# An Autonomous Ground Vehicle for Distributed Surveillance: CyberScout

Ashitey Trebi-Ollennu[1] and John M. Dolan[2•]
Institute for Complex Engineered Systems (ICES)[1]
The Robotics Institute[2]
Carnegie Mellon University
5000 Forbes Avenue,
Pittsburgh, PA, 15213, USA

## Abstract

This paper describes the development of an autonomous robotic vehicle for tactical distributed surveillance under the CyberScout project at Carnegie Mellon University's Institute for Complex Engineered Systems. The primary aim of the CyberScout project is to develop novel mobile and unattended ground sensor platforms that will extend the sphere of awareness and mobility of small military units. The research is focused on developing algorithms for multi-agent collaboration, efficient perception, sensor fusion, heterogeneous swarm technologies, distributed command and control, and task decomposition. The mobile robotic platforms within CyberScout range in size from microrobots (<5x5x5 cm) to All Terrain Vehicles (ATV). In this paper, the hardware system, control architecture, sensor suite, current capabilities, future research, and applications for the robotic ATV (CyberATV) are described.

## 1 Introduction

In the past decade there has been considerable effort to develop autonomous robotic vehicles for random patrols, barrier assessment, intruder detection, reconnaissance and surveillance, building entry, target detection, building or terrain mapping, and explosives neutralization. Mobile robotic platforms with the above capabilities will improve the ability to counter threats, limit risks to personnel, and reduce manpower requirements in hazardous environments. The ultimate goal of the CyberScout Project at the Institute for Complex Engineered Systems, Carnegie Mellon University is to develop mobile robotic technologies that extend the sphere of awareness and mobility of an individual or group performing such operations. By increasing sensory "reach", giving the robots a significant degree of autonomy, and providing the ability to task multiple platforms as a single logical entity, CyberScout seeks to augment human capabilities, reduce exposure to risk, and present timely, relevant information to the user.

Similar initiatives to develop Unmanned Ground Vehicles for remote reconnaissance and surveillance have been reported in the literature. Sandia National Laboratories developed the Surveillance And Reconnaissance Ground Equipment (SARGE) [1] robotic vehicle for the US DoD. SARGE is a teleoperated robot for battlefield surveillance applications without computing power to support autonomous navigation or vision processing. SARGE was built around a Yamaha Breeze 4-wheel all-terrain vehicle. It was used for early user appraisal to aid the US Army and Marines in developing tactics and doctrine for robotic ground vehicles in tactical surveillance and reconnaissance missions. Similar teleoperated battlefield mobile robot research sponsored by US DoD is reported in [2],[3],[4]. SARGE II is a second generation of SARGE with a multiprocessor computing system. SARGE II employs differential GPS (DGPS) for autonomous navigation and neural network processing of Sandia's Laser Radar (LADAR) range images for local obstacle detection and avoidance.

Robotic Systems Inc. [5] developed the Mobile Detection Assessment and Response System-Exterior (MDARS-E) for the US Army's Physical Security Equipment Management Office. The MDARS-E platform is built around a diesel-powered, hydrostatically driven vehicle. Its navigation employs a differential GPS receiver integrated with inertial and landmark referencing sensors. MDARS-E uses low-

---

[•] Send correspondence to Dr John M. Dolan at jmd@cs.cmu.edu bandwidth radio-frequency communications to transmit its real-time position for display on a site map. MDARS-E uses innovative video compression, image processing, and radar and spread spectrum techniques.

In [6] an Autonomous Robot for Surveillance Key Applications (ARSKA) is presented. This UGV is built around a Honda TRX 350 all-terrain vehicle. ARSKA is equipped with ultrasonic sensors for contour-following and obstacle avoidance. Two positioning systems have been used with the vehicle: an external optical measurement device called a tachymeter, and differential GPS (DGPS). ARSKA can be controlled using a ground station. The basic functions of the ground station are task preparation and monitoring and on-line telecontrol of the vehicle. ARSKA has been used to guard an Army storage area.

The focus of the above autonomous robotic surveillance and reconnaissance systems is the use of a single platform to accomplish the task at hand. In this type of autonomous surveillance system, minimal supervised autonomy is achieved at the expense of costly and power-intensive sensors (e.g. optical range imaging sensors) and high-bandwidth, bulky teleoperated command and control ground stations. In general, a high degree of autonomy requires considerable computing power. The path planning algorithms used generally require *a priori* knowledge of free paths and obstacles in the environment, which are not always available. Another major drawback is that the platforms' autonomous capabilities are not designed in a modular framework. It is therefore not easy to systematically incorporate higher levels of autonomous capabilities and new sensors without a significant redesign of the autonomous control architecture. These deficiencies pose considerable challenges when such autonomous platforms are to be employed in applications or environments that they were not initially designed for.

It is well known that social interactions between animals allow them to accomplish larger tasks than any one organism could accomplish alone. Similarly, autonomous robotic systems may also benefit from the use of multiple platforms. This has led to several research thrusts in the area of multi-agent or swarm systems [7], [8]. The essential characteristic of this kind of system is a non-centralized collection of relatively autonomous entities interacting with each other and a dynamic environment.

In single-platform surveillance systems, the principal cost is the sensor suite and payload. However, if a distributed multi-agent approach is employed, a larger number of sensors with limited capabilities can be utilized. Therefore it is feasible to employ more of them cost-effectively. The use of a distributed multi-agent surveillance system also enhances mission robustness, since if a few agents fail, others remain to perform the mission. Furthermore, missions can be more flexible, since groups of agents can perform different tasks at various locations. For example, the likelihood of classifying an object or target increases considerably if multiple sensors (of possibly different modalities) are focused on it from different locations.

In the CyberScout project a distributed multi-agent surveillance approach is being considered. Unlike the mass-on-mass engagements of past land warfare, future land warfare is likely to be distributed over larger areas (e.g. Kosovo, Bosnia), thus rendering single surveillance platform systems ineffective. In this type of distributed warfare, surveillance information from internetted and arrayed sensors are essential to the field commander to maintain comprehensive awareness of the surrounding battlespace and have the ability to exploit that information. In the CyberScout project, both homogeneous and heterogeneous groups of agents are being researched.

Although distributed multi-agent surveillance systems offer significant cost savings and potentially more efficient and robust solutions to the remote surveillance problem, the design of a generic system architecture that will enable the multi-agents to be tasked as a single logical unit poses considerable challenges. A key design issue that must also be addressed is the control systems architecture, which must scale well to efficiently and effectively be applicable to different agents, numerous sensors, actuators, and processes. Additionally, the architecture must be designed in a modular framework so that additional sensors, actuators or processes can be integrated readily.

Users task, control, and monitor the ATVs and other platforms which make up CyberScout through CyberRAVE (Real and Virtual Environment), a software framework and graphical user interface developed under the CyberScout project which enables rapid configuration and prototyping of cooperating groups of real and virtual robots. When no human user is present, CyberRAVE allows robots to continue to operate in autonomous mode. Further details on CyberRAVE can be found in [9]. In this paper, the hardware, control, and systems architecture for CyberATV, one of several mobile platforms with the CyberScout project, will be presented.

We have retrofitted two Polaris ATVs (named Lewis and Clark, after the famous explorers, Figure 1), automating their throttle, steering, braking, and gearing functions and giving them computation for control, navigation, sensing, and communication. The ATV is a particularly suitable platform for augmenting mobility due to its ability to carry a person for a distance of approximately 200 miles on a tank of gas. For this purpose, our ATVs can operate in manual mode, which allows actuation of the four vehicle functions (throttle, steering, gearing, and braking) via R/C joystick or wireless Ethernet, as well as in autonomous mode. In autonomous mode, the ATV has a variety of behaviors, which allow it to navigate and explore its environment.

**Figure 1 Picture Of ATV Lewis**

The rest of this paper is organized as follows. Section 2 describes the vehicle retrofit, including actuation, computing, sensing, communications, and power aspects. Section 3 describes the autonomous control architecture. Section 4 outlines current vehicle capabilities and future research directions.

# 2 CyberATV Platform

CyberATV Lewis is built around a Polaris Sportsman 500 ATV. The four vehicle functions most critical for autonomous control are throttle, steering, gearing, and braking. Each of these is provided with a mechanical actuator controllable by a computer. The computing architecture is two-tiered: a PC/104 controls vehicle locomotion, while a group of three PCs in a custom housing perform perception, planning, and communications. A generator provides auxiliary power. A functional representation of the ATV Lewis' hardware is depicted in Figure 2. In this section, the vehicle retrofit, including actuation, computing, sensing, communications and power aspects will be described.

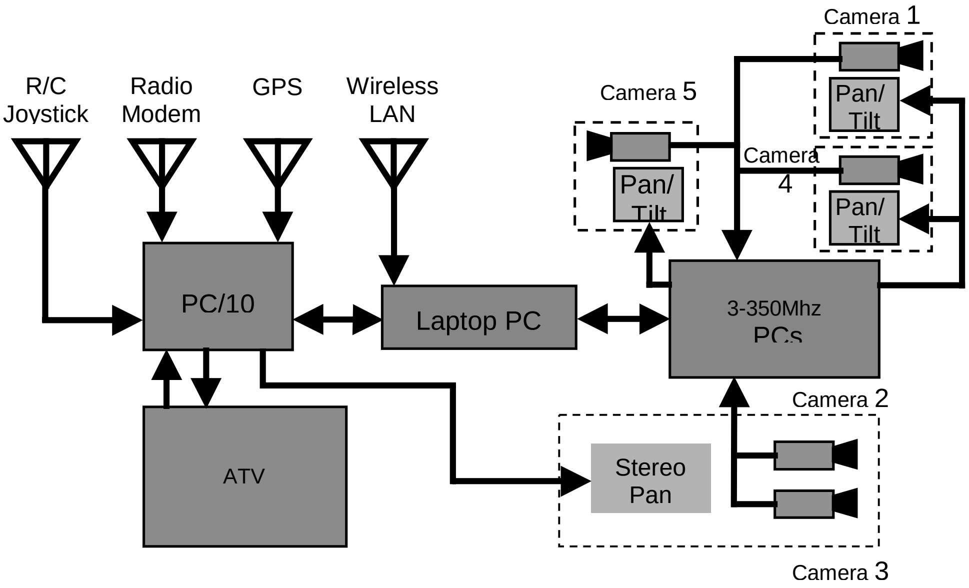

**Figure 2 CyberATV Lewis Systems Hardware Architecture**

## 2.1 Actuation

The throttle function is actuated by an R/C servomotor, whereas the automatic steering, braking and gearing functions are actuated hydraulically. Turning the steering requires considerable torque when the vehicle is stationary or moving at low speed, and the amount of space for mounting actuators on the steering is fairly limited. A hydraulic actuator was therefore chosen due to its high power-to-weight ratio and resulting compact size relative to a comparably powerful electric motor. The same hydraulic system was then used for gearing and braking. A block diagram of Lewis' actuation is shown in Figure 3.

### 2.1.1 Steering Actuator

The steering actuation consists of a 4.5" hydraulic piston aligned parallel to the steering push rod on the right front wheel (from the rider's view). The rod end of the cylinder is attached to the suspension crank arm. The cap end of the cylinder has a trunnion mount and is attached to the ATV chassis. A proportional directional control valve is used to control hydraulic flow to the steering cylinder. The flow output of the valve can be affected by a variable voltage input command between 0 and 10 volts [10]. The slew rate, rather than the position, of the steering is roughly proportional to this command. A resistive linear potentiometer is incorporated at the cap end of the cylinder to provide feedback (0-5 volts) and is connected to a DM5416 Real-Time Devices analog/digital I/O board on the PC/104 stack.

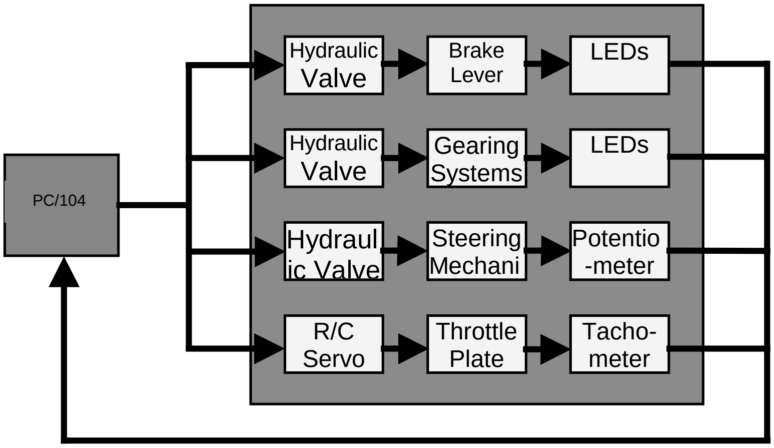

**Figure 3 Block diagram of CyberATV Actuation Systems**

### 2.1.2 Gear Actuator

Two linear hydraulic actuators are mounted back-to-back for shifting into neutral, high and reverse gears. The two cylinders are connected to the gear selection assembly by an adjustable rod. Neutral position corresponds to cylinder 1 retracted and cylinder 2 extended, high gear corresponds to cylinder 1 retracted and cylinder 2 retracted, and reverse gear corresponds to cylinder 1 extended and cylinder 2 extended. A three-way control valve is used to control the hydraulic linear actuators [10]. The existing ATV gear selection light is used for feedback and is connected to a DM 5416 Real-Time Devices analog/digital I/O board on the PC/104 stack.

### 2.1.3 Brake Actuator

The pre-retrofit ATV braking system consists of a single-lever, all-wheel hydraulic disc brakes and a hydraulic rear foot brake. In the retrofit, only the rear brake is actuated, using a low-profile air/oil cylinder. A directional control valve is used to control the brake cylinder. This nominally provides for only on/off braking, but the brake can be pumped, or cycled, at different rates in order to provide intermediate levels of braking. There is no explicit feedback from the brake.

### 2.1.4 Hydraulic System

The hydraulic pump is a Parker series D gear pump equipped with an integral four-quart reservoir and a system relief valve. The pump is driven by a NEMA 56 C frame single-phase AC motor which uses a 120 V AC input. The pump motor is controlled by a single solid-state AC contactor located inside the electric enclosure [10]. To reduce heat and power, the pump circuit includes an unloading valve circuit to dump fluid at low pressure when high-pressure fluid is not required. A 2-quart piston-type accumulator is included in the circuit to reduce pressure instabilities.

### 2.1.5 Throttle Actuator

The ATV is equipped with a 4-stroke, liquid-cooled engine with a 34-mm Mikus carburetor, which is regulated by a throttle plate. A Futaba R/C servomotor is used to accomplish the opening and closing of the throttle plate via two cables. One cable connects the servo to the throttle control lever on the handlebar, while a second cable connects the servo to the throttle plate. Each cable is passed through a friction clamp that holds the cable in place and also provides a means to adjust the length of the cable. The throttle plate setup incorporates a torsional spring, which applies a closing torque on the throttle plate sufficient to back-drive the servo to idle throttle position when power is turned off. The choice of an R/C servo for throttle actuation is based on its low cost, small size, internal closed-loop position control, low power consumption and easy control signal generation. Speed feedback is obtained via a tach generator mounted in the gearbox. The tachometer is a Servo-Tek SB-763-2 A, designed for use in applications requiring an output signal between 1 and 10 volts/1000 RPM. The output of the tachometer is connected to a DM5416 Real-Time Devices analog/digital I/O board on the PC/104 stack.

## 2.2 Computer System

The computational architecture is two-tiered: locomotion (low-level processing) is performed by a PC/104, while planning, perception, and communications (high-level processing) are performed by a set of three networked PCs in a custom housing mounted on the front of the vehicle.

The PC/104's economy of size (4 by 4-inch cards), low power requirements, and mechanically rugged design are well suited for outdoor applications. Further advantages are the availability of components from a variety of vendors and the ability to use virtually all of the program development tools used for IBM PC's.

The configuration of the PC/104 employed includes four boards from three different vendors:

- One Versalogic VSBC-2 CPU featuring a 32-bit, 133 MHz, 5x86 CPU chip. Its performance is equivalent to a Pentium 75 MHz machine.
- One WinSystems PCM-COM4A serial interface board. This board is a 4-channel serial INS 8250-compatible PC/104 module. Each channel supports RS-232, with RS-485 and RS-422 as options.
- One WinSystems PC/104 video board, the PCM-FPVGA. It supports standard VGA CRT output as well as a variety of flat panel displays using optional flat-panel adapter (FPA) modules. Other options include multi-video display capability, PC video input, and NTSC video output.
- One DM5416 analog/digital I/O board from Real-Time Devices. The DM5416 has 8 differential or 16 single-ended analog input channels with an input range of -10 to +10 volts, 8 bit-programmable digital I/O lines with advanced digital interrupt modes, one 16-bit digital-to-analog output channel, 8 port-programmable digital I/O lines, two 16-bit timer/counters, and an 8 MHz clock.

The high-level processing suite consists of 3 Pentium II 350MHz PCs, each equipped with Ethernet card, Matrox frame grabbers. Each PC has access to all the cameras on the ATV via a customized designed terminal switch. Each PC controls one pan/tilt mechanism. The three PCs are connected together in a star network with cable segments to a hub. The wiring employed on the star-based network is 10Base-T, but different cable types can be used to connect to the hub if higher bandwidth is desired.

## 2.3 Sensors

Figure 4 depicts Lewis with sensors labeled. Navigational sensing is performed by a 20-cm resolution NovAtel DGPS unit. Perceptual sensing is currently restricted to vision, which we are trying to exploit to the fullest due to its passive, unobtrusive nature. Currently each vehicle is equipped with five cameras, a panning stereo pair in front for obstacle avoidance and mapping, and three pan/tilt cameras for surveillance, one each located at the front left, front right, and rear. Stereo vision is employed to have a 3-dimensional understanding of the environment to perform free space mapping, thus enabling obstacle detection and path-planning. Further details about CyberATV's perceptual sensing can be found in [11].

## 2.4 Communications

Communications between the low-level and high-level processors are currently via a serial cable, but can be upgraded to wired Ethernet if increased bandwidth is desired. Communications with other platforms and with remote users are performed via wireless Ethernet using 915MHz Wavelan technology.

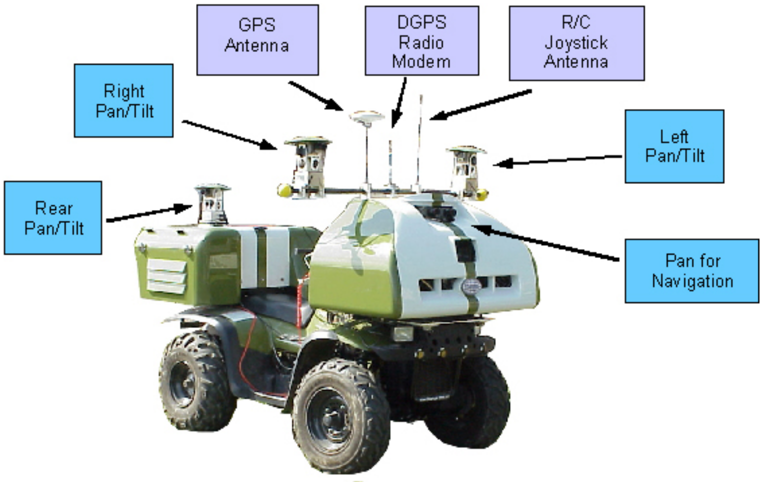

**Figure 4 ATV Lewis with labeled sensor suite**

## 2.5 Power System

Consideration was given to adding a take-off to the vehicle clutch in order to provide power for the systems added to the vehicle by the retrofit. We discarded this solution, however, for two reasons. First, it was mechanically infeasible due to excessive motion of the clutch drive shaft. Second, the amount of power available from the alternator was projected to be insufficient. As a result, a 2.5kW Honda generator was mounted on the rear of the vehicle to provide auxiliary power.

The generator provides two 120V AC, 15A outputs. One of these powers the hydraulic motor. The other is converted to 24V, 12V and 5V DC for use by the sensors, relays, optoisolators, and hydraulic valves. 5 volts are supplied separately to the PC/104 by the ATV's battery via a 12-to-5V DC converter.

# 3 CyberATV Autonomous Control Architecture

The autonomous control architecture design is partitioned into three levels (see Figure 5). The highest level of control, the Task module, executes mission tasks phrased in symbolic terms, e.g., “explore the environment”. The highest level of control is built into CyberRAVE and therefore will not be presented here. See [9]for further details.

The lowest level of control, the vehicular control module, and it drives the actuators for steering, braking, throttle, and monitors navigation sensor data (position, velocity, acceleration and posture) of the mobile platform.

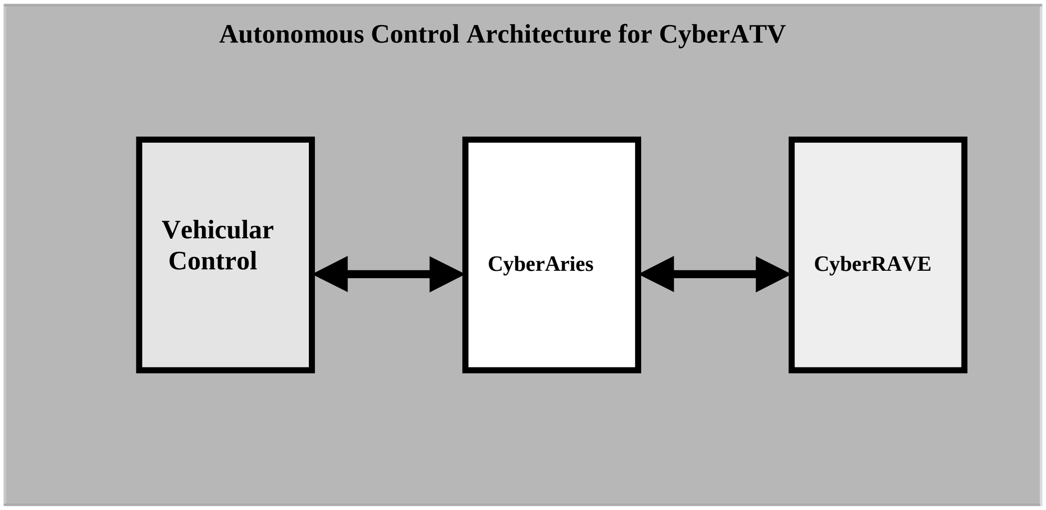

**Figure 5 Block diagram of CyberATV Control Architecture**

The mid-level of the control architecture is called CyberAries (Autonomous Reconnaissance and Intelligent Exploration System). CyberAries is designed according to an agent-based computational paradigm. The CyberAries and vehicular control blocks are resident on the high-level (Pentium II 350 MHz) and low-level (PC/104) vehicle processors, respectively. CyberRAVE is usually resident on a remote laptop or workstation, and it is linked to CyberAries via a wireless Ethernet using 915MHz Wavelan technology. Communication between the vehicular control block and CyberAries is via a RS232 serial cable. The rest of the section will detail some of the critical components of the vehicular control block and CyberAries.

## 3.1 Vehicular Control Block

The vehicular control block (Figure 6) receives commands from CyberAries, e.g. mode selection, desired position, desired steering angle, etc. The vehicular control block also feeds back the vehicle states, e.g., speed, steering angle, position and posture to CyberAries. There are two basic modes in the vehicular control block, manual and autonomous Mode. In manual mode, the four functions of the vehicle can be controlled directly from CyberRAVE (using a laptop or wearable computer) or a radio control joystick. Remote control of the vehicle is currently done using line of sight, but a teleoperational system is being considered.

In autonomous mode, CyberAries commands desired speed and waypoints to the vehicular control block. Waypoint navigation is accomplished using GPS for position feedback and pure pursuit steering.

The automatic control of gearing and braking involves the straightforward setting of various hydraulic valves. Speed/throttle and steering control, however, involve interesting challenges which are detailed below.

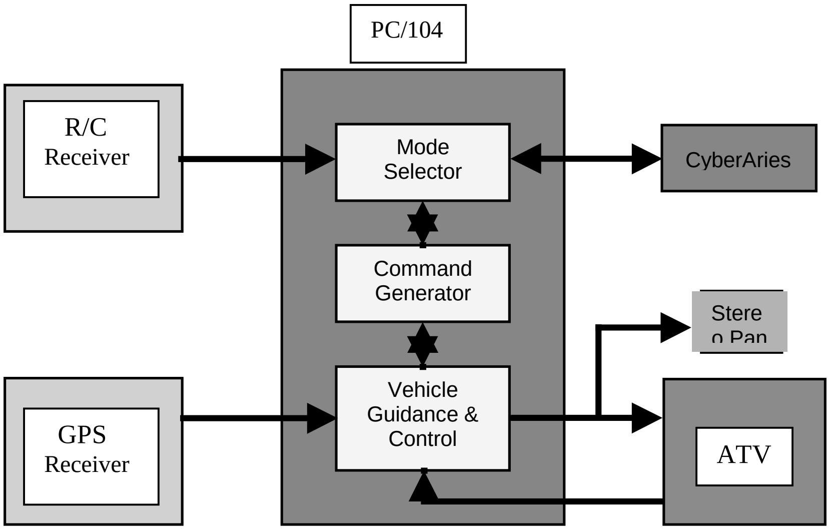

**Figure 6 Vehicular control block diagram**

## 3.1.1 Steering control for ATV

The objective of the steering control system design is to provide precise locomotion, pointing, posture and robustness to external disturbances. A block diagram of the steering controller is shown in Figure 7. A simple initial design was bang-bang control. This resulted, however, in large errors and instability when small steering angles were commanded. A fuzzy PD controller was then developed using data collected and experience gained from the bang-bang control scheme. The fuzzy PD algorithm was found to produce smooth control and fairly accurate pointing for navigation and visual tracking. Figure 9 depicts the ATV steering response to commanded steering angle.

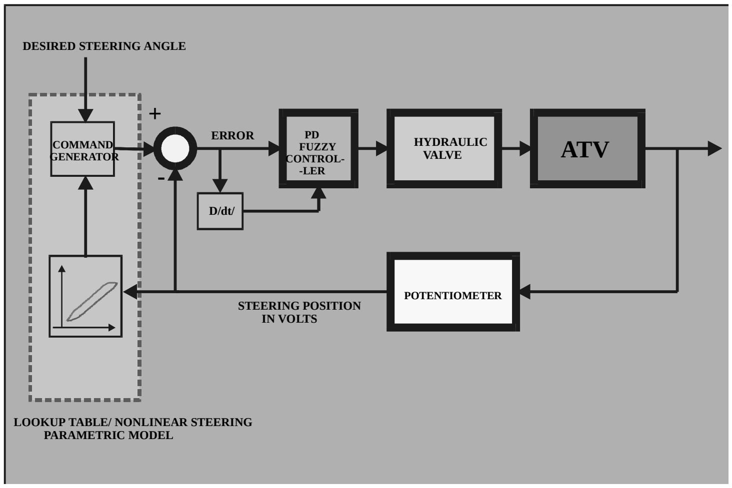

Figure 7 Block diagram for CyberATV steering controller

The described steering control was initially applied to the voltage feedback received from the steering potentiometer. Due to mechanical hysteresis in the steering, however, this voltage is not linearly related to the actual steering angle. An experiment was set up to calibrate the output of the steering potentiometer to the effective front wheel angle. Based on measurements characterizing the steering hysteresis loop, we developed a steering model, which establishes the nonlinear relationship between voltage and angle (Figure 8, in which pixels correspond to steering angle). This model allows us to reduce steering angle error from as great as 5 degrees to +/-1 degree.

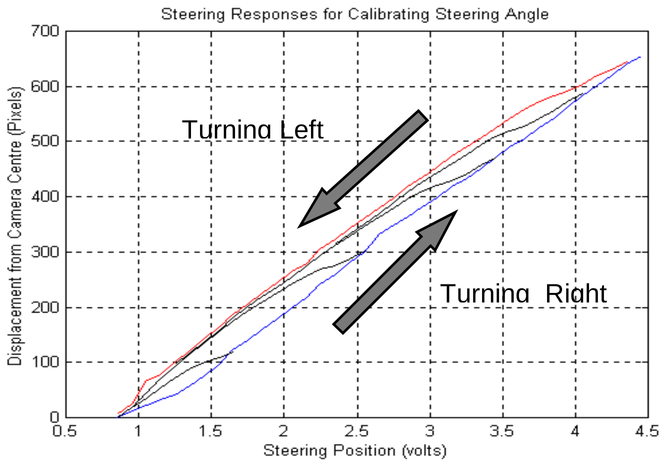

**Figure 8 Steering angle Calibration**

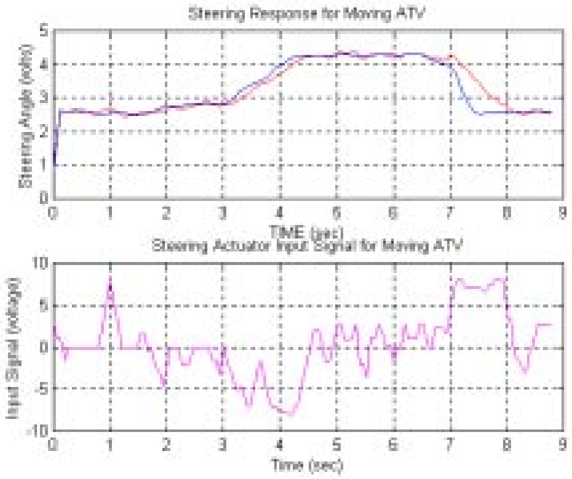

**Figure 9 ATV steering response for Fuzzy PD steering controller**

## 3.1.2 Speed Control

Since the ATV is equipped with cameras for navigation, it is essential that the vehicle move smoothly at fairly low speeds that will allow real-time processing of images. A block diagram of the speed controller is shown in Figure 10. The two main challenges in designing an effective speed controller for the ATV are:

- the lack of a complete mathematical model for the engine;
- the highly nonlinear nature of the engine dynamics, especially for the target low speed range of 3-30MPH;
- the belt slippage of the Automatic Polaris Variable Transmission (PVT) at low speeds.

Each of these factors makes the use of classical control strategies such as PID control ineffective. Using experience and data collected from extensive experiments conducted on the ATV throttle mechanism, an adaptive fuzzy throttle control algorithm was designed [12]. A candidate Lyapunov function was employed in the adaptive law synthesis to ensure convergence. The adaptive fuzzy throttle control produces smooth vehicle movement, robustness with respect to varying terrain, and commanded speeds in the range 2MPH to 30MPH. Figure 11depicts the ATV speed response to a selected speed.

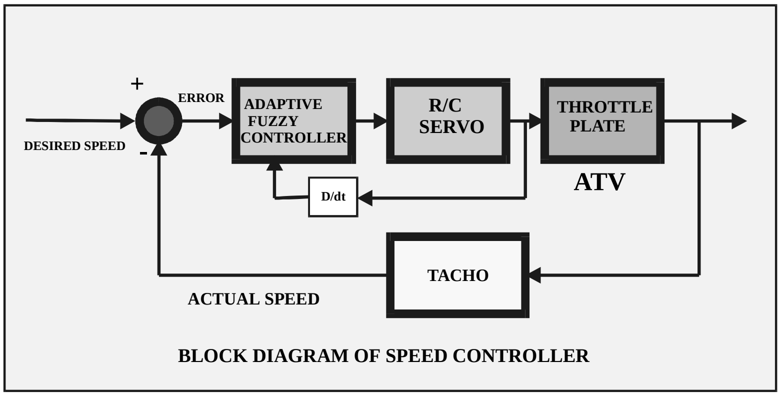

**Figure 10 Block diagram of Speed control**

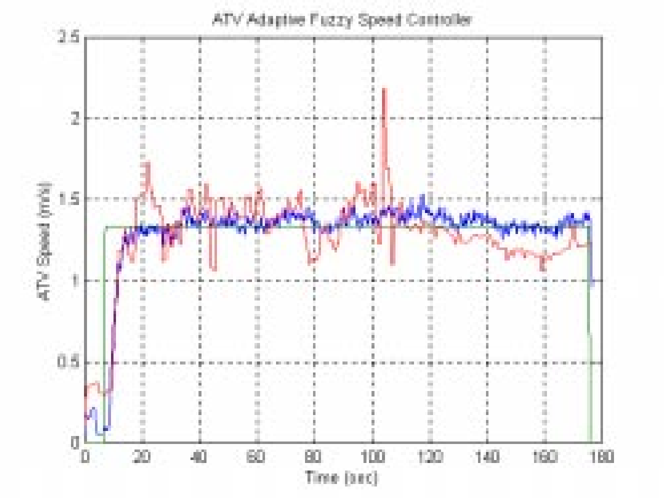

**Figure 11 ATV speed response for adaptive fuzzy throttle controller for 2.9 MPH (1.3m/s)**

The described low-level control algorithms have been implemented on two other ATVs with very little modifications in line with the design objectives.

## 3.2 High-Level Control Architecture

Figure 12 depicts the block diagram of CyberAries. There are four basic functional blocks: Perception, Mission Planner, Distribution Layer and World Model. The fifth block, Vehicular Controls is added here for completeness to assist the reader in the discussions and descriptions that follow. In the subsequent discussions each of the five blocks will be referred to as agents, e.g., distribution layer agent, mission planner agent, perception agent, world model agent, and vehicular controls agent. Agents are independent processes that can be written by an end-user and started on an as-needed basis. Each of the four agent blocks of CyberAries is made up of a collection of agents, thus offering a distributed and decentralized system in which agents run in parallel, concurrently and asynchronously, performing their own sensing (stimulus sources) and computation to command distinct outputs to other agents or actuators.

Distributed agent architectures generally require a dependable form of inter-agent communications, because the agents that make up such systems are strongly interrelated. CyberAries provides powerful inter-agent communications capabilities that greatly simplify the job of developing ensembles of cooperating agents. It is this architecture that gives CyberAries unparalleled extensibility. It is easy to augment CyberAries by simply writing new agents to provide new services.

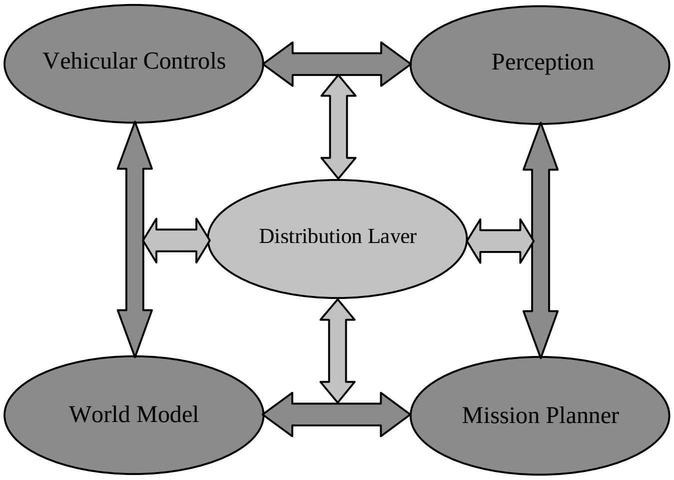

**Figure 12 CyberAries Block Diagram**

### 3.2.1 Distribution Layer

The core of CyberAries is the distribution layer agent. In the simplest case, it is inhabited by a group of agents with a unique schedule of activity for each agent. The distribution layer agent is responsible for facilitating the communications, processing, and sensing resources among the various agents active in CyberAries (sometimes across platforms). The three key agents that make up the distribution layer agent are the transmitting agent, observer agent and distribution agent.

The distribution agent handles the necessary details required to establish communications between two agents upon receiving a request from an agent. The distribution agent does the necessary checks and assigns a transmission agent for the particular request. The distribution agent is able to integrate the entire multi-agent surveillance system into a homogeneous set of resources. Any agent running on a surveillance platform can directly make use of any resource on any other platform within the multi-agent surveillance system.

The observer agent provides the necessary monitoring required to perform load balancing. The observer agent collects data on the various agents running, recording their processing rate with respect to the arrival rate of stimuli (input) to ensure that agents are not overloaded. If overloading is detected, the observer agent notifies the distribution agent and transmission agents. The distribution agent, in collaboration with the observer agent, tries to find another agent which has excess processing capacity and the necessary capabilities to handle the excess stimuli (inputs). Otherwise, the transmission agent is requested to reduce its transmission rate. Refer to [11] for further details on the distribution layer.

### 3.2.2 Perception

The perception agent consist of a collection of agents that perform tasks such as moving object detection, classification, geolocation and tracking, obstacle avoidance, and landmark detection. These agents run in parallel, concurrently and asynchronously, performing their own sensing (stimulus sources) and computation to command distinct output to other agents or actuators.  This approach allows for collaboration among the various agents in the perception agent and may lead to interesting emergent collaboration mechanisms.

### 3.2.3 World Model

The world model agent consists of a collection of agents. Current research is focused on developing a path planner agent, electronic map interpreter agent and a map building agent.

The map building agent will use a 2-D grid-based map of an area being explored by the vehicle(s). Information about the terrain will be received from the obstacle detection agent in the perception agent. The map building agent will also be capable of fusing two or more maps from other vehicles to get a global map.

The electronic map interpreter agent will be capable of converting relevant information from any given electronic map into a 2-D grid-based map format.

The path-planning agent will generate paths from 2-D grid-based maps that will guarantee successful completion of an assigned task. This requires algorithms for both static and dynamic path planning for multiple vehicles.

### 3.2.4 Mission Planner

The mission planner agent interfaces directly with CyberRAVE.  Commands received from CyberRAVE by the mission planner agent are further decomposed into subgoals that need to be accomplished in order to complete the task successfully.

By way of example, suppose a user tasks CyberATV Lewis to stakeout Hamburg Hall (a building at CMU), and assume that the world agent has a map of CMU. The mission planner agent will task the path-planning agent resident in the world model agent to plan a path from the current position of Lewis to Hamburg Hall. The rest of the mission planner agent's tasking mechanism is shown in Figure 13. Most of the tasks in the flow chart are self-explanatory. However, the flowchart illustrates some of the unique user interactions via CyberRAVE. If there is no DGPS, the vehicle can be moved manual via teleoperation or in autonomous mode via landmark navigation. Landmark navigation requires the user to select quite distinct landmarks for the vehicle to use as cues to navigate. These could be buildings, sidewalks or road signs.

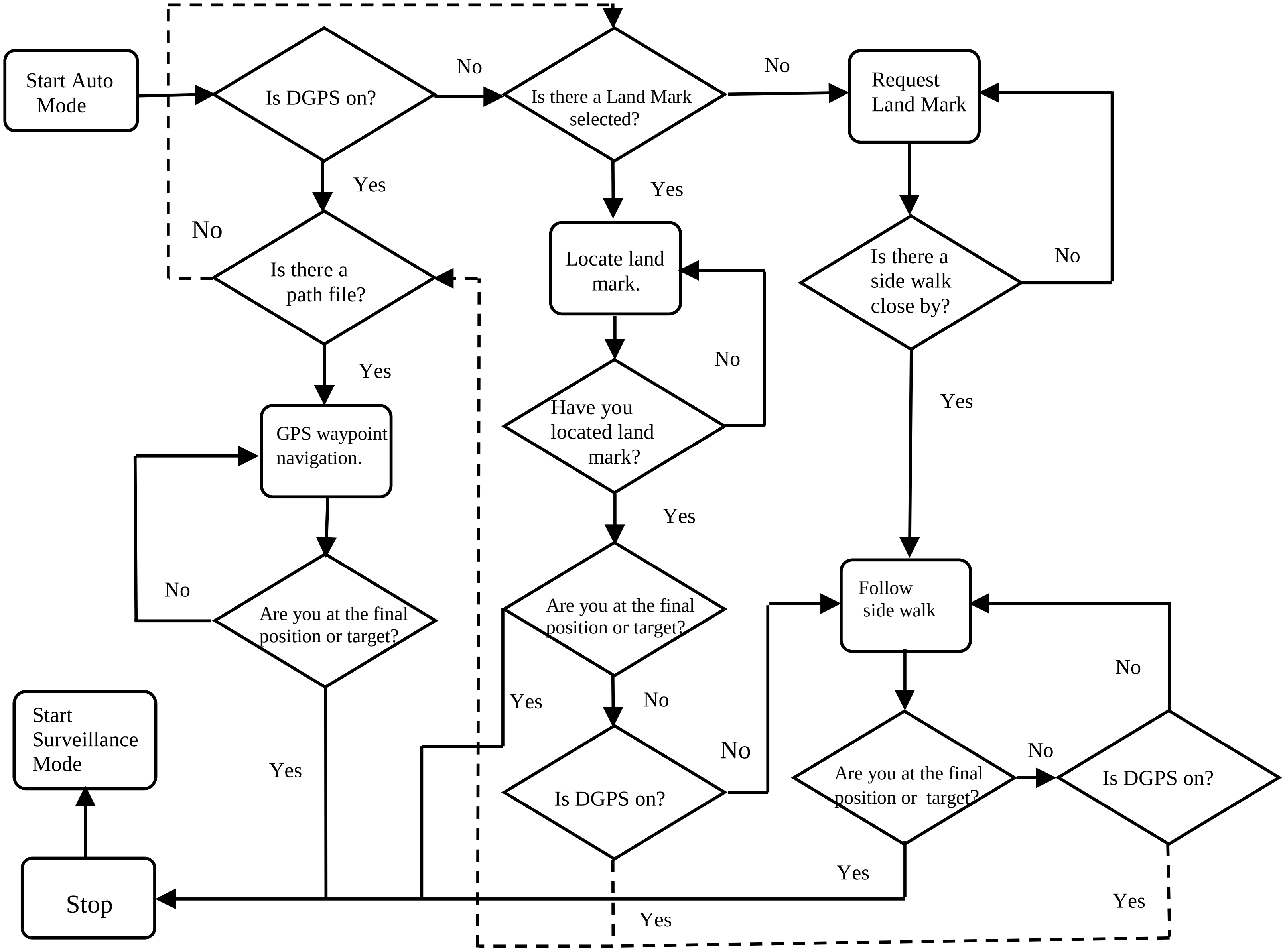

Figure 13 Flow chart of Mission Planner Agent Tasking

# 4 Current Capabilities and Future Work

We have demonstrated the following capabilities on the ATVs:

- remote manual control using various input devices (R/C joystick and laptop/wearable computer via wireless Ethernet).
- autonomous GPS-based waypoint navigation.
- autonomous convoying based on visual tracking.
- sidewalk and path-tracking.
- potential-field obstacle avoidance.
- robust personnel/vehicle classification using vision.
- window detection and classification for building surveillance.

Our current and future work involves integrating the above-described capabilities in order to give the ATVs the ability to conduct a building "stakeout". In this scenario, a lead vehicle travels in autonomous mode towards a site whose general location is known, using DGPS, sidewalk tracking, and landmark-based tracking as appropriate and available. When it reaches a location from which building surveillance can begin, it communicates its position and relevant information about the path it traversed to the stay-behind vehicle, which then joins it at a location appropriate for good coverage of the building of interest. Coverage is adjusted in response to activity at the site. The ATVs will simultaneously conduct window detection and personnel/vehicle classification, as well as perform correspondence in order to track targets of interest.
Using CyberRAVE, a remote user can intervene at any time in order to view information, manually adjust the action of an ATV, or provide tasking.

## Acknowledgements

We gratefully acknowledge the support of DARPA contract F04701-97-C-0022, "An Autonomous, Distributed Tactical Surveillance System", in performing this work. C Spence Oliver for his help with the pictures and block diagrams.

# 5 References

1. Sackos, J.B.P.a.J., *An Advanced Unmanned Vehicle for Remote Applications*, . 1998, Sandia National Laboratories.

2. C. D. Metz, H.R.E.a.S.M. *Recent Developments in Tactical Unmanned Ground Vehicles*. in *Association for Unmanned Vehicle Systems*. 1992. Huntsvilkle, AL.

3. S. Szabo, K.M., H. Scott, S. Legowik and R. Bostelman. *Control System Architecture for Unmanned Vehicle Systems*. in *Association for Unmanned Vehicle systems*. 1992. Huntsvilkle, AL.

4. Aviles, W.A. *Issues in Mobile robotics: The Unmanned Ground Vehicle Program TeleOperated Vehicle*. in *SPIE*. 1990.

5. Kurtur, J., *Robotic Rover*, . 1993, Systems Technology, MD.

6. T. Schonberg, M.O., J. Suomela, A. Torpo and A. Halme. *A Small Scaled Autonomous Test Vehicle for Developing Autonomous Off-Road Applications*. in *Int. Conference on Machine Automation*. 1994. Tampere, Finland.

7. Nelson Minar, Roger Burkhart, Chris Langton and Manor Askenazi, *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*, . 1996.

8. Mataric, M.J., *Interaction and intelligent behavior.*, in *Department of Electrical Engineering and Computer Science*. 1994, Massachusetts Institute of Technology: Cambridge.

9. Kevin Dixon, J.D., Robert Grabowski, John Hampshire, Wesley Huang, Christiaan Paredis, Jesus Salido, Mahesh Saptharishi, and Pradeep Khosla. *RAVE: A Real and Virtual Environment for Multiple Robot Systems*. 1999. IROS'99.

10. Inc., K.T., *9702 Robor ATV*, . 1997, K2T Report.

11. Christopher P. Diehl, M.S., John B. Hampshire II, and Pradeep K. Khosla. *Collaborative Surveillance Using Both Fixed and Mobile Unattended Ground Sensor Platforms*. in *AeroSense' 99*. 1999. Orlando, Florada: SPIE.

12. A.Trebi-Ollennu and John M. Dolan, *Adaptive Fuzzy Throttle Control for an All Terrain Vehicle*, . 1999, Institute for Complexed Engineered Systems, Carnegie Mellon University.