Enhancing Functionality and Autonomy in Man-Portable Robots

E. Biagtan Pacis[a], H.R. Everett[a], N. Farrington[a], D. Bruemmer[b]
[a]Space and Naval Warfare Systems Center, San Diego
[b]Idaho National Engineering and Environmental Laboratory

ABSTRACT

Current man-portable robotic systems are too heavy for troops to pack during extended missions in rugged terrain and typically require more user support than can be justified by their limited return in force multiplication or improved effectiveness. As a consequence, today's systems appear organically attractive only in life-threatening scenarios, such as detection of chemical/biological/radiation hazards, mines, or improvised explosive devices. For the long term, significant improvements in both functionality (i.e., perform more useful tasks) and autonomy (i.e., with less human intervention) are required to increase the level of general acceptance and, hence, the number of units deployed by the user. In the near term, however, the focus must remain on robust and reliable solutions that reduce risk and save lives. This paper describes ongoing efforts to address these needs through a spiral development process that capitalizes on technology transfer to harvest applicable results of prior and ongoing activities throughout the technical community.

Keywords: robotics, sensors, autonomy, localization, mapping, SLAM, communications, technology transfer.

1. BACKGROUND

While historically there has been a definitive trend towards making mobile robots smarter in order to reduce the control burden on the operator (with much of the progress made in laboratory prototypes), all systems deployed by the U.S. Department of Defense (DoD) in theatre to date have been strictly teleoperated. Significant upgrades in both functionality and autonomy are required to improve overall effectiveness and increase the level of acceptance by military users. The Tactical Mobile Robot (TMR) program, sponsored by the Defense Advanced Research Projects Agency (DARPA), was transitioned to the Space and Naval Warfare Systems Center, San Diego (SSC San Diego) at the end of FY-02, providing a convenient enabling mechanism for technology transfer into ongoing development efforts funded by the Joint Robotics Program (JRP). Initial successes in this regard led to the formal establishment in FY-03 of the Small Robot Technology Transfer Program. Funded by the Office of the Secretary of Defense (OSD), SSC San Diego has been tasked to extract relevant aspects of various research activities, port them to related projects, and foster emergent technology transfer opportunities. The main objective is to improve the functionality and autonomy of the small mobile robots in the Robotic Systems Pool, a collection of available hardware maintained by SSC San Diego for temporary loan to various government organizations at the federal, state, and local levels.

1.1 Approach

In addressing specific needs identified by users of the Robotic Systems Pool (and other current programs), the initial focus will be to evaluate, integrate, and extend appropriate technologies developed under various DARPA programs, such as TMR, Software for Distributed Robots (SDR), and Mobile Autonomous Robot Software (MARS), with a structured approach as follows:

- Prioritize the potential functionality/autonomy upgrades based on user feedback.
- Evaluate state-of-the-art results of prior and ongoing development efforts that could support these upgrades.
- Extract relevant aspects of the different approaches and port into transition platforms for evaluation and characterization.
- Integrate and debug best features of the most promising algorithms for optimized solution.
- Ensure eventual Joint Architecture for Unmanned Systems (JAUS) compliance.

- Provide an enabling mechanism for the transfer of relevant results into ongoing development programs as needed.

Under a Memorandum of Agreement, SSC San Diego has subsequently tasked the Idaho National Engineering and Environmental Laboratory (INEEL) to assist in the coordinated development, evaluation, and transfer of robotics technology that mutually benefits both DoD and Department of Energy (DoE) missions. This arrangement has two obvious advantages: 1) The INEEL Robotics Group, with similar objectives and experience, can augment the available manpower resources, allowing more technology options to be evaluated; and 2) active DOE involvement opens up another major conduit for exporting results into relevant user applications. In addition to accelerating research and development efforts as outlined in OSD's *Joint Robotics Program Master Plan*[1], this synergistic teaming will also expedite progress towards functional objectives set down in DOE's *Critical Technology Roadmap for Robots and Intelligent Machines*[2]*,* thus benefiting a variety of DOE missions (i.e., decontamination and decommissioning tasks, environmental monitoring, security applications, homeland defense, critical infrastructure protection, and emergency response).

1.2 Test and Evaluation Platforms

The ROBART series of laboratory research prototypes has served in developing the component technologies needed to support the Mobile Detection Assessment Response System (MDARS) robotic security program.[3] While *ROBART I* (1980-1982) could only detect a potential intruder, *ROBART II* (1982-1992), shown in Figure 1, could both detect and assess, thereby increasing its sensitivity (i.e., probability of detection), with a corresponding reduction in nuisance alarms. Other research thrusts included implementation of an absolute world model, automated localization techniques to null out accumulated dead-reckoning errors, and reflexive (sensor-assisted) teleoperated control concepts to minimize the driving burden on the operator when under manual control.[4]

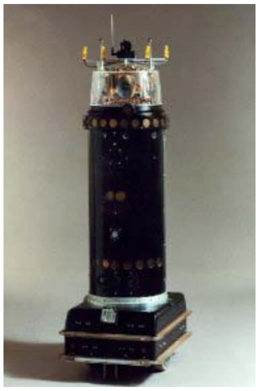


Figure 1. ROBART II.

As the third-generation prototype, *ROBART III* (1992-present) was originally intended to demonstrate the feasibility of automated response. For purposes of illustration, a pneumatically powered six-barrel Gatling-style weapon was used to that fire simulated tranquilizer darts or rubber bullets. Early work extended the concepts of reflexive teleoperation into the realm of coordinated weapons control (i.e., sensor-aided control of mobility, camera, and weapon functions).[5]

Starting in FY-03, the navigation and collision avoidance schemes began to undergo significant enhancements through technology transfer of improved algorithms developed under DARPA's TMR and MARS programs. To support the more sophisticated navigation, collision avoidance, mapping, and surveillance schemes, appropriate hardware upgrades have also been made, including a MicroStrain gyro-stabilized magnetic compass, a KVH fiber-optic rate gyro, a SICK scanning laser rangefinder, a Visual Stone 360-degree omni-cam, and a Canon pan-tilt-zoom (PTZ) camera. For these and other reasons (i.e., 90-amp-hour battery, available source code, extensive self diagnostics), *ROBART III* (Figure 2) was selected as the optimal laboratory development platform for evaluating the various candidate software algorithms under consideration. An iRobot *All Terrain Robotic Vehicle (ATRV)* has also been loaned to INEEL to serve as a secondary evaluation platform for their contributing efforts.

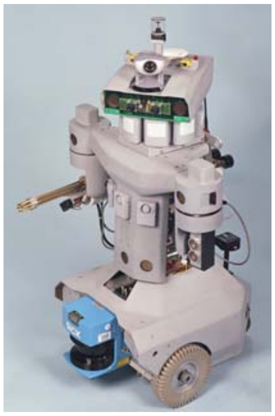


Figure 2. ROBART III.

2. CURRENT TECHNOLOGIES UNDER EVALUATION

SSC San Diego and INEEL are collaborating with a variety of TMR and MARS participants and other government agencies, with the initial focus (i.e., improved navigation) involving the partners and their associated contributions as depicted in Figure 3. Key aspects of these contributing technology sources will be briefly discussed in the following subsections.

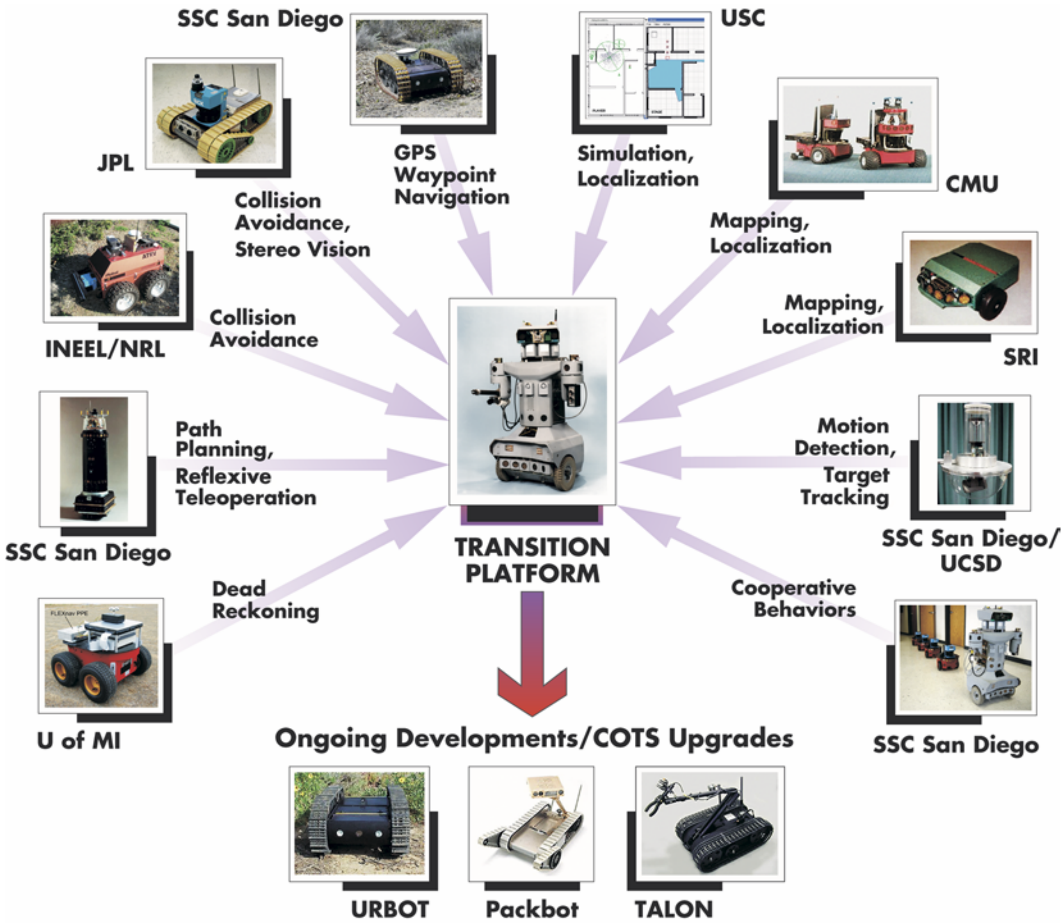


Figure 3. Current technology transfer efforts and partners. The degree of behavioral complexity generally increases in clockwise fashion from lower-left to lower-right.

2.1 Gyro-Enhanced Dead Reckoning

The "enhanced dead-reckoning" upgrade being integrated on *Robart III* is the *Fuzzy Logic Expert navigation (FLEXnav) Proprioceptive Position Estimation (PPE)* system developed at the University of Michigan, which employs three-axis gyro information to factor in vehicle tilt when calculating heading.[6, 7] While high quality three-axis inertial measurement units (IMUs) are commercially available, they typically are too expensive, bulky, and power hungry for most small mobile robot applications. The *FLEXnav PPE* system uses a low-cost two-axis Coriolis gyro for pitch and roll axes, two low-cost accelerometers, and a high-quality fiber-optic gyro for the yaw axis. In order to achieve the required accuracy from low-cost sensors (which have insufficient response time for high velocities and rough terrain), sensor integration conditions that reflect the physical functioning of each sensor are fused with the sensor data itself, using the rule-based *FLEXnav* method to map inputs and outputs, deal with imprecision associated with the noisy low-cost sensors, and handle nonlinear models of arbitrary complexity.

Custom-designed and custom-built, the *FLEXnav PPE* system can be highly optimized for the type of robotic vehicle as well as the intended application. The University of Michigan has installed *FLEXnav PPE* system on a variety of platforms, such as an iRobot *ATRV*, a Pioneer *AT*, a Pioneer *2-AT*, a Rocky Mars Rover clone, and a Segway Robotic Mobility Platform. *FLEXnav PPE* has been successfully tested over a wide range of terrain, including smooth flat surfaces (both indoor and outdoor), moderately rugged rolling terrain, and very rugged 3-D scenarios (i.e., the Waive Field, a commissioned sculpture at the University of Michigan).

2.2 Path Planning/Collision Avoidance/Reflexive Teleoperation

From a navigational perspective, the type of control strategy employed on a mobile platform runs the full spectrum defined by *teleoperated* at the lower end through fully *autonomous* at the upper extreme. A *teleoperated* machine of the lowest order has no onboard intelligence, and blindly executes the drive and steering commands sent down in real-time by a remote operator. A fully *autonomous* mobile platform keeps track of its position and orientation and typically uses some type of world modeling scheme to represent the location of perceived objects in its surroundings. A very common approach is to employ a statistical occupancy-grid representation,[8] where each cell in the grid corresponds to a particular "unit square" of floor space. The numerical value assigned to a cell represents the probability that its associated physical location is occupied by some object, with a value of zero indicating free space (i.e., no obstacles present).

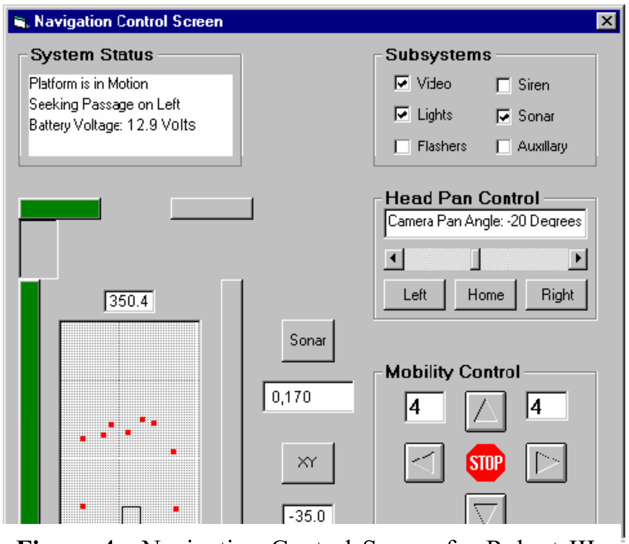
The existence of an absolute world model allows for automatic path planning and subsequent route revisions in the event a new obstacle is encountered. Unfortunately, the autonomous execution of indoor paths generally requires *a priori* knowledge of the floor plan of the operating environment, and in all cases the robot must maintain an accurate awareness of its position and orientation. Accordingly, traditional autonomous navigation techniques have until recently been of limited utility for applications where a requirement exists to enter previously unexplored structures of opportunity as the need arises. (More recent efforts have made some noteworthy progress in this arena, to be discussed later in Section 2.4.)

Teleoperated systems, on the other hand, permit remote operation in such unknown environments but conventionally place unacceptable demands on the operator. Simply driving a teleoperated platform using vehicle-based video feedback is no trivial matter and can be stressful and fatiguing even under very favorable conditions, particularly on small robots where the camera is very low to the ground. If a remote operator has to master simultaneous inputs for drive, steering, camera, and weapons control, the chances of successfully performing coordinated actions in a timely fashion are minimal.

2.2.1 SSC San Diego

Easing the driving burden on the operator was a major force behind the development of the *reflexive teleoperated control* scheme employed on *Robart II.* Speed of the vehicle and direction of motion were servo-controlled by an onboard processor in response to local sensor inputs but under the high-level supervisory control of the remote operator.[4] The robot's rich array of collision-avoidance sensors, originally intended to provide an envelope of protection during autonomous transit, were called into play during manual operation and greatly minimize the possibility of operator error. The operator-commanded velocity and direction of the platform were suitably altered as needed by onboard processors to preclude running into detected obstructions.

From these elementary beginnings in the late 1980's timeframe, there eventually evolved a fairly sophisticated high-level supervisory control strategy (later ported to *Robart III*), wherein the operator could easily control the platform in *reflexive mode* by clicking on special behavioral icons (Figure 4). For example, selecting a "wall" icon to either side of the robot's own icon would cause the platform to enter wall-following mode, maintaining its current lateral offset from the indicated wall using side-looking sonar data. Choosing the "door ahead" icon in front of the robot caused the system to seek out and negotiate a perceived opening dead ahead, while the "door left" or "door right" icons could be used to turn and enter the next doorway or opening encountered on the indicated side of the path. In all cases,



Navigation Control Screen

System Status

Platform is in Motion
Seeking Passage on Left
Battery Voltage: 12.9 Volts

Subsystems

☑ Video ☐ Siren
☑ Lights ☑ Sonar
☐ Flashers ☐ Auxillary

Head Pan Control

Camera Pan Angle: -20 Degrees

Mobility Control

4 4

350.4

0,170

-35.0

Figure 4. Navigation Control Screen for Robart III, showing the high-level driving icons surrounding the *Map Window* (lower left corner). The robot has been instructed to enter the next door encountered on the left.

the on-board driving camera would be automatically pointed in the correct direction at the appropriate time to afford the remote operator the proper field of view for supervising the behavior execution.

Work on *ROBART III* in the mid-to-late 1990's extended this reflexive-teleoperation concept into the realm of sensor-assisted camera and weapon control for indoor tactical systems.[5] The philosophy basically involved first making any two of three controllable elements (i.e., drive control, camera control, and weapon control) slaves to the third, so the human operator only had to deal with one entity. For example, the surveillance camera can be slaved to the weapon, so that the camera looks wherever the operator points the weapon. If either the weapon pan-axis controller or the camera pan-axis controller should approach their respective limits of allowable travel, the robot's drive controller causes the mobility base to rotate in place in the proper direction to restore the necessary range of motion. Alternatively, the weapon could be slaved to the surveillance camera, and so forth. In all cases, final closed-loop control of weapon pan-and-tilt can be provided by video and other on-board sensors, to be discussed further in Section 2.5.

2.2.2 NASA Jet Propulsion Lab (JPL)

FY-03 collaboration between SSC San Diego and JPL resulted in the successful integration of JPL's arc-based free-space reflexive obstacle avoidance/obstacle detection (OD/OA) software[9] onto *Robart III,* using range data from the newly-installed SICK laser. This capability was based on two software components (a behavior arbiter and an OD/OA module) developed under the TMR program. JPL also developed an extensive set of autonomous behaviors, including stair-climbing, visual servoing, GPS waypoint navigation, and retro-traverse. Each behavior outputs to the arbiter a vote-set that expresses the desirability of each of a finite set of driving paths (i.e., constant-curvature arcs). Those arcs that are most desirable according to the goals of that particular behavior are given large votes, while those that are strictly prohibited are vetoed.

By activating the OD/OA module, any autonomous behavior can become “safeguarded.” The basic OD/OA representation is an occupancy grid consisting of rectangular cells which record the presence or absence of obstacles as determined by the SICK's 180-degree scan. Each driving arc is evaluated by examining those cells that intersect that particular arc. Those arcs that do not intersect any occupied cells are marked as being most desirable, while those that do intersect occupied cells are marked as undesirable. The description of each possible driving arc's desirability represents a vote set that is also fed into the behavior arbiter. For each arc, the arbiter then sums all the votes received and selects the driving arc with the most votes and that has not been vetoed by any active behavior to be the executed driving direction.

2.2.3 Idaho National Engineering and Environmental Laboratory (INEEL)

In order to create obstacle avoidance routines that can be used in dynamic unknown environments, INEEL has adopted a behavior-based approach that emphasizes a tight coupling between sensing and action. Each of the sensors on the robot contributes to an array of robot-centric regions to which the robot responds based on fuzzy logic rules that control its translational and rotational velocities. These fuzzy logic rules not only apply to each individual region, but can be triggered by combinations and patterns found within the array of regions. In implementing this scheme, INEEL uses a subsumption architecture, wherein atomistic behaviors such as obstacle avoidance run in parallel with, and can subsume the output of, other reactive behaviors, such as "maneuver-around" and "get-unstuck." Originally developed by Brooks in 1986, the subsumption architecture provides a method for structuring reactive systems from the bottom up using layered sets of rules.[10] This approach can be highly robust when used in unknown or dynamic environments, precisely because it does not depend on any explicit plan of action.

Within INEEL's software control architecture, obstacle avoidance is a bottom-layer behavior, and although it underlies many different reactive and deliberative capabilities, it runs independently from all other behaviors. This independence reduces interference between behaviors and lowers overall complexity. INEEL has also incorporated deliberative behaviors, which function at a level above the reactive behaviors. Once the reactive behaviors are "satisfied," the deliberative behaviors may take control, allowing the robot to exploit a world model in order to support behaviors such as *area search*, *patrol perimeter*, and *follow route*.

INEEL's *guarded motion* (i.e., reflexive teleoperation) capabilities exploit several different sensors (i.e., scanning laser, infrared triangulation, sonar, tactile, inertial, and tilt), fusing available perceptual data into regions that represent the ability of the robot to move safely in a given direction. The algorithm continuously calculates an *event horizon* representing the last possible moment for the collision avoidance behavior to successfully intervene upon goal-based behaviors at the current speed. By calculating this *event horizon* many times each second, the robot can smoothly scale down its velocity as a function of congestion without necessarily fully impeding motion. When a full stop is required, use of the *event horizon* insures that the robot comes to a halt at the same distance from an obstacle regardless of its initial velocity. INEEL's remote-operation studies with human participants have shown that this predictability improves operator trust and overall usability. In terms of portability between robots, the *event horizon* also provides an implicit means to adapt the guarded motion to different deceleration schedules.

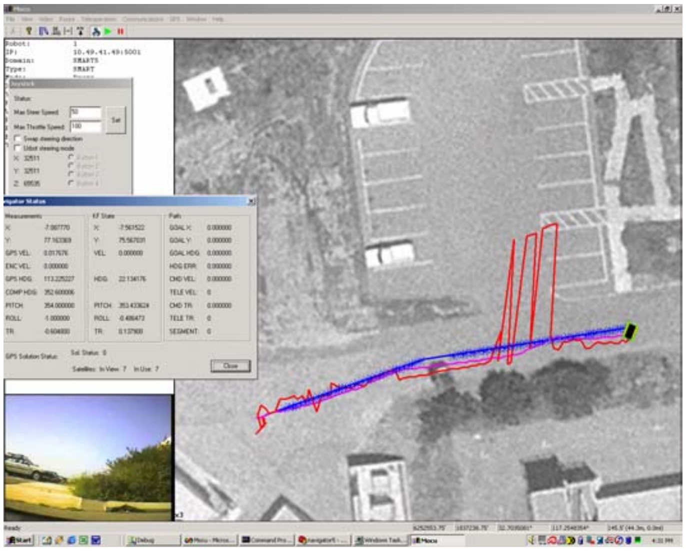
The *guarded motion* capabilities are based on a continuous assessment of available power, as well as the validity of sensor data. The robot provides the user with a holistic measurement of its health and may refuse to undertake a task it cannot safely accomplish. For instance, the robot may elect not to exceed certain speeds if its laser data becomes invalid and forced to rely only on sonar. The end result is that an operator may issue motion commands with near impunity, greatly accelerating the speed and confidence with which the user can accomplish remote tasks. Built on top of the guarded motion, the robot's obstacle avoidance capabilities allow it to circumvent obstacles and extract itself from box canyons.

Currently, these capabilities can be utilized in several different control modes available from INEEL's operator control unit (OCU). In *safe mode*, the robot will only take initiative to protect itself and the environment, but allows the user to otherwise drive the robot. In *shared mode*, the robot handles the low-level navigation, but accepts intermittent input, often at the robot's request, to help guide the robot in general directions. In *autonomous mode*, the robot decides how to carry out high-level tasks such as *follow that target* or *search this area* without any navigational input from the user.

2.3 Waypoint Navigation

A non-differential global positioning system (GPS) waypoint navigation capability was developed at SSC San Diego as part of the Man-Portable Robotic System (MPRS) program,[11] to support the projected need for a compact and inexpensive autonomous navigation capability on small tactical platforms. An OCU was developed to display real-time video from the robot along with an aerial photograph of the robot's operating area (see Figure 5), allowing the operator to select and download path waypoints to the robot for supervised execution.[12]

A non-differential solution was desired due to the nature of tactical operations, which often preclude the luxury of setting up a differential base station. In addressing this goal, an analysis of the various GPS error sources revealed two general cases: 1) high-frequency error components that can be traditionally minimized with a Kalman filter; and 2) low-frequency bias and drift that the Kalman filter cannot readily address. For the latter case, the operator can visually select prominent landmarks (as observed in the robot's video) to be correlated with known geodetic coordinates, generating an offset correction which is sent to the robot to compensate for the long-term bias error. The system combines inputs from drive-motor encoders and a gyro-stabilized MicroStrain *3DMG* electromagnetic compass to compensate for the high-frequency errors using a standard Kalman filter.



Robot: 1
IP: 10.49.41.49:5001
Domain: SMARTS
Type: SMART

Max Steer Speed: 50
Max Throttle Speed: 100
□ Swap steering direction
□ Unbot steering mode
X: 32511
Y: 32511
Z: 65535

Measurements		KF State		Path	
X:	-7.087770	X:	-7.561522	GOAL X:	0.000000
Y:	77.163369	Y:	75.567031	GOAL Y:	0.000000
GPS VEL:	0.017676	VEL:	0.000000	GOAL HDG:	0.000000
ENC VEL:	0.000000			HDG ERR:	0.000000
GPS HDG:	113.225227	HDG:	22.134176	CMD VEL:	0.000000
COMP HDG:	352.600006			TELE VEL:	0
PITCH:	354.000000	PITCH:	353.433624	CMD TR:	0.000000
ROLL:	-1.000000	ROLL:	-0.486473	TELE TR:	0
TR:	-0.604000	TR:	0.137900	SEGMENT:	0

GPS Solution Status: Sol. Status: 0
Satellites: In View: 7 In Use: 7

Figure 5. Screen shot of MOCU graphical user interface.

2.4 Localization/Mapping

GPS has provided outdoor mobile robots with a robust and very accurate absolute localization capability that (assuming sufficient satellite coverage) essentially reduces the exterior navigation problem to a matter of collision avoidance and terrain traversibility. Indoors, however, GPS is not available, and accurate dynamic localization under these conditions has been an ongoing research topic for decades. (Augmented GPS for indoor cell-phone location is on the near horizon, but the resultant accuracy is insufficient for robot localization.)

2.4.1 University of Southern California (USC)

While investigating *Player,*[13] an open-source robot software package developed by USC, SSC San Diego has demonstrated in simulation the Vector Field Histogram (VFH) obstacle-avoidance algorithm[14] and the Adaptive Monte Carlo Localization (AMCL) algorithm,[15] both of which are integrated into *Player*. The VFH algorithm builds a local map of the environment from range sensors and uses a local path-planning algorithm to navigate around obstacles. The AMCL algorithm matches range sensor measurements to an existing map of the environment in order to determine the robot's pose.

Stage is a 2-D robot simulator developed by USC that can simultaneously simulate more than 100 robots.[13] The *Stage* simulator shortens development time by allowing the same software to run without modification on either a simulated robot or an actual robot. Similarly, *Gazebo* is a newly developed 3-D simulator that models the actual rigid-body interactions of more than ten robots in real time. These versatile tools allow for repeatable robot simulation experiments with different control algorithms, for different scenarios with the same control algorithm, and also facilitate the prototyping of new robot and sensor models that previously did not exist.

2.4.2 Stanford University

Collaboration between SSC San Diego and Stanford University is underway to integrate Carnegie Mellon University's *simultaneous localization and mapping (SLAM)* algorithm,[16] distributed as part of the CARMEN open-source software package for unexplored interior structures. The probabilistic algorithm combines an incremental maximum-likelihood pose estimator with Mixture-Monte Carlo Localization. These algorithms are being evaluated in conjunction with SSC San Diego's Autonomous Mobile Communication Relays (AMCR) project,[17] wherein *Robart III* leads a convoy of slave robots that serve as RF repeaters (to be described later in Section 2.6).

2.4.3 SRI International (SRI)

INEEL has been working with SRI to address the fundamental problem of indoor positioning for small unmanned ground vehicles (UGVs). SRI has developed a mapping technique called *Consistent Pose Estimation (CPE)* that efficiently incorporates new laser scan information into a growing map. Within this framework, SRI has addressed the challenging problem of *loop closure*: how to optimally register laser information when the robot returns to an area previously explored. With *CPE*, it is possible to create high-resolution maps and repeatedly execute the accurate path following necessary for high-level deliberative behavior.

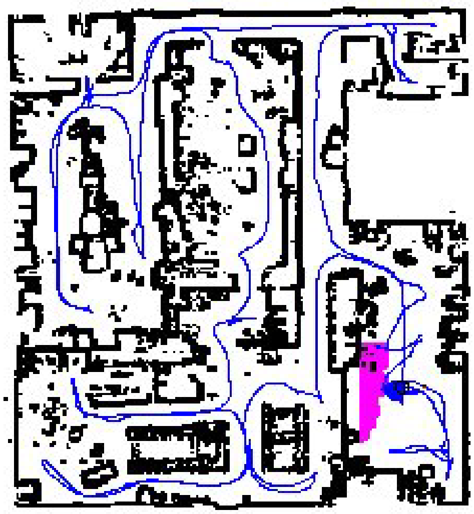


Figure 6. Map of the INEEL robot lab made using SRI's *CPE* algorithm.

CPE is another method for performing *SLAM.* It is based on original work by Lu and Milios,[18] who showed that information from the robot's encoders and laser sensors could be represented as a network of probabilistic constraints linking the successive poses of the robot. The encoders relate one robot pose to the next via dead-reckoning, and the laser scans are matched to give further constraints between robot poses, including constraints for when a robot returns to a previously-visited area.[19] *CPE* provides an efficient means of generating a near-optimal solution to the constraint network and yields high-quality metric maps (Figure 6). The work has been further extended to several interesting applications:

- The multi-robot case, in which several robots explore and communicate to cooperatively map an area. A very challenging condition is that the robots may not know their relative start locations and have to determine them by matching their local maps.[20, 21]
- Very large scale maps covering up to 100,000 scans.[22]

Once a map has been made, it can be used to keep the moving robot localized. SRI has implemented and further developed localization algorithms using a representation of the robot's state space, based on Monte Carlo sampling.[23] Introduced in 1970,[24] Monte Carlo localization (MCL) methods have more recently been applied in the fields of target tracking, computer vision, and robot localization[23, 25] with good results. The Monte Carlo technique inherits the benefits of previously introduced Markovian probability-grid approaches for position estimation[26] and provides an extremely efficient technique for mobile robot localization. One bottleneck in the MCL algorithm is the necessity for checking the posterior probability of each sample against the map, based on the current laser readings. SRI has developed an efficient method for performing this computation, using a correlation technique derived from computer vision algorithms.[27]

2.5 Motion Detection/Target Tracking

The first Distributed Interactive Video Array (DIVA) system developed at SSC-San Diego served as a network of wireless man-portable vision nodes, demonstrating the detection, tracking, and classification of moving targets in a force-protection environment, with autonomous coordination of a UGV response to detected events.[28] Each vision node incorporates an omni-directional camera (Figure 7, top), which provides continuous surveillance over a hemispherical 360-degree field-of-view. To acquire high-resolution views anywhere within its field-of-regard, the omni-camera cues a high-resolution PTZ camera (Figure 7, bottom).

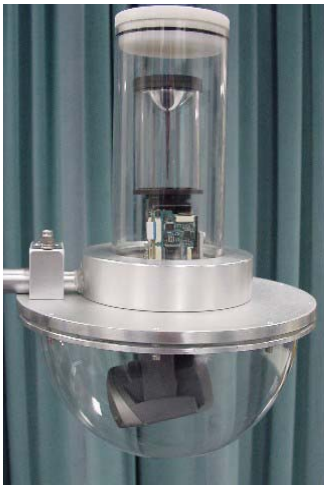


Figure 7. DIVA node.

Based on earlier work done by the Computer Vision & Robotics Research (CVRR) lab at the University of California, San Diego (UCSD), the sensor architecture is designed to be modular and expandable, allowing easy integration of maturing vision technologies. The software architecture incorporates a video compression and transmission module that allows any resource (such as a user, another DIVA node, or a UGV) to simultaneously request the raw video streams in a variety of formats. The computer vision algorithms developed and tested to date include moving object segmentation and tracking of moving objects.[29]

More recently, this DIVA technology has been ported over to ROBART III (Figure 8, top) to provide an advanced vision capability for the robot, as well as a research tool to investigate additional vision algorithms, such as face recognition, 3-D scene construction, and motion detection on the move. Instead of triggering the response of another DIVA station or an investigating UGV (as discussed above), the mobile DIVA implementation is intended to control the non-lethal weapon mounted on the robot (Figure 8, bottom left). The PTZ protocol has been integrated with a two-stage *search-and-engage* algorithm, wherein the vision system first performs a wide-area scan for a pre-taught class of objects, then cues the PTZ camera to zoom in and search for specific “vulnerabilities” associated with that particular target. The non-lethal weapon will be automatically trained accordingly with the aid of a bore-sighted targeting laser, and then fired under operator supervision.

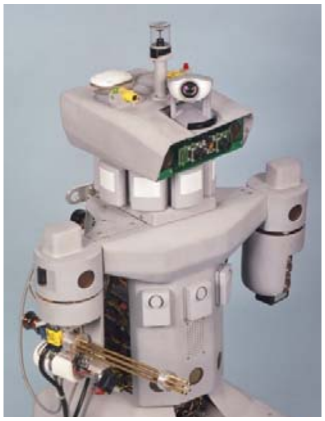


Figure 8. DIVA sensors on head are used to control a non-lethal weapon.

2.6 Cooperative Behaviors

The high-bandwidth communications link on *Robart III* includes a pair of compact ad-hoc networking wireless modems developed jointly with BBN Technologies under the

AMCR project at SSC San Diego.[17] The modems use a proactive link-state protocol that maintains the network links for optimal information transmission and minimal lag. In a demonstration of cooperative behaviors, *Robart III* serves as the lead robot in a convoy of robotic nodes that provides a guaranteed communications path to the remote operator in non-line-of-sight scenarios. Each node uses broadcast messages to obtain link quality information and ensure successful routing of data at all times. Distractions to the robot operator are thus significantly reduced. When any relay node detects that it is no longer needed in the network path, it can request a map generated by ROBART III and use it to reposition itself to be more optimally useful to the system. The map navigation function will also allow for autonomous extraction of the lead robot and its slave relay nodes after mission completion.

3.0 REMAINING TECHNICAL ISSUES

In reexamining Figure 2, it seems readily apparent that each of the identified players is making a synergistic contribution to the collective whole, which in turn should be rather impressive indeed in terms of autonomous functionality once all the individual pieces come together. In reality, however, making it all work in harmony is a bit like mixing apples and oranges. The various developers each have their own preferences and constraints in terms of computer architectures, operating systems, languages, data formats, sensors, embedded hardware, and even power sources.

3.1 Control Architecture

To facilitate integration and ensure the success of ultimate transfer to ongoing programs, the intent is to adapt and standardize on a reconfigurable software framework that can be easily ported from one robot system to another. In an attempt to exploit the best features of what has already been done, a number of existing approaches are being considered, discussed briefly below.

3.1.1 Joint Architecture for Unmanned Systems

The Joint Architecture for Unmanned Systems (JAUS) is a JRP initiative to define and implement an upper-level architecture design for a common interface to a variety of unmanned vehicles, sensors, and munitions.[30] JAUS is component-based, specifying data formats and methods of communication among computing nodes. The JAUS Working Group (made up of members from the U.S. government, industry and academia) defines methods for message passing and standards for component behaviors in order to be independent of technology, computer hardware, operator use, vehicle platform, and mission.

The focus of the JAUS OCU and Payloads Committee (OPC) is to expedite production of more effective and interoperable robotic systems, payloads, and user control devices. Information on performance and effectiveness of current and recommended approaches is collected through a series of experiments, the first of which was held at SSC San Diego in December 2003.[31] A message set implementing defined-transport-layer protocol and dynamic registration was successfully tested. Six different UGVs were wirelessly monitored by their respective OCUs, which were also able to take exclusive control of any other UGV and drive it using teleoperation commands.

3.1.2 Technical Support Working Group Common Architecture

The Technical Support Working Group (TSWG) has pioneered the TSWG Common Architecture: a continually evolving set of mechanical, electrical and logical-level interfaces designed to support interoperability for robotic platforms. While the mechanical connectors are as yet unspecified, the electrical level specifies 24 volts DC, and the logical level specifies Ethernet and JAUS messaging. TSWG has funded the development of this architecture through the Next Generation Explosive Ordnance Disposal Remote Controlled Vehicle (NGEODRCV) program. As part of this program, iRobot Corporation is currently implementing the TSWG Common Architecture under the NeoSuite development effort, which includes the following projects:

- NeoMover - a center-of-gravity-shifting, highly mobile, high performance platform.
- NeoReach - coordinated control of dual manipulators (one light precision; one heavy lift).
- NeoRoll - a wheeled platform for work-horse EOD applications.
- NeoLink - a cable spooler capable of actively deploying and retrieving up to a kilometer of fiber-optic cable.

- NeoComms - tools for better communications and interoperability.

3.1.3 SSC San Diego Small Robotic Technology Architecture

The Small Robotic Technology (SMART) architecture was developed at SSC San Diego for man-portable robots under the MPRS project.[11] SMART fuses concepts from both JAUS as well as the MDARS Multiple Resource Host Architecture (MRHA)[3, 32] and is designed to be efficient in message processing, adaptable to a variety of applications, and modular to support added capabilities in response to new requirements and technologies. Incorporating the JAUS concept of software components as functional *agents*, SMART *agents* are responsible for executing predefined operations (i.e., driving, navigating, communicating), usually representing a drive controller, an operator control unit, or a sensor data collector. Multiple *agents* can execute on a single computer as multiple concurrent processes. A grouping of *agents* that interoperate along control boundaries is collected in a *domain*, usually representing a complete system, such as a robot and its controllers. The SMART architecture supports dynamic discovery of resources (*agents*), using a registration table concept to maintain the current state of the system in terms of available *agents*. Because all *agents* communicate using the same underlying message protocol modeled from the MRHA, SMART systems can dynamically configure themselves to form networks of cooperating *agents* within and across domain boundaries.

3.1.4 INEEL Software Control Architecture

The INEEL software control architecture adapts easily to different robot geometries and to different sensor suites. The entire framework is object-oriented, such that when a new robot platform is available (see Figure 9), the entire software framework (complete with all behaviors and associated autonomous control) can be easily ported, simply by editing a few parameters (i.e., robot length, width, maximum speed) in a script file. Moreover, the system allows the robot to recognize what sensors it has available at any given time and adjust its behavior accordingly.

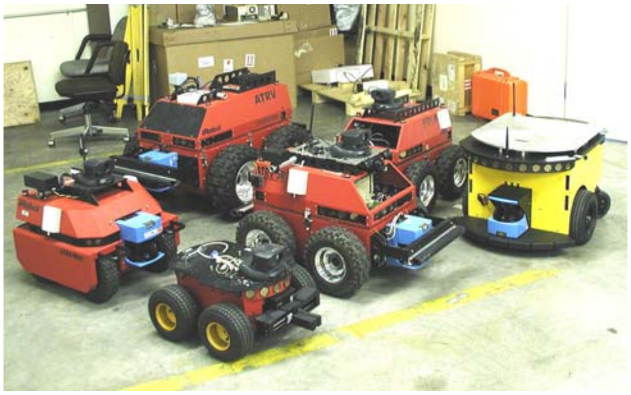


Figure 9: The INEEL control architecture has been installed on all the various robots shown above.

3.1.5 USC Player/Stage

The open-source *Player* project developed by USC is one example of robot software standardization. *Player* is not a control architecture per se, but a robot device server that allows control algorithms to access robot devices in a standard way. *Player* contains drivers for most common laboratory research robots, as well as drivers for different sensors and actuators, and contains high-level algorithms for mapping, localization, path planning, and obstacle avoidance. The control algorithms can be easily ported between robot systems because the devices have a small level of abstraction. Unfortunately, *Player* does have some architectural limitations: 1) *Player* doesn't support real-time operating systems; and 2) *Player* does not prioritize its algorithms, which allows CPU-intensive high-level behavior algorithms to degrade the performance of low-level safety and control algorithms.

3.2 Power

The perception, computational, and actuation schemes required for a supervised autonomous robot will collectively require some considerable power, and providing a reliable, safe, easily renewable energy source that can handle these needs over extended periods of time remains a big problem. Conventional batteries on current man-portable systems last only about four hours, and these teleoperated systems are nowhere near as complex or power hungry. (For example, the run-time for a PackBot equipped with JPL's stereo and laser-based obstacle avoidance systems developed under TMR dropped to a mere 20 minutes.) Solar power has been effectively employed for applications in space, such as the Mars Rovers built by JPL, where speed and endurance have been sacrificed for longevity, but is ill suited to most military applications. Fuel cells offer some near-term promise, particularly those using alcohol as opposed to hydrogen as a fuel, in that the latter cannot be transported on military aircraft due to safety restrictions that ban the requisite high-pressure

(2000 psi) containers. Accordingly, a major technological breakthrough is needed here for the long term, and until then the needs will most likely be met by more sophisticated battery technology, and/or hybrid fossil-fuel/electric systems, with the attendant tradeoffs in capability.

3.3 Sensors

A strong need exists to provide a small, light-weight, non-contact scanning optical range sensor, optimized for the size/weight/power restrictions of man-portable robots (i.e., less than 80 pounds) like the Foster-Miller *TALON*, iRobot *PackBot*, and SSC San Diego URBOT. Current state-of-the art (i.e., the SICK ladar) is typically geared towards automated guided vehicles used in factory automation, and as a result is too heavy (10 pounds) and power-hungry (20 watts) for use on small tactical robots. Under the Technology Transfer Program, conventional algorithms similar to those currently running on larger robotic vehicles are being scaled down and tailored to the available computational resources typically found on such smaller platforms, but there exists no comparably sized sensor to support these algorithms. SSC San Diego and other organizations have successfully implemented GPS waypoint navigation on small robots, but until a suitable sensor is made available, no effective collision avoidance capability to complement this functionality is realistically possible.

The National Center for Defense Robotics (NCDR) and the Penn State Applied Research Laboratory Electro-Optics Center (EOC) have teamed to devise research and development initiatives and fund projects intended to advance the state of sensors for unmanned navigation in general. In recognition of the above technology shortfall, this group has identified a small scanning optical rangefinder as a top priority and is expected to launch some preliminary analysis and assessment efforts in FY-04, while seeking funding for further development. In addition, JPL has developed a second-generation version of their TMR stereo vision system, specifically tailored to the needs of small mobile robots, which is currently being evaluated by SSC San Diego for use on the MPRS program.

4. CONCLUSION

The Small Robot Technology Transfer Program is a JRP-funded effort to evaluate, harvest, and further propagate prior and ongoing innovations from the robotic community, ultimately moving robotics technology towards a single coherent interface and control architecture for use throughout the military. This paper has primarily described the various candidate technologies being evaluated for transition, and alluded to some of the issues being investigated that directly impact integration, portability, and interoperability. Future papers[32] will address the improved functionality and autonomy that will be achieved and subsequent infusion into ongoing development programs.

5. REFERENCES

1. "Joint Robotics Program Master Plan FY 2003*",* Joint Robotics Program*,* <http://www.jointrobotics.com/activities_new/FY2003%20Joint%20Robotics%20Master%20Plan.pdf>.
2. "Critical Technology Roadmap for Robotics and Intelligent Machines," Robotics and Intelligent Machines, <http://www.sandia.gov/isrc/RIMfinal.pdf>.
3. Mobile Detection Assessment Response System program, Space and Naval Warfare Systems Center, San Diego, <http://www.spawar.navy.mil/robots/land/mdars/mdars.html>.
4. Laird, R.T., Everett, H.R., "Reflexive Teleoperated Control," Association For Unmanned Vehicle Systems, 17th Annual Technical Symposium and Exhibition (AUVS '90), pp. 280-292, July-August, 1990.
5. Everett, H.R., Gilbreath, G.A., Ciccimaro, D.A., "An Advanced Telereflexive Tactical Response Robot," Autonomous Robots, Vol. 11, No. 1, Kluwer Academic Publishers, pp. 39-47, July, 2001.
6. Ojeda, L. and Borenstein, J., 2002, "FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain," 2002 IEEE International Conference on Robotics and Automation, pp. 317-322, 11 - 15 May 2002.
7. Ojeda, L., Raju, M., and Borenstein, J. "Experimental Results from UM's Proprioceptive Position Estimation System Installed on CMU's ATRV," Technical Report, University of Michigan, Department of Mechanical Engineering, #03-01, February 19th, 2003**.**
8. Moravec, H.P., Elfes, A., "High resolution maps from wide angle sonar," 1985 IEEE International Conference on Robotics and Automation, St Louis, MO, pp. 116-121, 1985.

9. R. Simmons, L. Henriksen, L. Chrisman, G. Whelan, “Obstacle Avoidance and Safeguarding for a Lunar Rover,” School of Computer Science/Robotics Institute, Carnegie Mellon University.
10. Brooks, R.A., "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, pp. 14-20, 1986.
11. Man-Portable Robotic System program, Space and Naval Warfare Systems Center, San Diego, <http://www.spawar.navy.mil/robots/land/mprs/mprs.html>.
12. Bruch, M.H., Gilbreath, G.A., Muelhauser, J.W. and J.Q. Lum, "Accurate Waypoint Navigation Using Non-differential GPS," AUVSI Unmanned Systems 2002, July 9-11, 2002.
13. B. Gerkey, R.T. Vaughan, and A. Howard, “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems,” 11th Int. Conf. on Advanced Robotics (ICAR), 30 June – 3 July 2003.
14. I. Ulrich and J. Borenstein, “VFH+ Reliable Obstacle Avoidance for Fast Mobile Robots,” IEEE Int. Conf. on Robotics and Automation (ICRA), 1998.
15. D. Fox, “KLD-Sampling: Adaptive Particle Filters,” Advances in Neural Information Processing Systems (NIPS), 3-8 December 2001.
16. S. Thrun, “A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots,” Int. Journal of Robotics Research, 20(5), pp. 335-363.
17. Autonomous Mobile Communication Relays project, Space and Naval Warfare Systems Center, San Diego, <http://www.spawar.navy.mil/robots/research/amrr/amrr.html.>.
18. Lu, F. and E.E. Milios, “Globally Consistent Range Scan Alignment for Environment Mapping,” Autonomous Robots, 4(4), 1997.
19. Gutman, J.S. and K. Konolige, “Incremental Mapping of Large Cyclic Environments,” CIRCA 99, 1999.
20. B. Stewart. J. Ko, D. Fox, and K. Konolige, “The revisiting problem in mobile robot map building: A hierarchical Bayesian approach,” UAI, 2003.
21. D. Fox, J. Ko, B. Stewart, K. Konolige, and B. Limetkai. “Distributed multi-robot mapping,” In A. Schultz, L. Parker, and F. Schneider, editors, Multi-Robot Systems: From Swamrs to Intelligent Automata, Volume II. Kluwer, 2003.
22. K. Konolige, “Large-Scale Map Making,” Submitted to AAAAI04.
23. D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," Sixteenth National Conference on Artificial Intelligence (AAAI'99), July, 1999.
24. J. Handschin. “Monte Carlo Techniques for prediction and filtering of non-linear stochastic processes.” Automatica6, 1970.
25. F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo Localization for Mobile Robots," IEEE International Conference on Robotics and Automation (ICRA99), May, 1999.
26. W. Burgard, D. Fox, D. Hennig, and T. Schmidt. “Estimating the absolute position of a mobile robot using position probability grids,” Thirteenth National Conference on Artificial Intelligence, pp. 896-901, 1996.
27. [Konolige IJCAI].
28. Kogut, G., Blackburn, M., and H.R. Everett, "Using Video Sensor Networks to Command and Control Unmanned Ground Vehicles," AUVSI Unmanned Systems in International Security 2003 (USIS 03), September 9-12, 2003.
29. Everett, H.R., “Robotic Security Systems,” Instrumentation and Measurement Magazine, IEEE, Vol. 6, No. 4, pp. 30 – 34, December, 2003.
30. "Joint Architecture for Unmanned Systems," Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics. <http://www.jauswg.org/>.
31. Mullens, Kathy, "JAUS Experiment Successful," Robotics Update March 2004, Vol 4, No.1, <http://www.spawar.navy.mil/robots/newsletter/RoboticsUpdate_4_1.pdf>.
32. Carroll, D.M., Gilbreath, G.A., Heath-Pastore, T.A.. "Interface Design Document for the Mobile Detection Assessment Response System (MDARS) Multiple Resource Host Architecture (MRHA)," Technical Document 3039, Space and Naval Warfare Systems, San Diego, January, 2004**.**
33. Everett, H.R., Irie, Robert, Pacis, Biagtan E., Kogut, Greg, Farrington, Nathan, “Towards a Warfighters Associate: Eliminating the Operator Control Unit,” <http://www.spawar.navy.mil/robots/pubs/PubsIdx.html>.