Association of Unmanned Vehicle Systems
Washington, DC, 10-12 July, 1995

MDARS Interior Platform

J.M. Holland, A. Martin
Cybermotion, Inc.
115 Sheraton Drive, Salem, VA 24153
POC: 703-562-7626 or jholland@infi.net

R.P. Smurlo, H.R. Everett
Naval Command Control and Ocean Surveillance Center
RDT&E Division 531, San Diego, CA 92152-7383
Point of Contact: 619-553-3668 or smurlo@nosc.mil

1. Abstract

The Mobile Detection Assessment Response System (MDARS) is an automated robotic security and inventory system capable of controlling multiple indoor and/or outdoor autonomous platforms from a single host console. Separate development efforts target: 1) warehouse interiors and 2) outdoor storage areas. The *MDARS Interior* system is designed to run on its own, with each robot patrolling a designated region within a warehouse environment until an "exceptional event" occurs. Typical exceptional events include the detection of an intruder, the robot becoming trapped, or discovery of a fire. When such an event occurs, a guard can intervene from the host console and directly interact with the platform that reported the event.

During patrol the robot performs automated security, environmental, and inventory assessment functions. Each platform is equipped with microwave and passive-infrared motion sensors to detect and a video camera to track potential intruders. A platform-mounted tag-reading device performs inventory assessment functions while the robot is patrolling the warehouse. The tag reader communicates with small radio frequency (RF) transponder tags attached to high-valued or pilferable items. Tag responses pinpoint each item, its location, and time of detection. This information is uploaded (via the command and control console) to an inventory database that can be queried on demand to report misplaced or missing stock.

As the initial *MDARS Interior* implementation involves eight Cybermotion *K2A Navmaster* robots, coordinating and controlling a system of this complexity requires the fusion of data from many sources. This paper describes upgrades to the basic *K2A* platform that were required to support the MDARS requirements (and some commercial applications as well) in the areas of navigation, intrusion detection, and automated inventory assessment, as well as the development of programming constructs to accomplish this integration in a cost-effective fashion.

2. Background

The MDARS program is a joint Army-Navy effort to develop a robotic security and automated inventory assessment capability for use in Department of Defense warehouses and storage sites. The program is managed by the US Army Physical Security Equipment

Management Office, Ft. Belvoir, VA, with the Naval Command, Control and Ocean Surveillance Center (NCCOSC) providing all technical direction and systems integration functions. Near-term objectives are improved effectiveness (with less risk) to a smaller guard force, and significant reduction in the intensive manpower requirements associated with accounting for critical and high-dollar assets (Everett, 1995).

The *MDARS Interior* platform (Figure 1) has successfully demonstrated sustained autonomous navigation within a semi-structured warehouse environment with periodic and unpredictable relocation of oddly-shaped obstacles, and few definitive walls for navigational referencing (Laird, et al., 1993). The ability to detect intruders using passive-infrared and microwave motion detection sensors has been implemented and extensively tested (Smurlo & Everett, 1993). A Broad Agency Announcement (BAA) contract was awarded in 1995 to Cybermotion, Inc., Salem, VA, to improve the probability of detection and integrate an additional capability to perform automated inventory assessment using a tag-reading device and interactive RF transponder tags attached to inventory.

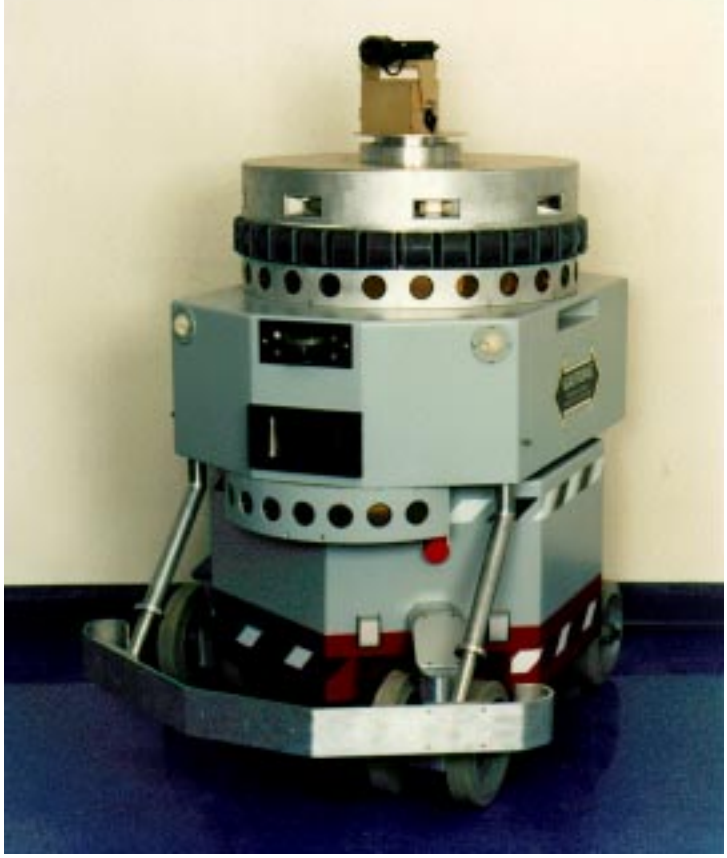

Figure 1. The original *MDARS Interior* prototype was based on the Cybermotion *K2A Navmaster* robotic base, and incorporated a video surveillance camera in conjunction with a staring array of ultrasonic, microwave, and passive infrared motion detectors.

3. Navigation Technology

The most challenging technical risk faced by the *MDARS Interior* program is autonomous navigation. A typical warehouse environment consists of aisles of various widths ranging from 4 feet to over 15 feet, bordered by shelves, storage cages, building supports, or just open areas in which items may be stacked at random. Cybermotion's *K2A* vehicle was developed to navigate primarily from sonar information (Holland, 1993; DeCorte, 1994). By the beginning of MDARS feasibility testing in 1993 at the Camp Elliott warehouse facility in San Diego, CA, this capability was already at a reasonably robust level for offices and other environments affording rich sources of sonar data (walls, columns, railings, etc.). These commercial navigation techniques, however, proved less than adequate in the semi-structured warehouse environment.

The Cybermotion navigational approach uses a layered architecture with dead reckoning at the lowest level. Accumulated navigational errors are corrected according to data extracted from sonar or other sensor readings, using combined concepts of fuzzy logic and acceptance windows. The closer a calculated correction is to the center of its associated *acceptance window*, the more aggressively the correction is used to correct the vehicle's position or heading estimates. A subgroup of the *K2A's* more than 90 instructions is used to trigger this process when pre-specified environmental features are expected to be observable. With names like WALL, HALL, APPROACH, WENDS@, WBEGINS@, and GATE, each instruction activates a particular data collection and filter algorithm and sets its expectation parameters. These instructions precede movement instructions such as RUN or RUNON during which the actual imaging and filtering occur.

The navigational cues or features described by these instructions can be unambiguously discerned by taking a series of readings, plotting them in two dimensional space, and then filtering (curve fitting) for a known attribute characteristic. The simplest example is a WALL, which should yield a series of points lying on a straight line. Just how closely these points fit a straight line (and the position and orientation of that line) define the "quality" of the fit and thus its "believability."

The *window of acceptance* was initially implemented only as a simple value associated with each quality. Although some of these windows were automatically "opened" in response to conditions encountered as the vehicle moved, in general they represented a rigid benchmark that did not accurately reflect reality. This limitation placed the burden on the path programmer to individually adjust these windows in various regions of operation in order to optimize performance. Furthermore, an additional referencing technique was needed that could correct the vehicle's position and heading from sparse data like the observation of structural posts every 20 or 30 feet along the path.

3.1 Sensor Fusion and Arbitration

The *K2A's* navigation software largely evolved instruction by instruction, and any interplay between one navigation technique and another was handled on a case by case basis. If, for example, one algorithm were able to successfully correct the position estimate of the vehicle, it would then have to flag other algorithms that might be working in tandem, announcing the position or heading estimate had changed and collected data would have to be translated or

discarded. As the number of navigation instructions grew, the combinations of potential interactions became very difficult to handle.

Resolving conflicts of this nature necessitated the use of a single *Arbitrator* responsible for all corrections. The software was written such that any algorithm could inform the *Arbitrator* that it was beginning to collect data, and could thereafter determine: 1) if collected data were still valid, 2) if it needed to be translated, or 3) if it should be discarded. As an algorithm arrived at a correction recommendation it passed this information to the *Arbitrator*, which would make (or not make) any given component of the recommended correction.

The *Arbitrator* greatly streamlined navigation programming and made it more object-oriented and general purpose in nature. New sensors and algorithms could now be added in a matter of hours rather than days, with no significant computational load on the vehicle's processor when the algorithms were dormant. As an added advantage, the *Arbitrator* was able to provide important diagnostic data, such as: 1) when any given axis had last been corrected, 2) what algorithm had corrected it, 3) the magnitude and quality of the correction, 4) how far the vehicle had traveled before and after the correction of that axis, and 5) many other nice tidbits of information. This self-assessing diagnostic capability turned out to be invaluable in modeling the navigational uncertainty.

3.2 Optical Re-referencing

Cybermotion had integrated a *lidar* laser ranging system manufactured by Transitions Research, Inc. (Everett, 1995) onto the *K2A* as a navigational referencing option for commercial applications. This horizontal scanning system provides excellent continuous-position updates but requires the vehicle to be in sight of at least two fixed-location retro-reflective targets to make a position and heading correction. Unfortunately, the *lidar* unit also adds a scanning mirror and laser to the overall system MTBF calculation and is of significant cost relative to the rest of the vehicle. For these cost and reliability reasons, a simpler non-scanning optical referencing system was pursued for MDARS use.

The biggest problem with using sonar data to correct odometry from natural landmarks such as structural support posts is the danger of applying erroneous corrections due to target ambiguity. Furthermore, the small amount of data makes meaningful fuzzification rather problematic. In recognition of these problems, Cybermotion and NCCOSC worked together to integrate a simple and inexpensive retro-reflective optical sensor (Banner Engineering Part No. Q85VR3LP) into the existing navigation system. One of these sensors was mounted on each side of the vehicle and integrated with the side-looking sonar such that when the optical sensor detected a small piece of retro-reflective tape *(stripe)* on a post or shelf, this information would be combined with the respective sonar range and the vehicle's heading estimate to predict the X-Y position of the robot (Everett, 1995). This system added only a few hundred dollars to the overall cost while providing a correction of both the lateral and longitudinal axes of the vehicle.

As this optical landmark system was more fully explored it became obvious the vehicle could infer a heading correction from consecutive *stripe* readings. Within a few weeks the *MDARS Interior* robots were navigating the unstructured Camp Elliott environment with an order of magnitude fewer errors (Everett, et al., 1994; Gage, et al., 1995). Since the sonar beam is much wider than the optical beam, however, the acoustical energy can reflect from nearby objects other

than that on which the *stripe* is located. Therefore, it was still possible to confuse the vehicle's navigation solution along the lateral axis.

3.3 Uncertainty Modeling

What was needed was a universal algorithm that could interplay with the *Arbitrator* and existing systems to further improve not only *Stripe* navigation, but all of the other techniques as well. This goal was consistent with the theory of management that says "always try to kill more than one bird with any given stone." To accomplish this, a real-time model of the *K2A Navmaster* dynamics was written to run onboard the vehicle. This model (called the *System Uncertainty Model* or *SUM*) receives all corrections to odometry as well as input from the steering and drive servos. Using this information the *SUM* predicts the worst possible position and heading uncertainty of the vehicle. These uncertainties accumulate as the vehicle executes runs and turns; conversely, as corrections are made the uncertainties diminish.

The vehicle uses this information to automatically and dynamically change the acceptance windows for the various environmental features. The results were even better than expected: the size of many path programs was drastically reduced (i.e. as much as 50 percent) by the process of automatic windowing, and the robustness of the navigation improved by a similar amount.

3.4 States

As part of the restructuring of the navigation algorithms, a set of STATE flags was defined in the blackboard memory of the vehicle, to include PATROLLING, CIRCUMNAVIGATING, PORTING, DOCKING, and CAUTION, among others. Of particular interest is the CAUTION state. This state is initiated in a wide range of navigation problems when uncertainty becomes excessive, or if the vehicle needs to circumnavigate an obstacle. Once triggered, the state remains in effect for a preprogrammed distance. If other problems occur during this time, the distance is simply extended. When the vehicle enters the CAUTION state it slows to half the programmed speed and uses a shortened kickout delay on any servo that begins to stall. The CAUTION state is also fed to the *SUM* where it influences the estimates. Thus in rare cases where the navigation estimate begins to "unlock" from the real world, the platform will slow down and become more “open minded” about the navigation data it may have available.

3.5 Concurrent Processes

Another issue was that of making decisions on the run during path execution: up to this point, all vehicle control programs consisted of a single thread of instructions. Some instructions (i.e., WALL) would in fact elicit a concurrent monitoring, acquisition, or filter task during the next movement instruction, but there was no way to write such concurrent processes. With the exception of preprogrammed behaviors (i.e., stopping if a flame is detected), decisions had to be executed between RUN or RUNON instructions.

A new construct allows simple tests to be performed concurrently during the subsequent RUN. This group of instructions is referred to as a *Do Group*. In the main line program, each *Do Group* instruction is followed by an elaborating set of instructions that will be executed if the specified *Do* conditional is met. In other words, if and when the result of a conditional test is true, the *Do Group* of instructions will be invoked. Several forms of this construct were created:

- DO@ – causes concurrent execution of the associated instructions when the vehicle reaches a designated point on the path.
- DOSTOP@ – causes the vehicle to pull to a halt at the designated point, execute the *Do Group*, and then continue. Execution of this instruction looks like the vehicle has stopped at a node between RUNs, but navigational context is preserved.
- DOWHEN – Specifies a variable to be tested and a comparison to be made (less than, greater than, equal, or not equal) to a constant. The first time the test is true, the *Do Group* executes concurrently and the function is terminated.
- DOWHILE – Specifies a variable exactly like DOWHEN, but the *Do Group* of instructions will continue to execute as long as the condition is valid.

A new concurrent sub-behavior called WATCHing was also added. Similar to a TSR (Terminate and Stay Resident) program on a personal computer, this behavior is induced by executing a WATCHXY instruction and passing the coordinates of the place to be watched. As the robot drives and turns, the surveillance camera pan direction will be continually corrected to keep the camera pointing toward the specified location. This instruction is most useful with fast pan-and-tilt units. A relative pan instruction PANREL has been added as well. Either of these new instructions, or any other camera instruction (PAN, TILT, ZOOM, etc.), may be executed in a DO@, a DOSTOP@, a DOWHEN, or a DOWHILE.

4. Intrusion Detection

Development of the MDARS intrusion detection system began in 1989. As shown in Figure 1, the original hardware incorporated multiple sensor arrays, each array consisting of a different type of one or more sensors (i.e., passive infrared (PIR), ultrasonic, acoustic, microwave, and video motion detection). With the exception of video, each array contained enough fixed sensors (arranged in a circular fashion) to cover the entire 360-degree view around the robot. The video motion detector employed a high-resolution zoom camera mounted on a pan-and-tilt unit that was activated only when other sensors detected a possible intruder, at which point the surveillance camera was automatically panned to the appropriate bearing to either confirm or counter a human presence. This preliminary system served to effectively prove concept feasibility, but a decision was made in December 1993 to switch to the recently introduced Cybermotion *SPI (Security Patrol Instrumentation)* module, developed under a cooperative Research and Development Agreement between Cybermotion and NCCOSC.

Cybermotion now produces for commercial use a scanning sensor array that interfaces to its *CyberGuard SR2 Security Robot* through the *SPI (Security Patrol Instrumentation)* system (Figure 2). The *SPI Scanner* rotates at one revolution per second and contains a vertical array of PIR (passive infrared) detectors, a K-Band microwave transceiver, and an optical flame detector (Everett, 1995). During earlier phases of the *MDARS Interior* program, this array was extensively tested by the Night Vision and Electronic Sensors Directorate, Ft. Belvoir, VA, and determined to be functionally equivalent to "staring" sensors that were much more complex (see again Figure 1).

Figure 2. The prototype Cybermotion *SPI* module shown on the upgraded *MDARS Interior* robot patrolling in the Camp Elliott warehouse in San Diego.

Several areas were identified, however, where the *Scanner* fell short of the performance desired for MDARS, with one of the most prominent shortcomings being mechanical robustness. The prototype *Scanner* was developed on a very limited budget for operation in indoor office environments where the floor surfaces are typically quite smooth. This is generally not the case in warehouses, where adjacent floor slabs may introduce vertical misalignments of up to an inch, and where cracks or holes have appeared in the surface for various reasons over the years. It is not desirable to slow the platform for each of these bumps as this would significantly reduce the effective operating speed. Furthermore, the commercial pan-and-tilt units that interface to the *SPI* for positioning the camera also suffer in this bone-jarring environment.

The early *Scanner* was also found to be less sensitive in detecting targets moving radially than in detecting targets moving in a transverse fashion with respect to the robot. This is quite understandable, since targets moving toward or away from the system simply appear to get larger or smaller, and even this apparent change of size is muted at longer ranges. It should be

mentioned that the *SPI Scanner* has no inherent method of range measurement (such as time of flight), and instead provides range estimates by a complex voting formula based on target size and strength as detected by multiple sensors. Within eight feet of the vehicle the navigation sonar reports targets to the *SPI*, providing very accurate range information.

Under the current BAA contract, Cybermotion is producing an improved system that combines the *Scanner* with a fast pan-and-tilt-mounted surveillance camera. The *Scanner* is secured to the top of a mounting plate and the pan-and-tilt unit is independently secured to the bottom as illustrated in Figure 3, providing optimal fields of view for both systems. Shock absorbers will be included to further improve reliability.

The *Scanner* is also being reconfigured to include: 1) an additional PIR sensor oriented 180 degrees with respect to the existing sensor, 2) a higher-gain microwave antenna (developed by VSE, Inc.), 3) an upgraded CPU that provides the computing power required to run more sophisticated radial tracking algorithms, and 4) an improved slip ring. The combined *Scanner* and pan-and-tilt unit are enclosed in a custom cast-aluminum housing, with provisions to make the system water resistant.

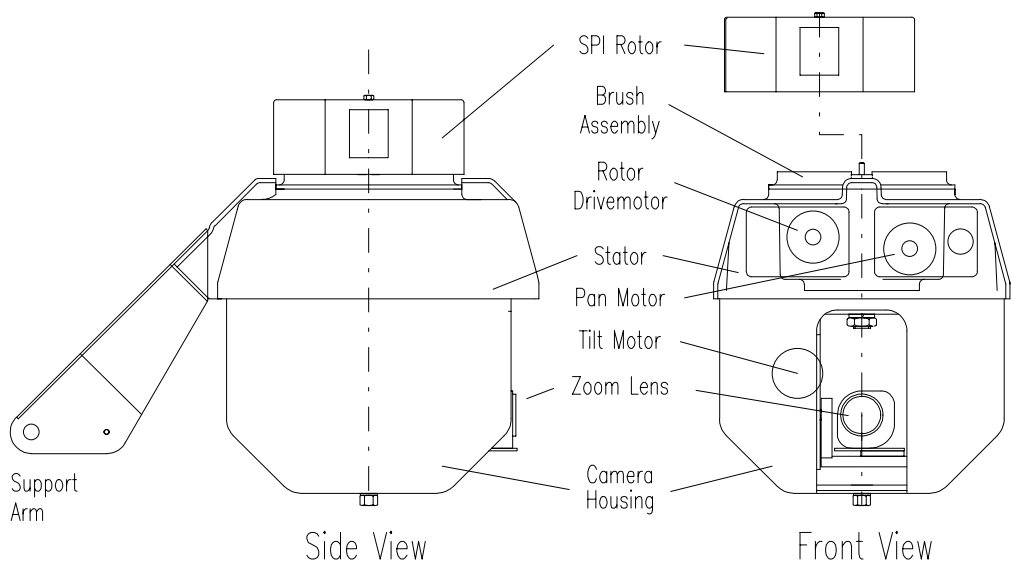

Figure 3. The new *SPI* Module being developed under the BAA contract incorporates an improved Scanner and integrated pan-and-tilt mechanism for the surveillance camera in a single cast housing (Everett, 1995).

5. Automated Inventory Assessment

The *MDARS Interior* platform is also capable of automatically assessing warehouse inventory during routine patrol. Development of the inventory assessment system was initiated in 1991 and included a *Barrier Assessment Module* and a *Tag Database Manager*, both of which were written in the C programming language. While this prototype software demonstrated the feasibility of the product assessment concept, it was severely limited in capability and not suited for operational use. The finalized operational system is being coded in Ada using a database that

supports the *Structured Query Language (SQL)*, and will eventually be capable of interfacing with existing *Depot Supply System (DSS)* inventory management systems.

The robot is equipped with a Savi Technologies, Inc., *Interrogator* that uses RF energy to communicate with interactive transponder tags, and a *Tag Reader Computer* that downloads tag information from the *Interrogator* and transmits the data back to the MDARS host console when requested. The RF tags are placed on high-valued or pilferable items located throughout the warehouse, and are read by the *Interrogator* when the robot is placed in *inventory mode* by the host console. When in *inventory mode* the robot traverses the warehouse stopping at predefined locations where it performs tag-read operations.

A tag-read operation begins with the *Tag Reader Computer* instructing the *Interrogator* to collect the data from all tags within its transmission range (about 100 feet), whereupon each tag replies with a unique tag ID and other programmed information about the item to which it is attached. This data is then downloaded from the *Interrogator* to the *Tag Reader Computer*, where it is buffered until requested by the host console. The data is eventually stored in a database where it can be compared to previously read data to determine if inventory items are missing, have been moved, or were not previously known to the system. The *Automated Inventory Assessment* system and its interrelationship with the entire *Product Assessment System* is described in more detail in the accompanying AUVS paper entitled *MDARS Product Assessment System* (Smurlo, et al., 1995).

6. Future Efforts

The navigation and control requirements of the *MDARS Interior* environment (Figure 4) are significantly different and more challenging than those for conventional office buildings. It has been shown, however, that by integrating fuzzy concepts with concurrent processing techniques and modeling, a system can be produced that will reliably navigate for extended periods of unattended operation. There is no need to install and maintain an active beacon system in the building, and only a minimal requirement for adding inexpensive passive landmarks. Although the *K2A's* onboard processor is presently being upgraded, all of the navigation software described in this paper is currently running on a CMOS *Z80* microprocessor operating at a 4-MHz clock rate, drawing approximately 70 mA. Additional efforts are underway to develop an automatic recovery routine to halt and re-reference the *K2A Navmaster* in the event the platform becomes lost or disoriented, without invoking human assistance.

The intrusion detection system is being upgraded to improve the probability of detection, especially in the case of radial motion, while maintaining or improving the nuisance alarm rate. This includes adding a second PIR array and augmenting the computing power of the CPU to handle the increased sensor data processing in support of more sophisticated detection algorithms. The system is also being ruggedized to better suit the conditions of the warehouse environment.

Figure 4. A typical MDARS warehouse environment offers few natural surfaces amenable to conventional sonar wall-referencing techniques.

The robot's inventory assessment system is being improved in several ways. First, additional electronics included in the currently available *Interrogator* that are not necessary in the MDARS system are being eliminated. The position of the antenna array is also being optimized to ensure that the metal robot housing does not adversely affect the antennae beam pattern. The antennae transmit power is also being optimized. While increasing emitted power improves the ability to read tags, decreasing transmit power facilitates localizing the X-Y position of the tag in the warehouse. Finally, an optimal tag reading strategy is being finalized; considerations include whether or not the robot will need to stop while reading tags, and a more robust protocol for sending the data to the host console. These decisions will be greatly influenced by the evolving requirements of the end user and by the specifics of the new *Depot Supply System (DSS)* interface.

7. References

DeCorte, C., "Robots Train for Security Surveillance," *Access Control*, pp. 37-38, June, 1994.

Everett, H.R., Gage, D.W., Gilbreath, G.A., Laird, R.T., Smurlo, R.P., "Real-World Issues in Warehouse Navigation," SPIE Vol. 2352, Mobile Robots IX, Boston, MA, 2-4 November, 1994.

Everett, H.R., *Sensors for Mobile Robots: Theory and Application*, ISBN 1-56881-048-2, AK Peters, Ltd., Wellesley, MA, 1995.

Gage, D.W., Everett, H.R., Laird, R.T., Heath-Pastore, T.A., "Navigating Multiple Robots in Semi-Structured Environments," ANS 6th Topical Meeting on Robotics and Remote Systems, American Nuclear Society, Monterey, CA, February, 1995.

Holland, J.M., "An Army of Robots Roams the Night," International Robot and Vision Automation Show and Conference, Detroit, MI, pp. 17.1-17.12, April, 1993.

Laird, R.T., Everett, H.R., Gilbreath, G.A.., "A Host Architecture for Multiple Robot Control," ANS Fifth Topical Meeting on Robotics and Remote Handling, Knoxville, TN, April, 1993.

Smurlo, R.P., Everett, H.R., "Intelligent Sensor Fusion for a Mobile Security Robot," *Sensors*, pp. 18-28, June, 1993.

Smurlo, R.P., Laird, R.T., Everett, H.R., Inderieden, R.S., Elaine, S., Jaffee, D.M., "MDARS Product Assessment System," Association of Unmanned Vehicle Systems, 22nd Annual Technical Symposium and Exhibition (AUVS '95), Washington, DC, July, 1995.