CONTROLLING MULTIPLE SECURITY ROBOTS IN A WAREHOUSE ENVIRONMENT

H.R. Everett, G.A. Gilbreath, T.A. Heath-Pastore, R.T. Laird

Naval Command Control and Ocean Surveillance Center
RDT&E Division 531
San Diego, CA 92152-7383

1.0 Abstract

The Naval Command Control and Ocean Surveillance Center (NCCOSC) has developed an architecture to provide coordinated control of multiple autonomous vehicles from a single host console. The Multiple Robot Host Architecture (MRHA) is a distributed multiprocessing system that can be expanded to accommodate as many as 32 robots. The initial application will employ eight Cybermotion K2A Navmaster robots configured as remote security platforms in support of the Mobile Detection Assessment and Response System (MDARS) Program. This paper discusses developmental testing of the MRHA in an operational warehouse environment, with two actual and four simulated robotic platforms.

2.0 Background

MDARS is a joint Army-Navy development effort intended to provide an automated intrusion detection and inventory assessment capability for use in DoD warehouses and storage sites. The program is managed by the Physical Security Equipment Management Office at Ft. Belvoir, VA. The Armament Research Development Engineering Center (ARDEC) at Picatinny Arsenal, NJ, has responsibility for inventory assessment and remote platform integration. The Belvoir Research Development and Engineering Center (BRDEC) at Ft. Belvoir, VA, is charged with security detection and assessment. NCCOSC is providing the command and control architecture and overall technical direction.

Reduction of manpower is a key factor in the MDARS cost benefit analysis. The objective is to field a supervised robotic security and inventory assessment system which basically runs itself until an unusual condition is encountered that requires human intervention. The host architecture must therefore be able to respond in realtime to a variety of exceptional events that may potentially involve several robots simultaneously. Distributed processing allows the command and control problems to be split among multiple resources, and facilitates later expansion via connection of additional processors.

2.1 Host Architecture Overview

A high-level block diagram of the MRHA is presented in figure 1. The number of Planner/ Dispatcher and Operator Station modules resident on the host LAN can be varied in proportion to the number of deployed platforms at a given site. The initial prototype MRHA systems being developed by NCCOSC are configured with a Supervisor, two Planner/Dispatchers, a Product Assessment module, a Link Server module, and one Operator Station for coordinated control of up to eight robotic patrol units.

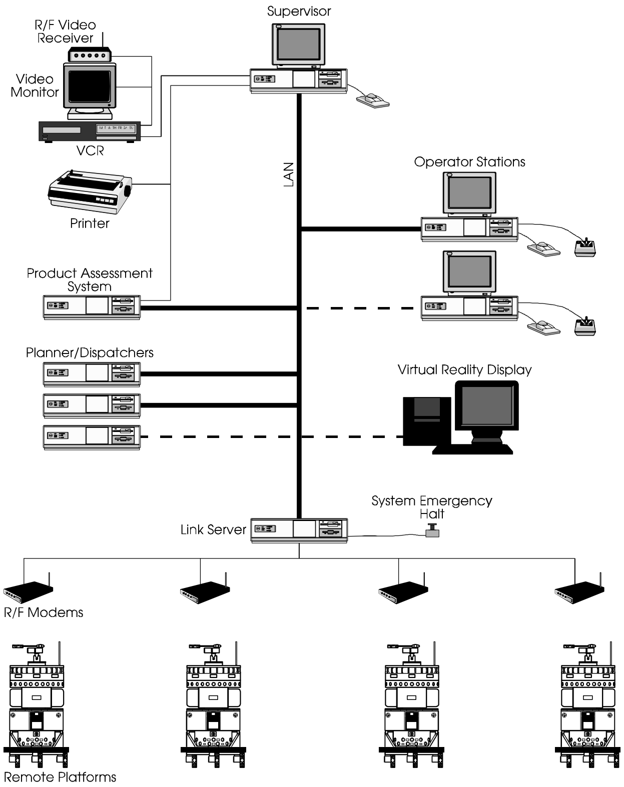


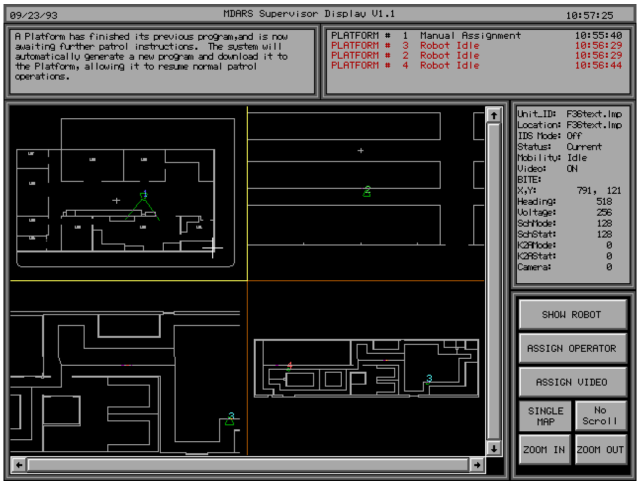
Figure 1. Multiple Robot Host Architecture (MRHA).

2.1.1 Supervisor

The heart of the MRHA is a 486-based industrial PC with a high-resolution display, referred to as the Supervisor. This module maintains a ready representation of the "big picture," scheduling and coordinating the actions of the various platforms, and displaying appropriate status and location information to the guard. Any hands-on control by the guard in response to situations requiring human awareness or intervention (i.e., alarm conditions, teleoperation) takes place at the Operator Station (see below).

Automatic assignment of resources (Planner/ Dispatcher, Operator Station) will be made by the Supervisor in response to exceptional conditions as they arise, based on the information contained in a special data structure that represents the detailed status of all platforms. Such exceptional conditions are referred to as *events*, and typically require either a Planner/Dispatcher, or both a Planner/Dispatcher and an Operator Station. Example *events* include: 1) an intrusion alarm, 2) a lost platform, 3) a failed diagnostic, and, 4) a low battery. The Supervisor will assign the highest priority need to the next available Planner/Dispatcher or Operator Station.

The Supervisor *Map Display Window* will automatically center on the platform listed at the top of the *Event Window*, which in essence represents the highest priority need. The guard can elect to split the screen and display up to four maps at once as shown in figure 2.



```
09/23/93                 MDARS Supervisor Display V1.1                 10:57:25
```

A Platform has finished its previous program,and is now awaiting further patrol instructions. The system will automatically generate a new program and download it to the Platform, allowing it to resume normal patrol operations.

```
PLATFORM #  1  Manual Assignment     10:55:40
PLATFORM #  3  Robot Idle            10:56:29
PLATFORM #  2  Robot Idle            10:56:29
PLATFORM #  4  Robot Idle            10:56:44
```

```
Unit_ID:   F36text.lmp
Location:  F36text.lmp
IDS Mode:  Off
Status:    Current
Mobility:  Idle
Video:     ON
BITE:
X,Y:          791,  121
Heading:          518
Voltage:          256
SchMode:          128
SchStat:          128
K2AMode:            0
K2AStat:            0
Camera:             0
```

Figure 2. MRHA Supervisor Display.

2.1.2 Link Server

All the distributed resources within the host architecture communicate with the various remote platforms via an RF Link Server, which is interfaced to the LAN as shown in figure 1. This 386-based computer acts as a gateway between the LAN and a number of dedicated full-duplex spread-spectrum RF modems operating on non-interfering channels. The various resources (Supervisor, Planner/Dispatcher, Operator Station) on the LAN can thus simultaneously communicate as needed in realtime with their assigned remote platforms. In order to offload from these resources the tedium of constantly requesting status information from the individual remote platforms, the Link Server will periodically poll each platform for critical data such as battery voltage, position, heading, etc. This information is then stored in a blackboard for ready access.

2.1.3 Planner/Dispatcher

Referring again to figure 1, the Supervisor has at its disposal a number of Planner/Dispatcher modules linked over a high-speed Ethernet LAN. These 386-based PCs are mounted in a 19-inch rack adjacent to the display console as shown in figure 3. A globally-shared world model is maintained to provide a realtime collision avoidance capability complementing the Cybermotion *virtual path* navigation scheme employed on the K2A robotic platform. The Planner/Dispatcher modules perform the current virtual path planning functions of the Cybermotion Dispatcher (Holland, et al, 1990), and the *unrestricted path* planning functions of the NCCOSC Planner (Everett, et al, 1990).

The principal function of the Planner/Dispatcher is to plan a path (virtual or unrestricted) and download it to the assigned platform. Under normal conditions, virtual paths are executed until circumstances arise which require deviation from the pre-defined route segment. The most common example would involve an obstacle that blocks the virtual path, whereupon the *unrestricted path planner* is invoked to generate a collision avoidance maneuver.

2.1.4 Operator Station

The Supervisor also has access to one or more Operator Stations via the LAN. These modules are essentially individual control stations that can be

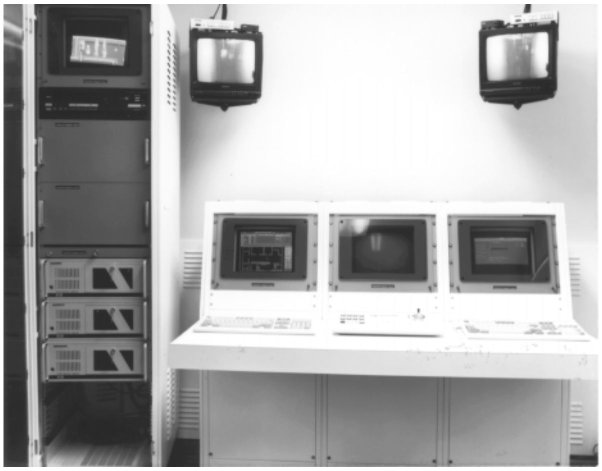
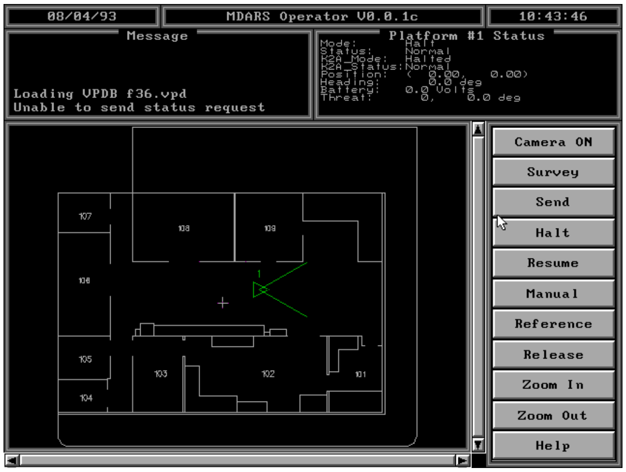


Figure 3. MDARS Control Console.

assigned to a particular platform when the detailed attention of a guard is required. In this fashion, the Supervisor can allocate both computational resources and human resources to address the various situations which arise in the control of a number of remote platforms.

The Operator Station allows a security guard to directly influence the actions of an individual platform, with hands-on control of destination, movement, mode of operation, and camera functions. An Operator Station is automatically assigned by the Supervisor if an *exceptional event* occurs requiring human awareness or intervention. In addition, the guard can manually assign an Operator Station to: 1) teleoperate a platform when necessary, 2) perform non-automatic path planning operations (with the aid of a Planner), 3) place a platform in *Surveillance Mode* for intruder detection, 4) control an onboard video camera, and, 5) assess a potential disturbance.

The Supervisor and Operator Stations have been similarly configured to provide consistent, user-friendly visual displays. Both modules support a point-and-choose menu interface for guard-selectable options, commands, icons, and navigational waypoints. A row of command-option menu buttons are located on the right side of the Operator Station display screen as shown in figure 4. Telereflexive operation of the platform (Everett & Laird, 1990) and camera pan and tilt functions will be controlled by a specialized joystick.



```
08/04/93        MDARS Operator V0.0.1c        10:43:46

Message
Loading VPDB f36.vpd
Unable to send status request

Platform #1 Status
Mode:       Halt
Status:     Normal
K2A_Mode:   Halted
K2A_Status:Normal
Position:   (   0.00,    0.00)
Heading:        0.0 deg
Battery:    0.0 Volts
Threat:        0,     0.0 deg
```

Figure 4. MRHA Operator Station Display.

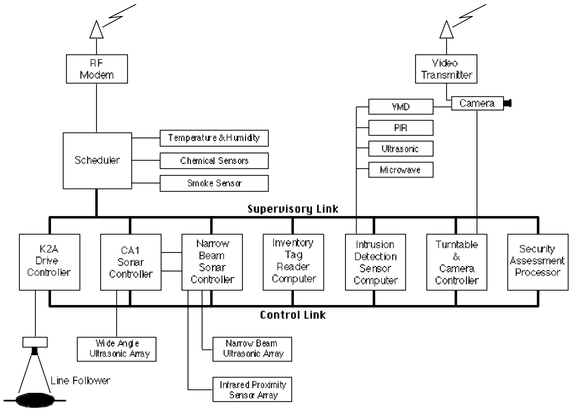
2.1.5 Product Assessment System

The Product Assessment System is responsible for receiving actual inventory data from an interactive tag reader, and then correlating results with a database representing the supposed inventory. The robotic platforms are each equipped with a Savi tag interrogator that communicates with special RF transponder tags attached to the high-value or sensitive items to be monitored. The Savi tags respond with a unique identification code, which is then location-stamped and buffered in memory by the controlling microprocessors onboard the individual robots.

The buffer contents are periodically uploaded by the Link Server and passed via the host LAN to the Product Assessment System. The Product Assessment System compares each tag ID with information recorded in the database to determine if an item is mislocated or missing altogether. It also flags any detected tag IDs which are not represented in the database. A discrepancy report is generated at the end of each 24-hour shift.

2.2 Patrol Unit Overview

A block diagram of the platform architecture is presented in figure 5. Each robot is equipped with the Cybermotion SPI security sensor module providing full 360-degree intrusion detection coverage, augmented by a video motion detector. The high-resolution video surveillance camera is automatically positioned to view the scene of any suspected disturbance by a computer-controlled pan-and-tilt unit, with live video relayed over a dedicated RF link to the guard console.



RF Modem
Video Transmitter
Camera
VMD
PIR
Ultrasonic
Microwave
Scheduler
Temperature & Humidity
Chemical Sensors
Smoke Sensor
Supervisory Link
K2A Drive Controller
CA1 Sonar Controller
Narrow Beam Sonar Controller
Inventory Tag Reader Computer
Intrusion Detection Sensor Computer
Turntable & Camera Controller
Security Assessment Processor
Control Link
Wide Angle Ultrasonic Array
Narrow Beam Ultrasonic Array
Infrared Proximity Sensor Array
Line Follower

Figure 5. Remote Platform Architecture.

An intelligent security assessment algorithm is employed to maximize the probability of detection while at the same time filtering out nuisance alarms. This onboard pre-processing of security-related data relieves the Supervisor of any security assessment responsibility, which is a key element of the control philosophy. The Supervisor basically gets involved only after an intruder has been detected and confirmed by the software onboard the robot.

To satisfy the immediate need for numerous remote platforms required to test the multiple robot control paradigm, a robot simulator was developed that implements the communications protocol of the Cybermotion K2A. The simulator hardware consists of a PC/AT-compatible laptop computer serially interfaced to an R/F modem of the type employed on the robots. The simulators are able to emulate specific platform functions, such as path downloading, decoding, and execution. During current development and test activities at Camp Elliott, four simulators are used in conjunction with the two Cybermotion K2A platforms.

3.0 The Navigational Problem

A wide variety of techniques have been developed over the years for the autonomous navigation of indoor vehicles. For purposes of this discussion, these may be grouped into three general categories: (1) guidepath following, (2) unrestricted path planning, and (3) virtual path navigation. Each of these guidance methods has advantages and disadvantages which determine its appropriate applications. MDARS seeks to integrate the desired features of all three techniques into a robust navigational package better able to cope with the varied demands of realworld operation.

3.1 Fixed Guidepaths

The simplest form of autonomous control involves a navigational control loop which reflexively reacts to the sensed position of an external guiding reference. Industrial vehicles have been guided by physical paths including slots, buried wires, optical stripes, and other methods for almost thirty years. Such automated guided vehicles (AGVs) have found extensive use in factories and warehouses for material transfer, in modern office scenarios for material and mail pickup and delivery, and in hospitals for delivery of meals and supplies to nursing stations.

The most common guidepath following schemes in use today involve some type of stripe or wire guidepath permanently installed on the floor of the operating area. Specialized sensors on the front of the platform are used to servo-control the steering mechanism, causing the vehicle to follow the intended route. These guidance schemes can be divided into three general categories: (1) those which sense and follow the AF or RF field from a closed-loop wire embedded in the floor, (2) those which sense and follow a magnetic tape in or on the floor, and, (3) those which optically sense and follow some type of stripe affixed to the floor surface.

The fundamental disadvantages of guidepath control are the cost of path installation and maintenance, and the subsequent lack of flexibility: a vehicle cannot be commanded to go to a new location unless the guidepath is first modified. This is a significant factor in the event of changes to product flow lines in assembly plants, or in the case of a security robot which must investigate a potential break-in at a designated remote location.

3.2 Unrestricted Paths

The term *unrestricted path planning* implies the ability of a free-roaming platform to travel anywhere so desired, subject to nominal considerations of terrain traversability. Most of the path planning work to date has been done on the premise that the ultimate navigation system would be capable of mapping out its environment with sensors, and then planning routes accordingly. While such systems have a great deal of appeal, they encounter several difficulties in practice.

The most significant problem associated with building a world model is the poor quality of most sensor data. There are many choices available to the designer of such a navigation system, but in every case good data is expensive. In practice, reflective sensors (ultrasonic rangefinders and near-infrared proximity detectors) have predominated (Everett, et al, 1992). Such sensors are subject to the problems of noise, specular and secondary reflections, and signal absorption to one extent or another. Furthermore, the perceived position of objects viewed from different locations will be distorted by any errors in the vehicle's dead reckoning accuracy as it moves between vantage points. Template matching of sensor data can thus be very difficult (Holland, et al, 1990).

Providing an autonomous capability to support non-restricted motion involves the implementation of an appropriate map representation, the acquisition of information regarding ranges and bearings to nearby objects, and the subsequent interpretation of that data in building and maintaining the world model.

3.2.1 Selecting a Map Representation

Several different map representation schemes have been devised, including polyhedral objects (Lozano-Perez, 1979), generalized cones (Brooks, 1983), certainty grids (Moravec, 1987), and quadtrees (Fryxell, 1988). The simplest scheme is a two-dimensional array of cells; each cell corresponds to a square of fixed size in the region being mapped. The map can be accessed and updated quickly, which is extremely important for realtime operation. Free space is indicated with a cell value of zero; a nonzero cell value denotes an obstruction.

The most compact form of a cell map consists of one bit per cell, and thus indicates only the presence or absence of an object. By using multiple bits per cell, additional descriptive information can be represented in the map, such as identification of structural walls and doorways. In addition, the probability of a given square being occupied can be easily encoded, which turns the map into a form of certainty grid (Moravec, 1987). This statistical approach is especially useful when the precise location of objects is unknown.

3.2.2 Unrestricted Path Planning Algorithm

A wide variety of path planning techniques have been developed over the years, each having various advantages and disadvantages. The actual planner employed in a given application is often dictated by which world modeling scheme has been chosen. For a certainty grid representation, the most straightforward planner is derived from the Lee maze router (Lee, 1961), with the cell coding enhancements suggested by Rubin (1974). The basic search algorithm begins by "expanding" the initial cell corresponding to the robot's current position in the floor map, (i.e., each unoccupied neighbor cell is added to the expansion list). Then each cell on the expansion list is expanded, the process continuing until the destination cell is placed on the expansion list, or the list becomes empty, in which case no path exists. This algorithm will find the minimum distance path from the source to the destination.

The minimal distance path, however, is not necessarily the "best" path. Sometimes it is more desirable to minimize the number of turns, or to maximize the distance from obstacles, for example. The search strategy can be altered accordingly by assigning a cost to each cell prior to adding it to the expansion list; only the minimum-cost cells are then expanded. This is known in the literature as an A* search (Winston, 1984), and was adopted by NCCOSC for use in this work (Everett, et al, 1990) due to the inherent flexibility of the associated cost function.

3.2.3 The Problem - Dead Reckoning Errors

Appropriate sensors must be coupled with some type of *world modeling* capability representing the relative/absolute locations of objects to support intelligent movement in unstructured environments. The accuracy of this model, which is refined in a continuous fashion as the platform moves about its workspace, is directly dependent upon the validity of the platform's perceived location and orientation. Accumulated dead-reckoning errors soon render the information entered into the model invalid since the absolute reference point for data acquired relative to the platform is incorrect. As the accuracy of the model degrades, the ability of the platform to successfully navigate and avoid collisions diminishes rapidly, until it fails altogether.

3.3 Virtual Paths

The virtual path concept was developed by Cybermotion to provide a routine mechanism for correcting dead reckoning errors in the normal course of path execution. Each desired route is pre-programmed by a technician to take advantage of any available environmental cues that the robot can recognize with its sensors. Each path begins and ends on named *virtual nodes* as shown in figure 6. A database is constructed that associates each virtual node with one or more *virtual path segments* entering or leaving that location. The Planner/Dispatcher uses this database to link several discrete virtual path segments together to form a complete route from any given node to any other node.

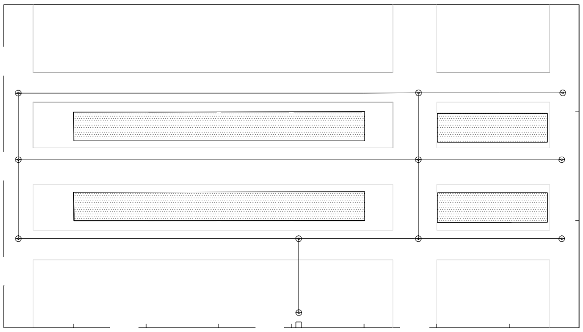


Figure 6. Map of Camp Elliott Showing Virtual Paths.

Correction of dead reckoning errors during run time is most commonly accomplished by indicating in the virtual path program (at the time of installation) one or both of the following cues: 1) the distance to the wall (or walls) on the left (and/or right) side of the robot, and, 2) the expected standoff from a wall in front of the robot at the completion of the route segment.

In the *wall-following* mode, the robot uses its lateral sonars to maintain the specified offset from the indicated wall while traversing the distance between two given points. Knowing the starting and ending locations of the virtual path segment, the robot can correct its heading as well as the lateral axis position coordinate. When approaching a wall, the robot uses the forward sonars to measure its actual distance from the wall. By comparing the observed range with the anticipated distance specified in the program, and knowing the X-Y coordinate of where it should be when it stops, the robot can correct the longitudinal axis of its dead-reckoned position. When *wall-following* and *wall-approach* are used together, both the X and Y coordinates can be corrected, in addition to heading.

Although the virtual path approach does not provide the flexibility of unrestricted path planning, it can be implemented with relatively low-cost sensor and computing hardware. Many practical applications can be addressed in this fashion, but the fundamental deficiency is the lack of collision avoidance capability. If an obstacle blocks a virtual path route segment, the platform must halt and wait for assistance.

3.4 MDARS Hybrid Navigation Scheme

The navigation scheme employed on MDARS is basically an integration of the Cybermotion *Dispatcher* and the NCCOSC *Planner*, which were separately employed on an earlier prototype to generate virtual paths and unrestricted paths, respectively. Integration of these two planning algorithms was accomplished in FY-92 under a Cooperative Research and Development Agreement with Cybermotion, giving rise to the term *Planner/Dispatcher*.

The hybrid navigational scheme exploits the inherent re-referencing ability of virtual paths, while retaining the free-roaming flexibility of unrestricted path planner control. Under normal conditions, the robotic platform traverses virtual paths, which are kept relatively obstacle free, at significantly higher speeds than typically possible in the unrestricted path mode. In the event of an impending collision, the platform is halted, and the unrestricted path planner generates an avoidance maneuver around the obstacle to the desired *virtual point* destination.

Three guidepath following strategies were investigated for incorporation into the hybrid navigation concept to support warehouse navigation. The location of a guidepath segment would simply be encoded into the virtual path database, the idea being to look down for a pathway instead of to either side for a wall. Finite path sections could then be strategically placed along troublesome route segments, as opposed to throughout the entire site.

The first of these path-following schemes was originally developed for AGVs by Litton Corporation, and employed a chemical stripe that glowed brightly when irradiated by an ultraviolet source. Disadvantages (from an MDARS perspective) included excessive power consumption, insufficient clearance between the floor and sensor, interference from ambient lighting, and periodic lamp failure. A alternative retro-reflective near-infrared design was developed by NCCOSC to overcome these concerns, but performance ultimately suffered from abrasive wear on the tape guidepath.

To address the problem of path degradation, a Hall-effect guidepath sensor developed by Apogee Robotics was purchased for evaluation, the intent being to bury the flexible magnetic tape in a saw kerf cut into the floor. This attempt proved futile as well due to limited

(2-inch) sensor standoff, and the constantly changing magnetic signature of the K2A platform, which is an artifact of synchro-drive steering. As a consequence, the guidepath option has been indefinitely suspended until a practical solution compatible with the needs of a warehouse environment is found.

4.0 Warehouse Navigation

During the phased development of MDARS, three general classes of autonomous navigation are being addressed:

- Structured navigation - operation in a conventional walled office or laboratory environment.
- Semi-structured navigation - operation in a warehouse environment, with some structured order in the form of shelving (or inventory) forming permanent aisles.
- Unstructured navigation - operation in an open warehouse environment with no definitive aisles.

The MDARS cost-benefit analysis indicates the vast majority of operational sites visited to date fit the category of semi-structured, with fixed shelving but few or no unobstructed walls available for re-referencing. The remainder of this section discusses this class of navigation in the actual Camp Elliott warehouse environment. Camp Elliott is a government storage facility adjacent to Miramar Naval Air Station in San Diego, CA.

4.1 Random Patrols

The Supervisor automatically assigns idle platforms to a Planner/Dispatcher for random patrols of the warehouse environment. The Planner/Dispatcher generates a random virtual path patrol route, and downloads it via the Link Server to the assigned platform. This onboard K2A program contains instructions which cause the platform to halt and enter Surveillance Mode at randomly chosen virtual points along the path. When a platform arrives at its commanded destination, it reports back an *Idle Mode* status to the Supervisor. The Supervisor then reassigns a Planner/Dispatcher, which generates and downloads a new patrol route.

4.2 Obstacle Avoidance

Potential obstructions in the vicinity of the robot are detected by an array of Polaroid ultrasonic ranging sensors and Banner diffuse-mode near-infrared proximity sensors mounted on the front of the turret of the K2A platform as shown in figure 7.

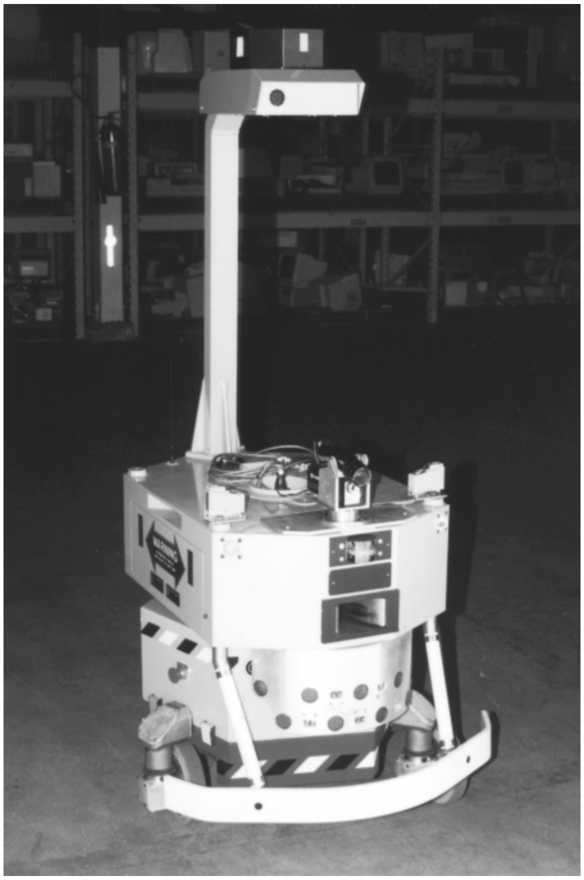


Figure 7. MDARS Remote Platform.

Range and bearing information collected over the last 10 feet of travel are stored in a circular buffer maintained by the Narrow-Beam Sonar Controller (figure 5). If a threatening object enters the protected envelope, the platform is halted with a *Blocked* status, which is in turn detected through routine polling by the Link Server.

Once the Supervisor has been made aware that a platform is blocked by an obstacle, it assigns a Planner/Dispatcher to resolve the problem. The historical sonar and proximity sensor data are uploaded from the platform after-the-fact via the Link Server, and used to update the world model for path planning purposes. The Planner/Dispatcher downloads the resultant avoidance maneuver to the platform for execution.

4.3 Navigational Re-Referencing

The big challenge posed by semi-structured warehouse navigation is nulling out accumulated dead-reckoning errors without any definitive walls. The wall-following and wall-approach instructions provided by Cybermotion are powerful enough in themselves to satisfy on-the-fly re-referencing needs in most structured environments. In a semi-structured warehouse environment walls are generally not available, but racks of storage shelves are usually abundant. If the inventory items do not significantly protrude over the lip of the shelf, the shelf itself can be treated as a wall and imaged by the side-looking sonars. Problems have been observed at Camp Elliott, however, in that some shelf sections are actually misaligned several degrees with respect to the path axis.

Experience has shown the primary source of accumulated position errors is erroneous heading information. During an extended dead-reckoning run, perceived position along the longitudinal path axis is usually quite good. Positional uncertainty perpendicular to the direction of motion, on the other hand, degrades much more rapidly due to minor inaccuracies in perceived heading. The problem manifests itself when the robot attempts to follow a rack or wall: if the measured range is too far (or too close), it will be treated as an anomalous sensor reading, and subsequently ignored. To compensate, an ultrasonic range "sniff" can be performed at the start of path segments where this problem is known to occur. This corrective action is accomplished by programming a very short *approach* instruction at the beginning of the virtual path, perpendicular to the direction of intended travel, on a distinctive target (i.e., a post, wall, or rack).

In most proposed depot locations, the MDARS robot will be patrolling multiple bays (typically 300’ by 300’) within the same warehouse. The robot traverses from one bay to the next through large fire doors that provide operational access for forktrucks and warehouse personnel. These definitive openings offer excellent on-the-fly opportunities to re-reference the X and Y axes of the robot through use of the Cybermotion *gate* instruction. Information is encoded into the virtual path telling the robot where to expect the doorway, the width of the opening, and path displacement from doorway centerline.

When approaching the programmed location, the robot begins to closely monitor the side sonars, looking for range readings that match the pre-specified characteristics of the gate. If a match is found, the vehicle’s position along the length of the path is updated by substituting the longitudinal coordinate of the gate. Actual lateral position is determined by comparing the measured offset within the opening to what was expected. Unfortunately, however, no heading updates are directly obtained.

To augment the traditional *wall following*, *approach*, and *gate* instructions, an active re-referencing technique called *lateral post detection* has recently been incorporated. Short vertical strips of 1-inch retro-reflective tape are placed on various immobile objects (usually posts) on either side of a virtual path segment. The exact X-Y locations of these tape markers are encoded into the virtual path.

A pair of circularly-polarized Banner Q85VR3LP retro-reflective sensors are mounted on the turret of the K2A robot, facing outward as shown in figure 7. When the robot travels down a typical aisle as illustrated in figure 8, the Banner sensors respond to the presence of markers on either side, triggering a *snapshot* virtual path instruction that records the current side-sonar range values. The longitudinal position of the platform is updated to the known marker coordinate, while lateral position is inferred from the sonar data, assuming both conditions fall within specified tolerances.

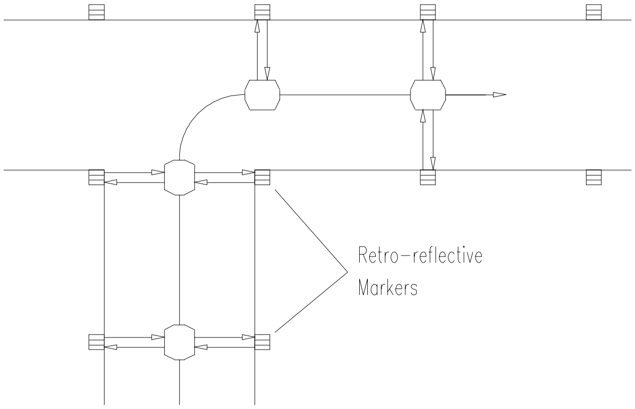


Figure 8. Lateral post detection referencing technique.

5.0 Camp Elliott Site Preparation

An 8’ x 20’ weatherized equipment shelter is used to house the host console electronics. The shelter is located adjacent to the warehouse as shown in figure 9, and is intended to represent a typical MDARS guard station.

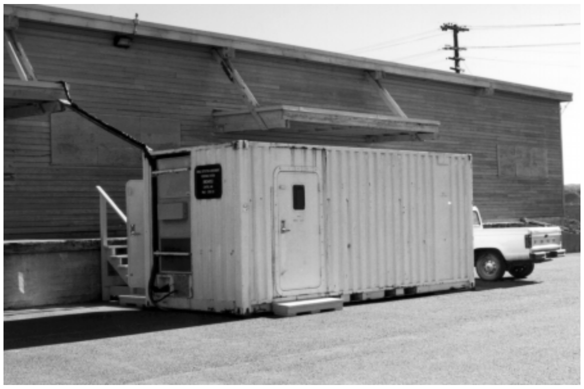


Figure 9. MDARS Camp Elliott control van.

An ARLAN 100-series R/F modem network is used to communicate simultaneously with multiple robots. The ARLAN 110 network controller connects to multiple daisy-chained ARLAN 010 transceivers located throughout the warehouse. Individual RS-232 serial channels are routed from the host console to the 110 controller within the warehouse.

If an exceptional event requiring human awareness or intervention should arise during non-attended operation, the Link Server will dial an outside line to alert a designated individual via appropriate speech output. This feature was implemented using a Versicom PROCAM development toolkit for creating custom automated telephone voice messages.

In order to respond to the automated operator intervention telephone calls, a remote control capability was implemented using the commercially available telecommunications package pcAnywhere by SYMANTEC. A high-speed modem is interfaced to each of the computers at the console via a code-activated switch. Users can dial-in and connect to any of the MRHA processors over the phone line and take control of the system just as though they were actually present at the remote site.

To aide in monitoring the movements of the robots from the host console, several CCD surveillance cameras were installed within the warehouse. The cameras are positioned strategically along and across aisles such that nearly the entire warehouse bay can be viewed. Since only two of eight video signals are returned to the console from the warehouse it is necessary to switch cameras in order to track the robots’ movements.

The conical field-of-view for each camera is reduced to its projection on the X-Y floorplan, then further simplified to a rectangle to take into account restrictions imposed by the shelving on either side of the aisles. The positional coordinates of a pre-specified robot are repeatedly checked by the Link Server against each of these rectangular coverage areas to determine which camera holds the platform within its field-of-view. A digital command corresponding to the camera ID is then output over an RS-232 serial link to a one-of-eight video multiplexor, thus selecting the appropriate source for display on a monitor in the control van.

6.0 Status and Future Work

With the exception of the Product Assessment module, the hardware and software described in this paper is installed and functioning in the Camp Elliott warehouse facility. Savi tag reader hardware is currently being evaluated at ARDEC and should be installed at Elliott by March 1994; the first actual Product Assessment feasibility demonstration is scheduled for July 1994.

Version 1.1 of the MRHA was distributed to designated recipients in January of 1994, and several enhancements to the system are in progress or planned for the near future. These include automatic wall/shelf following during unrestricted planning operations, a scripting capability that allows more elaborate "canned" paths (useful for inventory monitoring), coordinated movement of multiple robots sharing the same operating environment, and improved methods of navigational referencing.

To enhance the effectiveness of the remote control capability described in section 5.0, a video capture and transmission feature will be added to the host console. This will allow video from one of several sources including the camera on-board the robot to be captured and transmitted to a remote user over the same phone line used for data and voice communications.

7.0 References

Brooks, R.A., "Solving the Find-Path Problem by Good Representation of Free Space," IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-13, No. 3, 1983.

Everett, H.R., Gilbreath, G.A., Tran, T.T., Nieusma, J.M., "Modeling the Environment of a Mobile Security Robot," Technical Document 1835, Naval Command Control and Ocean Surveillance Center, San Diego, CA, 1990.

Everett, H.R., Laird, R.T., "Reflexive Teleoperated Control," Association for Unmanned Vehicles Symposium, Dayton, OH, Jul-Aug, 1990.

Everett, H.R., DeMuth, D., Stitz, H., "Survey of Collision Avoidance and Ranging Sensors for Mobile Robots," Technical Report 1194, Revision 1, Naval Command Control and Ocean Surveillance Center, San Diego, CA, December, 1992.

Holland, J., Everett, H.R., Gilbreath, G.A., "Hybrid Navigational Control Scheme for Autonomous Platforms," SPIE Mobile Robots V, 1990.

Fryxell, R.C., "Navigation Planning Using Quadtrees," SPIE Mobile Robots II, W.J. Wolfe, W.H. Chun, Eds., Proc. SPIE 852, pp. 256-261, 1988.

Lee, C.Y., "An Algorithm for Path Connections and its Applications," IRE Transactions on Electronic Computers, Vol. EC-10, September, pp. 346-365, 1961.

Lozano-Perez, T., and M.A. Wesley, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Communications of the ACM, Vol. 22, No. 10, pp.560-570, 1979.

Moravec, H.P., "Certainty Grids for Mobile Robots," Proceedings of the Workshop on Space Telerobotics, JPL, Pasadena, CA, January, 1987.

Rubin, F. "The Lee Path Connection Algorithm," IEEE Transactions on Computers, Vol. C-23, No. 9, September, pp. 907-914, 1974.

Winston, P.H., *Artificial Intelligence*, Addison-Wesley, Reading, MA, 1984.