor retrieves the current state of the devices and calculates servo goals. Certain reflex actions are also handled by the executor. For example, when the world model indicates that information from the operator devices is not available due to a communications error, or when an obstacle is detected while the vehicle is moving, the executor issues commands that bring the vehicle to a smooth halt. Various vehicle platform sensors provide information such as vehicle speed, engine temperature and engine rpm. The MPC sensor processor acquires this information and places it in the world model where it is accessible from the OCU.

In the OCU, the Mobility Station Controller (MSC) ensures that operator mobility commands are routed to the vehicle and vehicle state information is displayed to the operator. The MSC sensor processor acquires the current state of the operator's drive controls and places the data in the world model. The state of the vehicle is retrieved from the world model by the MSC executor and becomes part of a display data command issued to the appropriate display device.

Retro-traverse

The retro-traverse mode of driving gives the TEAM vehicles some degree of autonomy during a mission. Initially, a path (i.e., a set of INS data points) is recorded while the operator drives the vehicle. Then, at a later time, the vehicle retraces the path without the direct control of the operator. This is analogous to teach programming and playback commonly used for industrial robots.

The performance of the mobility control system is nearly identical in both the remote control mode and in the teach phase of retro-traverse. The main differences are that in the teach phase, the operator indicates when to record a path and the Mobility Platform Controller world model records the path based on navigation data acquired by the MPC sensor processor. The navigation data may be produced by several different types of systems, but the source is transparent at this level of sensor processing.

At some point in a mission, the operator initiates the autonomous drive phase of retro-traverse. If the vehicle is not facing in the proper direction, the vehicle must be maneuvered onto the path. One method being investigated to accomplish this is the use of an automatic path planner to generate simple trajectories. Once the vehicle is on the path, the MPC executor retrieves the path points from the world model. Each desired path point is compared with the current position and orientation of the vehicle obtained from the world model. A new steering command is generated that minimizes both lateral and angular deviations from the path. The operator can select a speed setting, as with an automobile cruise control. The MPC executor generates new throttle commands when the vehicle drifts from the desired speed setting. Brake commands may also be generated when the vehicle exceeds the desired speed setting.

The effectiveness of this mode of operation depends primarily on the repeatability of the navigation system. Retro-traverse relies on the operator to establish a safe route, which the vehicle retraces blindly. If the navigation data is not repeatable, the vehicle will deviate from the course. Several types of navigation systems are undergoing investigation. Each system has its advantages and disadvantages. A simple obstacle detection system will be incorporated as a safety mechanism.

Communication Design Details

Both video and data communication are required between the OCU and the vehicles. The transmission of video signals for driving requires high-bandwidth communications which is undesirable in tactical operation. Data compression techniques will be used to reduce the quantity of video data transmitted. Work in this area is presented in Reference [6]. For data communication, a single rf (radio frequency) link carries all information between the vehicle and OCU. Four categories of information are multiplexed on the single data channel. The categories are time-critical, emergency, time-noncritical, and development.

Time-critical data is transmitted frequently, often at a fixed rate, and between processes that cannot tolerate large transmission delays. This class includes servo goals used for driving (steer, brake, and throttle) and for manual control of the mission package platform. This class represents the most demanding communication occurring on the rf link.

Emergency messages are time-critical but are transmitted infrequently. These messages communicate emergency stop commands to the vehicle and emergency status from the vehicle.

Time-noncritical data is transmitted relatively infrequently and not with real-time urgency. Examples include file transfers, commands that set various vehicle modes, such as gear selection, and high level command and status exchanges.

The development data supports code development, downloading, process monitoring and debugging. This information may be exchanged at any time, including during vehicle operation.

Figure 4 illustrates how the communication functions are integrated into the TEAM architecture. The Data Device Controllers serve all RCV-OCU data communications. Any relevant portions of the vehicle and OCU world models may be transmitted over the rf link. Several vehicle and OCU control modules will require this type of communication. Shown here is an example of information flow during the remote control driving mode. Vehicle state information flows from the vehicle Mobility Platform Controller to the OCU Mobility Station Controller. A similar path is used, in the opposite direction, to convey vehicle goal states from the OCU to the vehicle.

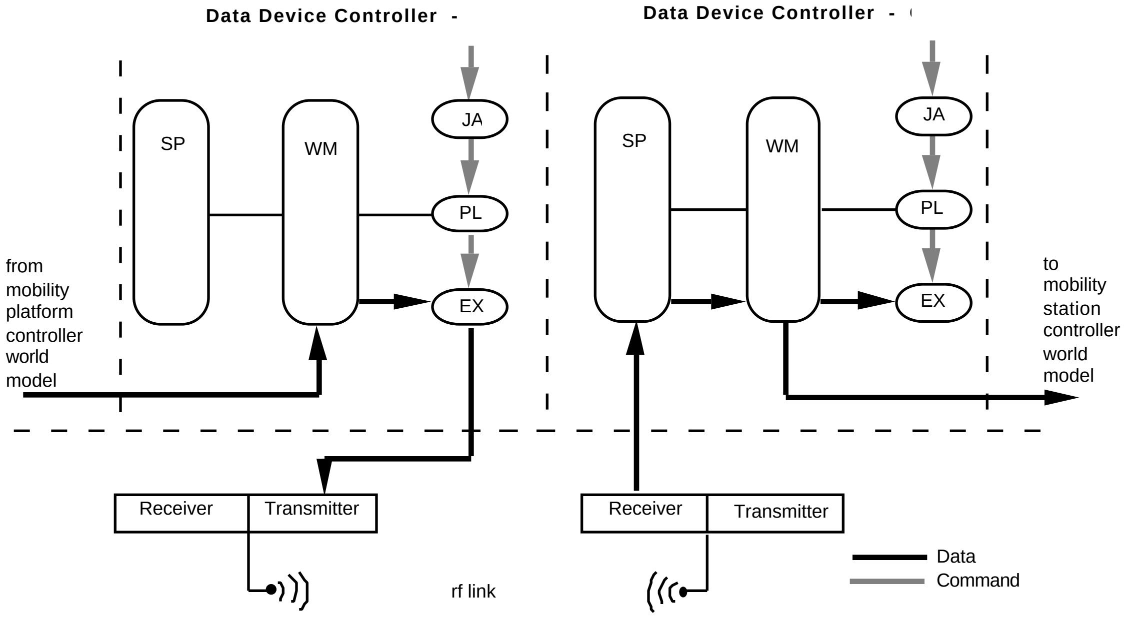

Figure 4. Data Communications - Vehicle to OCU.

When new vehicle state information is detected in the Mobility Platform Controller world model, it is passed to the vehicle Data Device Controller world model. An inter-process communications (IPC) mechanism, which will be discussed later, is used to facilate communication between processes. The executor obtains this data, prepares the appropriate message structure for it and executes the control functions required to transmit the rf message. It also monitors feedback from the radio that indicates its performance.

In the current design, the job assigner and planner establish the remote operation mode, which results in the information flow described above. In other hardware configurations the job assigner would distribute information over several separate radio equipment interfaces and the planner would select schemes to optimize communications.

The information transmitted from the vehicle radio is detected by the OCU receiver and retrieved by the OCU Data Device Controller sensor processor. The executor evaluates the reception and, as a result of certain error conditions, may report failures to the Data Communications Controller. The vehicle state information is then sent to the Mobility Station Controller world model via the IPC mechanism.

The inter-process communication mechanism is a set of utilities supporting mailbox type communication between processes in an RCS application. It provides a consistent user view for communications, while hiding the various underlying mechanisms employed. Process-to-process, board-to-board and subsystem-to-subsystem communication benefit from a standard communication interface. The underlying communication techniques include socket-based and common memory-based methods.

The IPC user can connect to a mailbox using a symbolic name, send variable length messages to a mailbox, and synchronize control processes through appropriate use of blocking reads. A read operation may be directed to block (with a timeout period) for synchronization, or may be non-blocking. Double buffering and internal semaphoring free the control processes from concern about overwriting a previous message. These capabilities provide a communication mechanism particularly well suited to an RCS environment and their ease of use encourages modularization of the system code.

Implementation

Several organizations are involved in the development of the TEAM control system. An important consideration in such an endeavor is integration of the individual subsystems that function in both the vehicle and the OCU. The system architecture is a useful tool for managing the software. At a system level, NIST has selected a standard computing environment that is commercially available. The environment utilizes a real-time UNIX-like operating system, VxWorks, on the target hardware and a separate UNIX system (Sun Microsystems) for software development. The environment hardware is based on Motorola CPUs interconnected by VME backplanes. Individual backplanes are typically connected using an ethernet local area network (LAN). NIST is developing the mobility and data communications subsections using this environment. An initial implementation supports remote control and retro-traverse mobility functions and contains the basic data radio communication functions.

The hardware configuration of the modules described in the previous sections appears in Figure 5. There are presently two processor boards on the vehicle. One contains the Mobility Platform Controller module and the Data Device Controller module. An RS-232 serial line connects the board to the Vehicle Platform Controller module. This module, developed by Kaman Sciences, provides servo control of steering, throttle, brake, and gear selection. The other processor board contains the INS Sensor Controller module. The proposed navigation system for the vehicles is a U.S. Army Modular Azimuth and Position System (MAPS). The MAPS uses accelerometers, an odometer and optical ring laser gyros to sense vehicle position and orientation. In addition, a Radio Frequency Navigation Grid (RFNG) is being used to provide vehicle position data during testing and development. The RFNG employs radio beacons placed at the corners of the test area which emit timing pulses received by the vehicle. Differences in phase information from each transmitter are used to compute vehicle position and heading. (For information on both MAPS and RFNG see Reference [7]).

A processor board in the OCU contains the Mobility Station Controller and the Data Device Controller. This board communicates with the Mobility Device Controller, currently a NIST developed portable unit that controls displays and reads the operator input devices.

As part of the computing environment, NIST is developing a standard communications system. The goal is to provide two types of communications capabilities. A subsystem may use a dedicated serial link that provides a point-to-point connection between CPU boards in the vehicle and in the OCU. Also, the ethernet LAN will be bridged between the vehicle and OCU. The communication system hardware is also shown in Figure 5. The Data Device Controllers in the vehicle and OCU exchange information through the serial multiplexers and rf transceivers. The rf radios selected employ a direct sequence spread spectrum modulation technology and operate at 915 MHz. The maximum rf data rate is 230 kilobits/sec. The multiplexer provides multiple 19.2 Kbaud asynchronous RS-232 ports for connection to the control system. One of the RS-232 ports will support the Serial Line Interface Protocol (SLIP) to bridge the LANs on the vehicle and the OCU. This is indicated by the multiplexer line labeled SLIP.

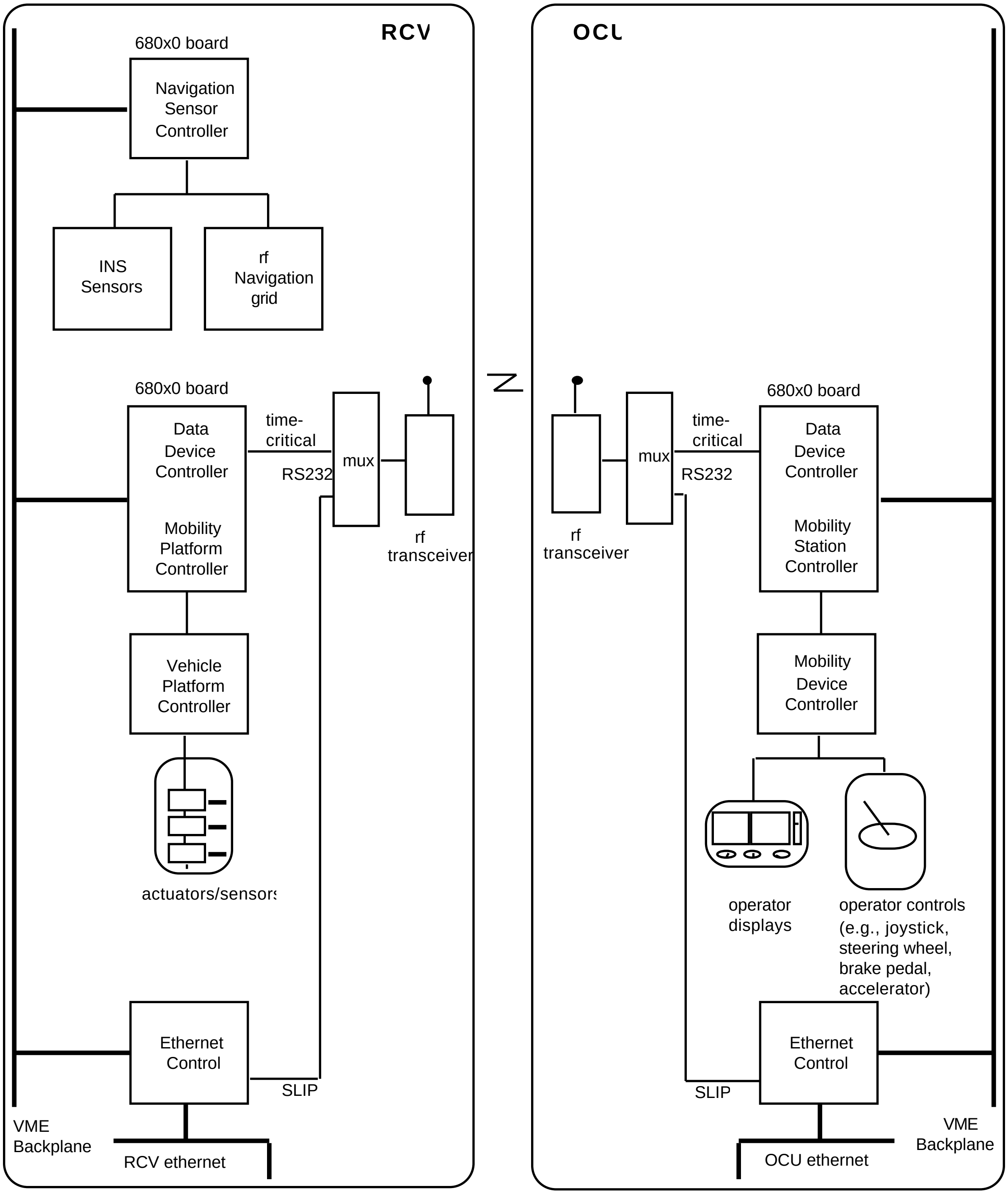

Figure 5. Current TEAM hardware configuration.

Summary

The TEAM control system requires the development and integration of many elements which allow two vehicles to perform autonomous and teleoperated functions under the control of an operator from a remote site. NIST is supporting this program by developing the vehicle control system based on their Real-time Control System. The RCS is a hierarchical, sensory-based control system, initially developed for the control of industrial robots and automated manufacturing systems. In this application, the RCS controls all vehicle mobility functions, coordinates the operations of other subsystems on the vehicle, and communicates between the vehicle and the remote operator control station.

To date, the architecture has been defined in terms of the major functional modules. A high level computing environment supports development of the modules. The mobility and communications modules required to support remote control and retro-traverse driving have been designed in detail and are in various stages of implementation and testing.

This paper has reviewed the overall control system architecture, and presented the design and implementation of the current mobility and communication functions.

References

[1] Albus, J. S., Barbera, A. J., and Nagel, R. N., Theory and Practice of Hierarchical Control, Proceedings of the Twenty-third IEEE Computer Society International Conference, Washington, DC, September 15-17, 1981.

[2] Barbera, A. J., Fitzgerald, M. L., Albus, J. S., and Haynes, L. S., RCS: The NBS Real-Time Control System, Proceedings of the Robots 8 Conference and Exposition, Volume 2 - Future Considerations, Detroit, MI, June 4-7, 1984.

[3] Albus, J. S., McCain, H. G., and Lumia, R., NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM), NIST Technical Note 1235, 1989 Edition, National Institute of Standards and Technology, Gaithersburg, MD, April 1989 (supersedes NBS Technical Note 1235, July 1987).

[4] Szabo, S., Scott, H. A., and Kilmer, R. D., Control System Architecture for the TEAM Program, Proceedings of the Second International Symposium on Robotics and Manufacturing Research, Education and Applications, Albuquerque, NM, November 16-18, 1988.

[5] Balakirsky, S., David, P., and Hillis, D., A Real Time, Automatic Target Acquisition System, Proceedings of AUVS-90, Dayton, OH, July 30 - August 1, 1990.

[6] Herman, M., Chaconas, K., Nashman, M., and Hong, T., Low Data Rate Remote Vehicle Driving, Proceedings of the Third IEEE International Symposium on Intelligent Control, Arlington, VA, August, 1988.

[7] Kraetz, W.F., Application of Robotic Convoy Technology to Combat Training Centers - Phase I, Report Number 7-90-3, Honeywell Defense Systems Group, Edina, MN, May 11, 1990.