Proceedings of 2001 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
July 29 - August 1, 2001, Banff, Alberta, Canada

Supervisory Control of Mobile Robots using Sensory EgoSphere

K. Kawamura, *Senior Member, IEEE*, R. A. Peters II, C. Johnson, P. Nilas and S. Thongchai
School of Engineering, Vanderbilt University, Nashville, TN 37235
`kawamura@vuse.vanderbilt.edu`

Abstract

This paper describes the supervisory control of mobile robots using a biologically inspired short-term memory structure called a Sensory EgoSphere. The EgoSphere is implemented as a virtual geodesic dome upon which sensory data from the surroundings of the robot are written. The EgoSphere is managed by a distinct agent in a distributed agent-based robot control architecture called the Intelligent Machine Architecture. The paper also describes a human-robot interface and a testbed for evaluating the control system.

Index Terms: Sensory EgoSphere, agent-based system, supervisory control, Intelligent Machine Architecture, mobile robots

1 Introduction

The design and operation of user-centric graphical user interfaces is key to supervisory control of mobile robots. The Center for Intelligent Systems (CIS) at Vanderbilt University is conducting research and development in a robust graphical user interface (GUI) involving the human operator and mobile robots under a DARPA sponsorship. Control and communications among the human operator and robots are embedded within a parallel, distributed robot control architecture called the Intelligent Machine Architecture (IMA) [8, 10]. When the robot needs assistance, the human operator assesses the situation and provides help through the Sensory EgoSphere (SES).

2 The Intelligent Machine Architecture (IMA)

The Intelligent Machine Architecture (IMA) is an agent-based software architecture, designed in the Intelligent Robotics Laboratory at Vanderbilt University, that permits the concurrent execution of software agents on separate machines while facilitating extensive inter-agent communication. IMA can be used to implement virtually any robot control architecture, from Sense-Plan-Act to behavior-based and hybrid architecture as well. Moreover, different architectures can be implemented simultaneously within separate agents so that a robot can have reactive agents for fast interaction with the environment and deliberative agents for planning or other supervisory control. Interaction between agents with different architectures is facilitated since the internal structures are completely independent.

IMA provides a two-level software framework for the development of intelligent machines. The robot-environment level describes the system structure in terms of a group of atomic software agents connected by a set of agent relationships. (We use the adjective, “atomic” to mean “primary constituent”, the building blocks from which all compound agents are formed.) The agent-object level describes each of the atomic agents and agent relationships as a network of software modules called component objects. At the robot-environment level, IMA defines several classes of atomic agents and describes their primary functions in terms of environmental models, behaviors or tasks. Sensor and actuator agents provide abstractions of sensors and actuators and incorporate basic processing and control algorithms. Figure 1 shows IMA agents for a mobile robot, ATRV-Jr.

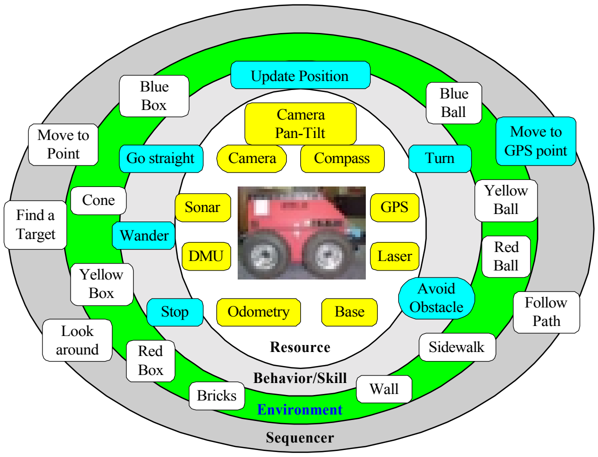


Figure 1: IMA agents for the ATRV robot.

0-7803-7203-4/01/$10.00 ©2001 IEEE

3 Human-Robot Interface for Supervisory Control

During supervisory control of the mobile robot, the robot provides the person with its sensory information and its status – a snapshot of the current state of the world – whereas the person provides supervision and assistance. We are implementing control of human-robot interaction (HRI) through an agent-based, distributed HRI architecture as shown in Figure 2. A key cognitive agent in the architecture is called the Commander Agent. It is a compound IMA agent that represents the user. The Commander Agent interacts with the robot through a robot-centric compound IMA agent called the Self Agent.

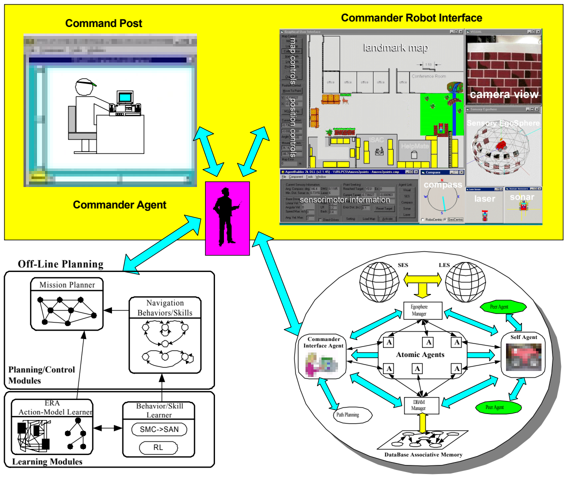


Figure 2: Agent-based HRI approach

The Self Agent decomposes a high-level user command into executable behaviors by using its short-term, long-term, and associative memories as described in section 4. These behaviors are then executed throughout the network of distributed agents in parallel [9].

3.1 Graphical User Interface

A GUI is an integral part of the supervisory control system. Robust interaction between user and robot is the key factor in successful human-robot cooperation. One way to facilitate the user's access to a wide range of robot information is to make the interface through agent-based multiple screens. This GUI provides communication between the human supervisor and the robot in the field. Data sent from the robot includes its current position and direction, sensor data and performance parameters such as the elapsed time since task initiation.

3.2 GUI Window

We implemented the GUI under the IMA architecture as shown in Figure 2. The main features include:

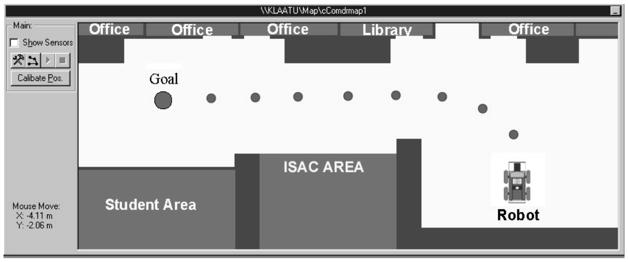
1. *A multiple map access screen*: This allows the user to access various world maps from a database. It also allows the user to specify any initial position or target position. The user has the option to manually calibrate the graphics scale for different locations. This manual calibration helps to reduce the error between the robot's actual position and the real-time on-screen data. An example is the calibration for GPS navigation.
2. *Mission Planner*: Path planning is performed using the Mission Planning Agent as shown in Figure 3. The planning module will be integrated into the HRI so that the user can perform complex planning on-line. The planning agent creates plans that can be stored as files for future use. The user can specify a mission using a series of tasks or using coordinates in the environment as shown in Figure 4. After the mission definition, the path planning algorithm will generate a path.

Name: Carlotta

Robot:
○ HepMate
● Scooter
○ Trilobot1
○ Trilobot2

Scooter: Go To, Coordinates, (-8686.8,2971.8)
Scooter: Go To, Table, Surachai
Scooter: Return, Home

Figure 3: Mission Planning Agent

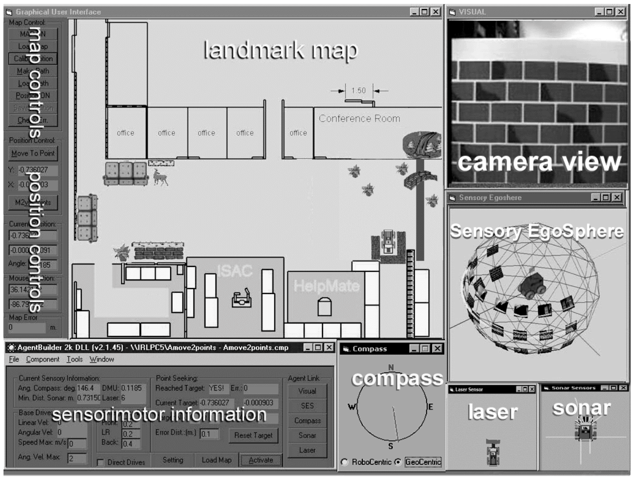


☐ Show Sensors

Mouse Move:
X: -4.11 m
Y: -2.06 m

Figure 4: Pop-Up Map from Mission Planning Agent

3. *Real-time robot information and support functions*: The GUI window can provide real-time robot data such as the current position and heading, a planned path, and the sensor data. The window also enables the user to control some of the properties of the agents that comprise the control system, such as the Commander Agent, the Self Agent, and various sensor databases.



Map Error: 0 m.

Current Sensory Information:
Ang. Compass: deg. 146.4
DMU: 0.1185
Min. Dist. Sonar: m. 0.73150
Laser: 6

Point Seeking:
Reached Target: YES!
Err.: 0
Current Target: -0.736027 -0.000903

Base Drive:
Linear Vel: 0
Angular Vel: 0
Speed Max: m/s 0
Ang. Vel. Max: 2

Front: 0.2
LR: 0.2
Back: 0.4

Error Dist.:(m.) 0.1

Figure 5: GUI Window

4 The Sensory EgoSphere

To facilitate remote control of a robot, a supervisory control system should enable a user to view the current sensory information. On our robot this includes visual imagery, sonar and laser signals, gyroscopic vestibular data, speed of each motor, compass heading, GPS position, camera pan and tilt angles, and odometry. A useful display of this multifarious information is critical for its correct and efficient interpretation by a user. However, if each sensory modality is presented separately, the user's task of combining that disparate information to make sense of the current situation can be quite complicated. It is so especially if the user has only an instantaneous snapshot of the world as sensed by the robot. A record of past sensory events would help to establish a context for the current state of the robot. However, the sheer amount of sensory data that it acquires over time not only precludes the possibility of storing all the data but also chances to overwhelm the user's ability to interpret it. The point is: efficient and accurate remote control of a robot would be facilitated by an intuitively understandable display of the robot's current multimodal sensory information in the context of significant events in its recent past.

Perhaps the most natural remote control environment is a virtual one that puts the user inside the robot as if she or he were driving it. Within such an environment, if sensory information is displayed in temporal sequence in the direction from which it comes, a human operator can discern which sensory events belong together in space and time. A directional, egocentric display takes more advantage of the person's natural pattern recognition skills to combine sensory modalities than does the usual sort of disconnected numerical or graphical display of sensory data. The quantity of data could be limited by keeping directional information only until it is displaced by new data sensed from the same direction To enable such display and memory, we are using a data structure, called a Sensory EgoSphere (SES).

4.1 Spherical Map / Short-term Memory

The concept of an EgoSphere for a robot was first proposed by Albus [1]. He envisioned the SES as a dense map of the visual world, a virtual spherical shell surrounding the robot onto which a visual snapshot of the world was projected, more or less instantaneously. Our definition and use of the EgoSphere differs somewhat. We define it as a database – a 2-D spherical data structure, centered on the coordinate frame of the robot, spatially indexed by azimuth and elevation. Its implicit topological structure is a that of a Geodesic Dome, each vertex of which is a pointer to a distinct data structure. The SES is a sparse map of the world that contains pointers to descriptors of objects or events that have been detected recently by the robot. As the robot operates within its environment events, both external and internal, stimulate the robot's sensors. Upon receiving a stimulus the associated sensory processing module writes its output data (including the time of detection) to the SES at the node that is closest to the direction from which the stimulus arrived. Since the robot's sensory processing modules are independent and concurrent, multiple sensors stimulated by the same event will register the event to the SES at about the same time. If the event is directional, the different modules will write their data at the same location on the SES. Hence, sensory data of different modalities coming from similar directions at similar times will register close to each other on the SES.

Our conception of the SES was inspired by a structure common to all mammalian brains, the Hippocampus (Greek for seahorse in reference to its shape). The hippocampus is a mammal's primary short-term memory structure. It lies along the base of the cerebral cortex. All cortical sensory processing modules have afferents into it, as do the brainstem and medulla oblongata [2, 4]. It has efferents into the frontal and prefrontal cortices. Research suggests that while an animal is awake, its hippocampus stores incoming sensory information while associating the sensory responses to events that are proximal in space-time. While asleep, especially while dreaming, the hippocampus stimulates regions in the frontal and prefrontal cortices. This is thought to be involved in the consolidation of short-term memory into long-term memory.

4.2 Geodesic Dome Topology

Given that the sensors on a robot are discrete, there is nothing to gain by defining the SES to be a continuous structure. Moreover, the computational complexity of using the SES increases with its size which is, in turn, dependent on its density (number of points on its surface). We use a (virtual) geodesic dome structure for the SES since it provides a uniform tessellation of vertices such that each vertex is equidistant (along geodesics) to six neighbors. The tessellation frequency is determined by the angular resolution of the sonar array.

The SES is a multiply-linked list of pointers to data structures. There is one pointer for each vertex on the dome. Each pointer record has seven links, one to each of its six nearest neighbors and one to a tagged-format data structure. The latter comprises a terminated list of alphanumeric tags each followed by a time stamp and another pointer. A tag indicates that a specific type of sensory data stored at the vertex. The corresponding time stamp indicates when the data was stored. The pointer associated with the tag points to the location of a data object that contains the sensory data and any function specifications (such as links to other agents) associated with it. The type and number of tags on any vertex of the dome is completely variable.

The SES is not a complete geodesic dome, instead, it is restricted to only those vertices that fall within the directional sensory field of the robot. Since the camera is mounted on a pan-tilt head, imagery or image features can be stored at the vertex closest to the direction of the camera. Sonar and laser work only in the equatorial plane of our robot and so their data is restricted to the vertices near the dome's equator. Figure 6 shows how the robot is posed in the SES.

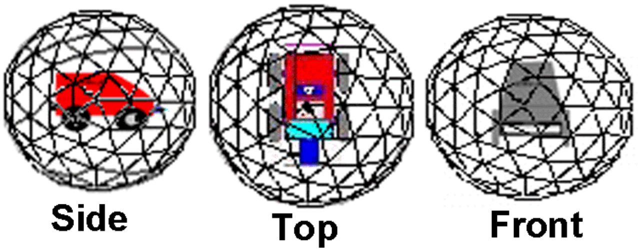


Figure 6: Relative position of robot to SES

4.3 Data Storage and Retrieval

Sensory processing modules (SPM) write information write information to the SES. A SPM calls the SES agent with a location, a tag, a time, and a pointer to its data. The SES agent finds the vertex closest to the given location and writes the tag and associated data in the vertex record, overwriting any existent tag record with the same name. Other agents, such as those performing data analysis, or data display can read from, or write to any given vertex on the SES. The SES agent will also search for the vertex or vertices that contain a given tag. Starting at a given vertex, it performs a breadth-first search of the SES. The agent requesting the search may or may not specify various search parameters such as, the starting location, number of vertices to return, search depth, etc.

On its vertices, the SES may contain links to data structures in the long-term memory (LTM) of the robot. For example, a landmark mapping agent could place a pointer to an object descriptor on the vertex in the direction at which the object is expected. Similarly, links could point to behaviors that can be executed in response to a sensory event.

When the robot is stationary, it can fill the SES with the data it senses. If the sensed objects are, likewise, stationary, then the data's location will not move on the SES. That was context of this research, since the use of the SES was to permit a remote operator to assess a situation in which the robot got stuck. To correctly register moving objects on a stationary SES requires object tracking, which requires searching. Moreover if the robot moves, the locations of data on the SES will also move – as functions of the heading and velocity of the robot and of the distances of the sensed objects from the robot.

In certain situations, the SES may relay unclear information to the supervisor concerning the present location of the robot. While traveling from point A to point B, there might be several locations where the SES appears similar. In these situations, the supervisor can then use the Landmark EgoSphere (LES) [7] in order to determine which region the SES actually corresponds to. Long term memory contains a two dimensional abstract map of the operating environment. The LES is the representation extracted from the long term memory used for localization of the robot by using current SES information, previous localization and rough odometry information.

5 Testbed Evaluation of Supervisory Control

In order to test the effectiveness of the supervisory control system, a scenario was implemented. An outdoor environment was simulated indoors by setting up the main aisle way of the research laboratory with various greenery, a simulated birch forest, deer, green turf and a cityscape. Independent of the simulated outdoors, the lab contained office furniture, graduate students, and other robots. The second setting was an outdoor parking lot with various obstacles, sidewalks and trenches.

5.1 Supervisory Intervention

In a supervisory control scheme, a person gives high level commands to the robot which then proceeds autonomously. Autonomous navigation can lead to problems, however. Certain relative spatial configurations of robot and environment may result in the robot being unable to move. This can occur, for example, if the robot becomes boxed in a corner, or strays off a path and tips over. The visual imagery from the robot's camera can be misleading or ambiguous to a supervisor who has not been monitoring the actions of the robot closely. This will happen sometimes since one reason for using supervisory control is to free the supervisor from following every move of the robot so that he or she can do other things – such as monitor several robots at once. If the supervisor was not monitoring the sensory data prior to the robot's jam, it may be difficult for the person to figure out the problem from the current, static sensory data and therefore unable to navigate the robot out of the predicament. Guesswork by the operator may worsen the robot's predicament. If, however, the robot has a spatially organized, short term memory that associates the various sensing modalities and if it can display with topological conformity the data it has stored, the task of manoeuvering the robot out the trap might be simplified for the supervisor.

5.2 Robot Operation

To test the advantage to supervisory control of using a Sensory EgoSphere, two scenarios were implemented. The first was an indoor scenario and the second was outside in a parking lot. In the former, the robot was given a command to travel from point A to point B. While traversing to the destination, the robot encountered a 3 way obstacle that it was unable to circumnavigate. The combination of the obstacle avoidance behavior with the attraction to the goal the robot enters a local minima situation. Figure 5 depicts the interface screen for the scenario. The supervisor used this interface to intervene in the situation. In the second scenario the robot overshot its path slightly and fell into a small ditch out of which it was unable to drive. The role of the supervisor in both situations was to determine the pose of the robot using only information supplied by the robot and to drive it remotely from its stuck position. An unprocessed time sequence of imagery from the camera of the robot in its stuck position is unlikely to provide enough information for the supervisor to discern the robot's pose with respect to the environment. Similarly, separate displays of the sonar and laser range data could be confusing. Moreover, these two modalities are prone to error depending on the surface characteristics of the objects (i.e. absorption, reflectivity, directivity) [5]. A hypothesis of this work is that the supervisor's task will be simplified by displaying the optical imagery and the data from the range finders on the SES, where the spatial and temporal associates between the two modalities is made explicit (see Figure 7).

5.3 Evaluation

One goal of our work was to determine the usefulness of using the SES in the HRI to determine the most appropriate way to drive the robot out of a difficult situation. A secondary objective is to evaluate the ease with which the supervisor can obtain information about the robot's status. We hypothesize that by using the SES, it is more effective and efficient to drive the robot out of difficult situations – the system is an improvement over a mobile robot interface that only provides instantaneous feedback from unassociated sensors.

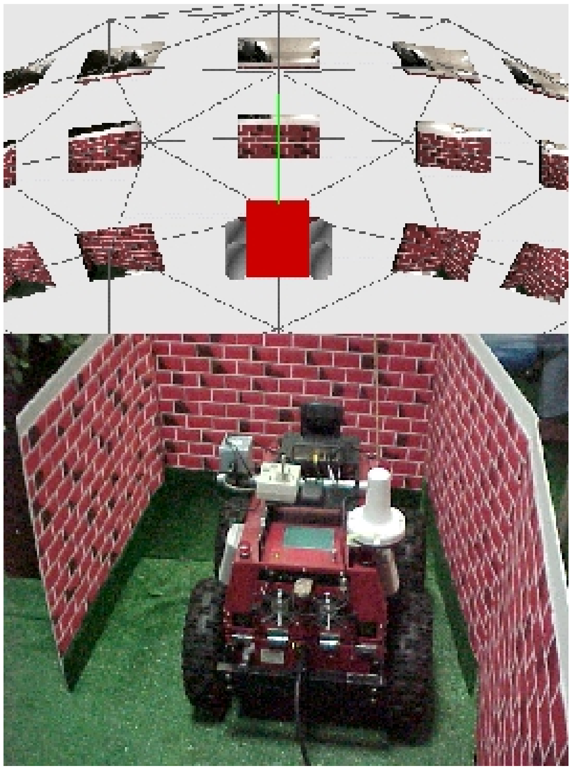


Figure 7: Sample SES

The test environment for the system evaluation enabled us to test the hypothesis as well as to explore other aspects of the robot's semi autonomous operation. The procedure is both an exploration and an assessment since it yields a causal hypothesis that can be tested by observation or through manipulation experiments. The study also establishes baselines and ranges for user behavior and system response. The controlling variables in this evaluation are the Sensory EgoSphere and the Human-Robot Interface. The dependent variables are the time it takes the user to become familiar with the user interface, and to drive the robot to a safe place. The time is the measure of the performance. The assumption was that the addition of the SES to the HRI decreases this time.

The supervisor will not be able to use the interface to drive the robot out of the situation, but only to evaluate and find an alternate solution. Figure 8 and 9 are the interface and the SES for the outdoor scenario and reflect the data that the supervisor had to manipulate in order to take corrective action.

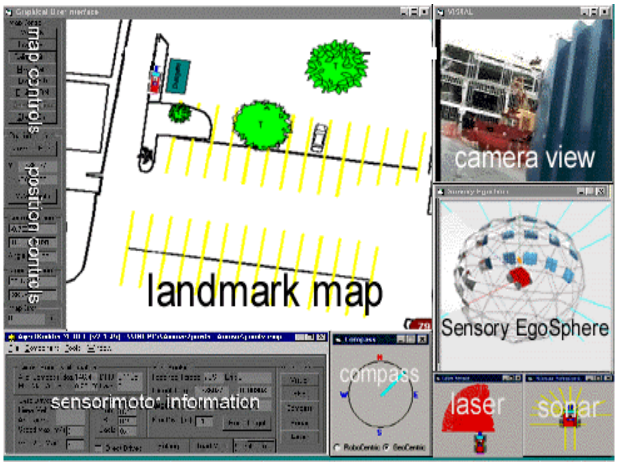


Figure 8: Scenario 2: Robot stuck in a ditch

The set up of the evaluation of this system involved the following steps. In preparation of arrival of the test users, the mobile robot was driven into the distress situation and the HRI generated and the SES built. Upon the arrival of the test user, the user was given a very brief introduction to the screens of the HRI. The users had a very low level of knowledge about robotics and in particular mobile robotics. After the introduction, the user will independently explore the windows and controls of the HRI and extract information about the present conditions of the robot. The amount of time required for the user to determine the state of the robot was recorded. The user then commented on the usefulness of the sonar, laser, compass and camera view in deciding what command to send to the robot to move away from the immovable state. The user lastly commented the usefulness and on how large a role did the SES play in helping to see what had happened and how to assist the mobile robot. The final stage, if possible, was for the user to use the HRI drive command to move the robot to a safe place. A safe place is defined as the location where all sensors are obstacle free and the robot can then autonomously continue on with the mission. While driving the robot to the safe place, the user will have real time feedback from the camera, sonar and laser as well as SES data in order to accomplish this task.

Our system was tested with several users, mostly undergraduate electrical engineering students, and the responses were timed. The robot was placed out of view to maintain the integrity of the test environment. In less than 3 minutes, the majority of the users determined the cause of the uncertain state for the robot. Two users were confused about the images on the SES because the dimensions were too small to extract relevant information. Most of the test subjects concluded that by driving the robot in reverse it would be possible to make an obstacle free path around the radius of the robot. The one drawback was that with the camera images alone, they were not able to determine much about the state of the robot or how to correct it. Most of the participants, felt the interface and SES were extremely easy and intuitive to use. From our experiments, we learn that the SES could enable the user to help the robot steer out of problematic locations.

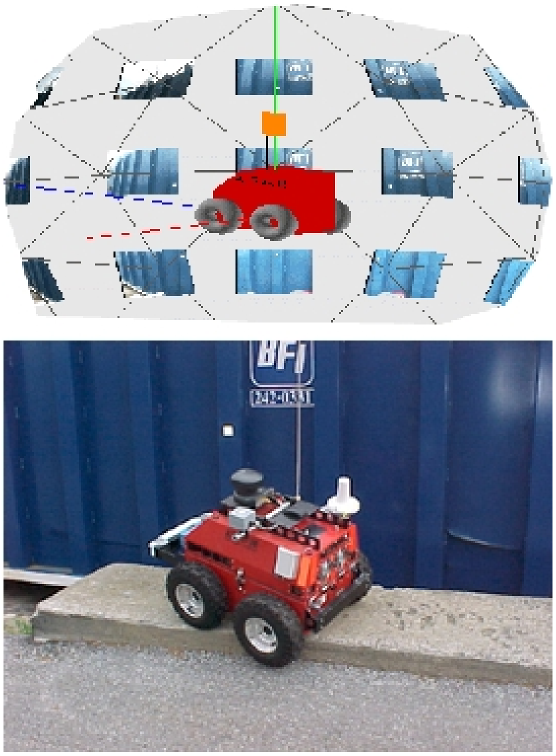


Figure 9: Scenario 2: Sample SES

6 Conclusion

Our system was tested with users who had low level of knowledge about robotics and mobile robotics. In this evaluation, we had a group of people determine the location of the robot given the HRI and SES. The user then had to determine why the robot was stationary and how to get it out of this situation. In less than 3 minutes, the majority of the users determined the cause of the uncertain state for the robot. Two users were confused about the images on the SES because the dimensions were too small to extract relevant information. Most of the test subjects concluded that by driving the robot in reverse it would be possible to make an obstacle free path around the radius of the robot. The one drawback was that with the camera images alone, they were not able to determine much about the state of the robot or how to correct it. Most of the participants, felt the interface and SES were extremely easy and intuitive to use. From our experiments, we learn that the SES could enable the user to help the robot steer out of problematic locations.

This paper presented the HRI which contained the SES, an environment map, sensory information and manual control. All of these elements proved to be very beneficial in human supervisory control compared to the classical method of vision feedback. The work detailed in this paper is currently in progress at CIS. The entire mobile robot architecture is a complex multi-agent structure. The SES was originally used for a humanoid robot ISAC in our lab [8]. Future work will include autonomous perception-based navigation of the mobile robot through the world using the Landmark EgoSphere, Sensory EgoSphere, and events [6]. The SES will be used for localization and navigation to targets using quantitative commands. In future work, solutions to the scenario from the supervisor's task will be merged with a long term memory capable of association. The sensory information will be displayed on the SES for the supervisor to view. Initially, the supervisor might guide the robot from its 'stuck' position. The database associative memory (DBAM) will collect the sensory signals from the SES and the motor signals triggered by the supervisor. These signals will be gathered into competency modules in the DBAM. The DBAM contains a spreading activation network (Bagchi, et. al [3]) that allows the robot to learn associations between competency modules. As the robot encounters this scenario more often, the DBAM should autonomously guide the robot from the rescue position.

Acknowledgement:
This research has been partially funded by the DARPA Mobile Autonomous Robotics Systems (MARS) program DASG60-99-1-0005. The authors thank the members of the Sensory EgoSphere Team: Kimberly Hambuchen, A. Bugra Koku, and Jian Peng.

References

[1] J. A. Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):473–509, May/June 1991.

[2] M. A. Arbib, P. Érdi, and J. Szentágothai. *Neural Organization: Structure, Function, and Dynamics*. MIT Press (Bradford), Cambridge, MA, 1998.

[3] S. Bagchi, G. Biswas, and K. Kawamura. Task planning under uncertainty using a spreading activation network. 30(6):639–649, November 2000.

[4] R. Carter. *Mapping the Mind*. University of California Press, Berkeley, CA, 1998.

[5] H. R. Everett. *Sensors for Mobile Robots: Theory and Application*, chapter 14. A K Peters, Canada, 1995.

[6] K. Kawamura, C. A. Johnson, and A. B. Koku. Enhancing a human-robot interface using sensory EgoSphere. In *IEEE Systems, Man and Cybernetics Conference*, Tucson, Arizona, October 2001. Submitted to.

[7] K. Kawamura, R. A. Peters II, A. B. Koku, and A. Sekmen. Landmark EgoSphere-based topolobical navigation of mobile robots. In *SPIE, Intelligent Systems and Advanced Manufacturing (ISAM)*, Newton, Massachussetts, October 2001. Submitted to.

[8] K. Kawamura, R. A. Peters II, D. M. Wilkes, W. A. Alford, and T. E. Rogers. ISAC: Foundations in human-humanoid interaction. *IEEE Intelligent Systems and Their Applications*, 15(4):38–45, July/August 2000.

[9] K. Kawamura, D. M. Wilkes, S. Suksakulchai, A. Bijayendrayodhin, and K. Kusumalnukool. Agent-based control and communication of a robot convoy. In *Proceedings of the International Conference on Mechatronics Technology*, Singapore, June 2001.

[10] R. T. Pack. *IMA: The Intelligent Machine Architecture.* Ph.d. thesis, Electrical and Computer Engineering, Vanderbilt University, Nashville, TN, May 1998.