# Performance Improvement Of A Mobile Robot Navigation System Through Ultrasonic Sensor Agents

**Eduardo Oliveira Freire, Teodiano Freire Bastos, José Eduardo Mendonça Xavier and Hansjörg Andreas Schneebeli**
Federal University of Espirito Santo
School of Engineering
P.O.B. 01-9011
Vitória - ES, 29060-970
BRAZIL
freire@ele.ufes.br

**Abstract**: The availability of sensorial data is a major desideratum for mobile robot navigation systems. For a mobile robot under construction at LAI/UFES, a behavior-based control system is being implemented, for which an agent-based ultrasonic system was developed. This sensing system is able to provide information on the type of detected obstacle as well as on the distance from it to the robot. By using the agents model, it was possible to design a ultrasonic sensing system with the capability of avoiding the generation of redundant or unnecessary information, as well as to decide whether information is relevant or not.

**Keywords:** Mobile Robots; Sensors; Ultrasonic Transducers; Behavior; Agents.

**Eduardo Oliveira Freire** was born in Aracaju, Brazil, in 1972. He received the BSc. degree in Electrical Engineering from Paraiba Federal University, Campina Grande, Brazil, in 1995. Currently he works at the Espirito Santo Federal University toward the MSc. degree with a thesis titled "Development of a Ultrasonic Sensing System for a Mobile Robot with Control of the Transducer Exciting Behavior-Based".

**Teodiano Freire Bastos** was born in Feira de Santana, Brazil, in 1965. He received the BSc. degree in Electrical Engineering from Espirito Santo Federal University, Vitória, Brazil, in 1987, the specialization degree from the Instituto de Automatica Industrial, Madrid, Spain, in 1989 and the Ph.D degree from Complutense University of Madrid, in 1995. His current research interests include applications of sensors in robotics.

**Jose Eduardo Mendonça Xavier** was born in Rio de Janeiro, Brazil, in 1968. He received the BSc. degree in Electrical Engineering and the MSc. degree from Espirito Santo Federal University, Vitória, Brazil, in 1990 and 1996, respectively. During his MSc. preparation, he developed a concurrent C++ tool to program behavior-based control systems for mobile robots. Currently he works at the Federal Technical School of Espirito Santo.

**Hansjörg Andreas Schneebeli** was born in Vitória, Brazil, in 1956. He received the BSc. degree in Electrical Engineering from Espirito Santo Federal University, Vitória, Brazil, in 1979, the MSc. degree from Instituto Tecnologico da Aeronautica, Sao Jose dos Campos, Brazil, in 1983 and the Dr. rer-nat degree from Universität Karlsruhe, Germany, in 1992. His current research interests include robotic systems.

## 1. Introduction

A differential drive robot using DC motors is under construction at the Electrical Engineering

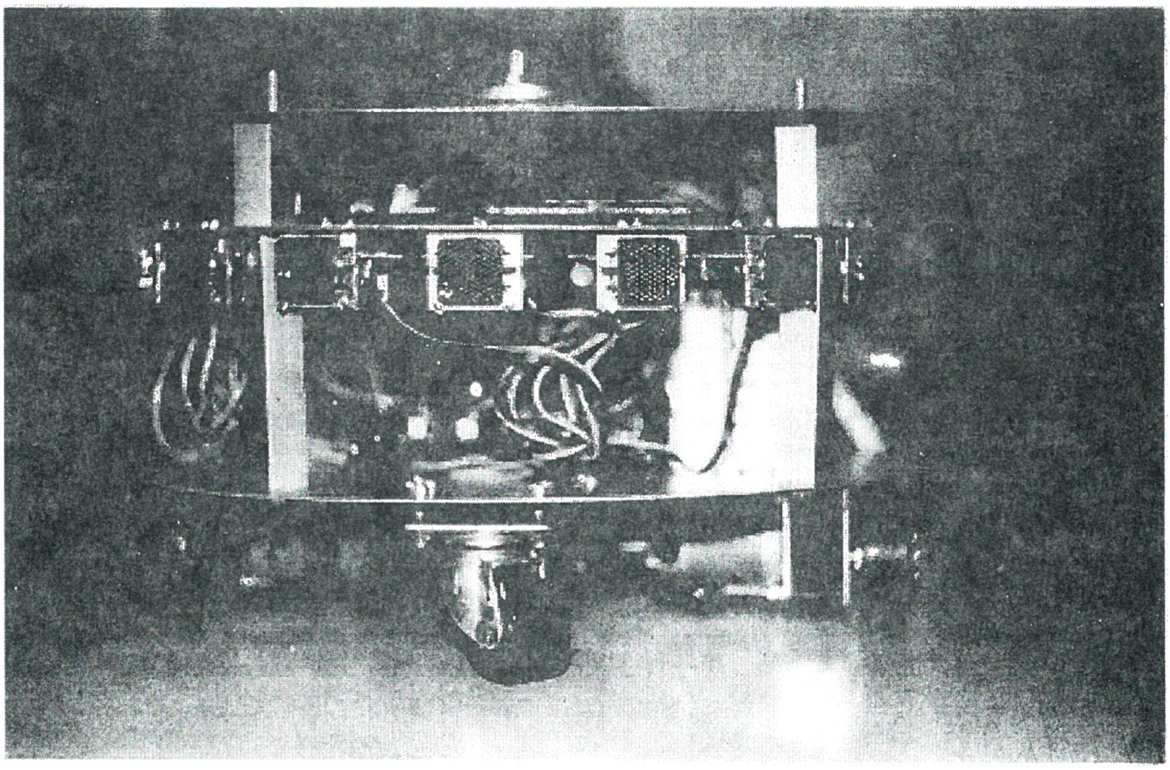

**Figure 1. Mobile Robot Brutus**

Department (DEL) of the Federal University of Espírito Santo (UFES), Brazil. This mobile robot is a round platform with four wheels: two of them are mounted on a common axis and are independently driven and the other two are free wheels. (Figure 1, the mobile robot Brutus). It is possible to see, beside the mechanical platform, some of the ultrasonic transducers. The round platform and the differential drive construction allow the robot to steer easily.

This mobile robot uses an MC-68332 microcontroller to control several processes involved in the mobile robot movement (e.g. internal and external sensorial perception and trajectory generation). The robot is designed to be able to avoid any collision, when moving itself in a semi-structured environment.

Semi-structured environments are those in which the limits are not known by the system, but those that have closed configuration and regular ground. The environmental configuration can change suddenly, but not very often. It can also change deeply, but not very fast. Finally the robot operating environment should have a finite number of known objects (walls, corners, chairs and tables), but unknown objects can suddenly appear in the trajectory of the robot.

## 2. Control Architectures for Mobile Robots

There are three main categories to classify the different control architectures for mobile robots into: classic architectures (functional); agent-based architectures and hybrid architectures (Coste-Manière et al, 1995).

### 2.1 Classic Control Architectures

The traditional control architectures for mobile robots decompose the control system of the robot in an ordered sequence of functional components (Jones and Flynn, 1993).

In these architec-tures, the data are initially collected from all sensors. Noise and conflicts in the data are solved in such a way that a consistent model of the "real world" can be constructed. This model should include information about the dimensions, forms, positions and orientation of all objects in the operation area of the robot.

Normally, the model or the map of the "world" is programmed in the robot memory before it begins to work. This procedure limits its operation to the environment that it "knows". In this case, the sensors are just used to locate the robot in the memory map of the world.

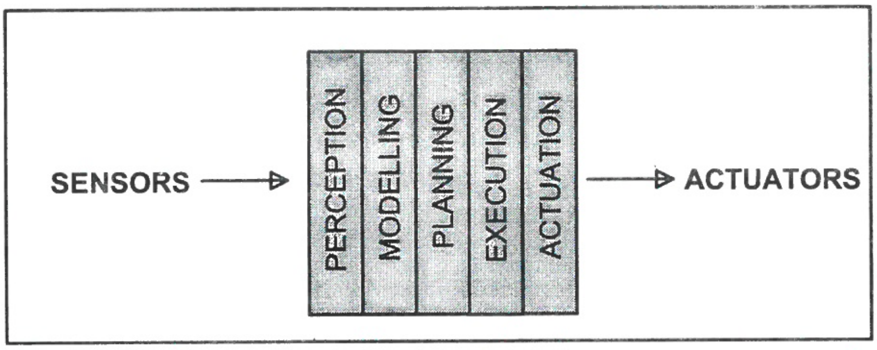


**Figure 2. Traditional Architectures for Mobile Robots Control (Brooks, 1986)**

The robot uses the world model to plan sequences of actions that will reach a final objective (execute a task). Finally, the plan is carried out sending suitable commands to the actuators. The sequence of actions described is illustrated in Figure 2.

### 2.2 Behavior-based Architectures

Behavior are layers of control systems that all run in parallel whenever appropriate sensors are fired (Jones and Flynn, 1993). The problem of conflicting sensor data is replaced by the problem of conflicting behaviors. Fusion is consequently performed at the output of behaviors (behavior fusion), rather than at the output of sensors. A what behavior is dominant for a given situation.

In these architectures there is not a notion of a behavior calling another behavior as a subroutine. Instead of this, all behaviors are executed in parallel, but higher-level behaviors are able to temporarily suppress lower-level behaviors. When the higher-level behaviors are no longer triggered by a certain sensor condition, the lower-level behaviors resume the control. So, in a certain way these architectures are inherently parallel, and the sensors interact directly with all behavior layers and each behavior interacts directly with the actuators (Figure 3).

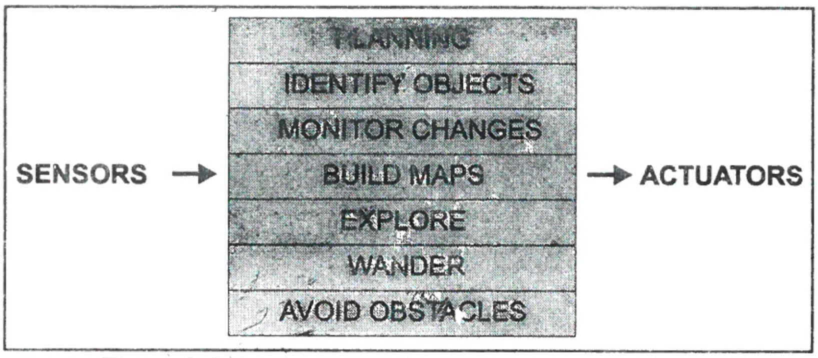


**Figure 3. Behavior-Based Architecture (Brooks, 1986)**

In the behavior-based architectures there are not unified data structures or "real world" models. This characteristic allows the implementation of such control architecture using computational resources, which is simpler than implementing the traditional control architectures. The behavior-based architectures also allow the insertion of new behaviors layers without changing the system configuration. Due to these advantages of the behavior-based architectures as compared to the traditional architectures, the mobile robot Brutus (under development at LAI/UFES) is being designed with a behavior-based control system.

### 2.3 Hybrid Architectures

The hybrid architectures are the most recent approach. They have been developed with a view at solving the limitations inherent to both mentioned approaches, using a combination of coherent and well defined models (Coste-Manière et al, 1995). The hybrid architectures integrate both low and high-level considerations in a coherent structure. Quite often, it results in a task/mission decomposition and explicit representation of the "world" at the highest level.

The lowest levels have similarities with the behavior-based approach, but are based on the traditional control theory. The middle-levels assume the work of translation and coordination of commands and actions to the outlying levels.

## 3. A Structure for Constructing Agent-based Control Systems

Schneebeli (1992) presented an abstraction for agent-based controllers, in which they are represented by concurrent and interconnected modules, distributed in three categories: sensor agents (primitive sensor agents and virtual sensor agents), behavior agents and actuator agents (virtual actuator agents and primitive actuator agents), shown in Figures 4 and 5. In this approach, like in Brooks' (Brooks, 1986), a group of modules is responsible for an activity (behavior) to be executed by the robot. Each activity represents a system goal. Mechanisms of interaction and mediation between modules define the main goal.

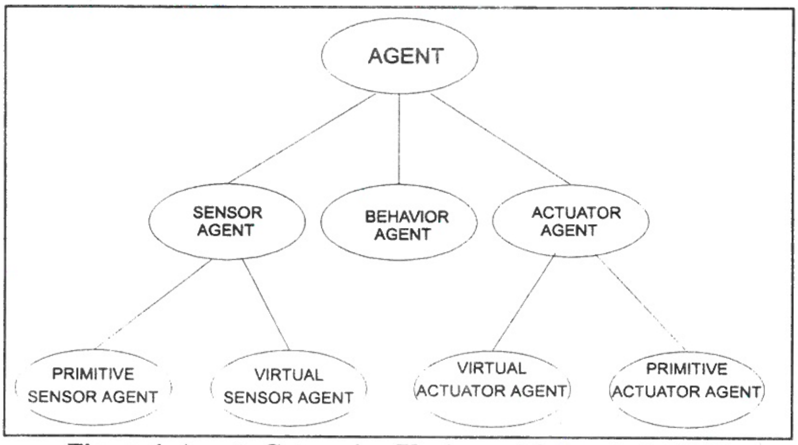


**Figure 4. Agents Categories (Xavier and Schneebeli, 1995)**

This module categorization provides an abstraction that makes the purpose of each module closer to its representation, giving to each agent category some particular characteristics, related to status and communication, and associated with their specific role For example: the actuator agents have such a characteristic that they can be commanded by just one behavior agent at any time. The sensor drivers are represented by primitive sensor agents while tasks like *build maps* are represented by virtual sensor agents. The modules that define actions or behaviors are represented by behavior agents and tasks like *move forward* are represented by virtual actuator agents. Finally, the actuators output drivers are represented by primitive actuator agents.

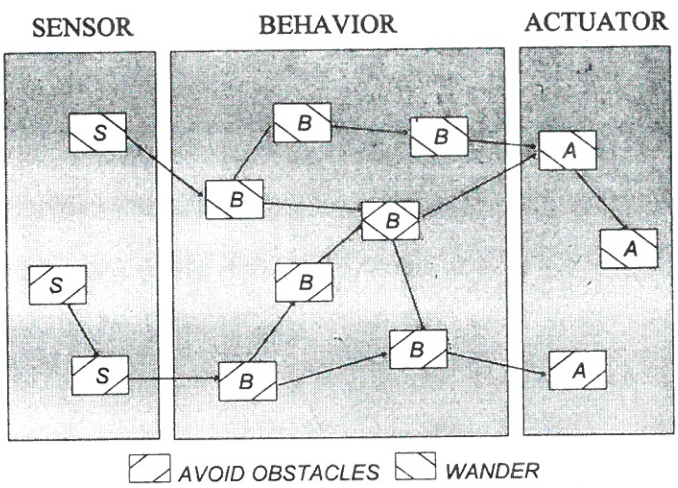


**Figure 5. The Categories of Agents of the Control System (Xavier and Schneebeli, 1995)**

The structure for constructing the agent-based control systems developed by Xavier and Schneebeli (1995) implements Schneebeli's (1992) abstractions using the C++ programming language and concurrent libraries. So, a concurrent object oriented structure is built. This structure is very suitable for behavior-based systems modeling, joining interesting characteristics such as modularity, encapsulation, inheritance, concurrence and the C++ efficiency. For those who are favourable to the use of symbolic programming in these systems, it can be obtained through the use of virtual machines implemented in some modules.

In the structure aforementioned, the control modules (agents) are constructed as instances of user defined classes derived from the sensor, behavior and actuator classes. So the definition of just a class can be used to create multiple agents, by instantiation and, if necessary, at running time. In such a case, the difference between the created agents would be given by the arguments passed to the constructor of the class, and by the communication gates established.

The dynamic instantiation and desinstantiation of agents plus the dynamic inter-module connection mechanisms defined in the structure (Yonesawa, 1988), provide a dynamic reconfiguration capability to the system that can be used to help the robot handle and manage diverse situations using different control strategies.

Mechanisms of inheritance and the modularity provided by the structure ensure an easy expansion of the control system. So the creation of new kinds of agents can be done by deriving classes from the sensor, behavior and actuator classes, thus minimizing the need for code rewriting. The encapsulation of data and functions in classes and objects ensures the system modularity.

## 4. Ultrasonic Sensing System Hardware

The ultrasonic sensors may be used to help the navigation system of the robot, providing useful information to the trajectories planning. To implement the ultrasonic sensing system, Polaroid (7000 series) electrostatic transducers (see Figure 1) were used. The resonance frequency of each one is about 50 kHz, and the radiation lobe aperture is about 34° wide (Figure 6).

The most common way of obtaining information about the environment where the robot operates is to use one transducer or a transducer vector fixed on the axis of a step

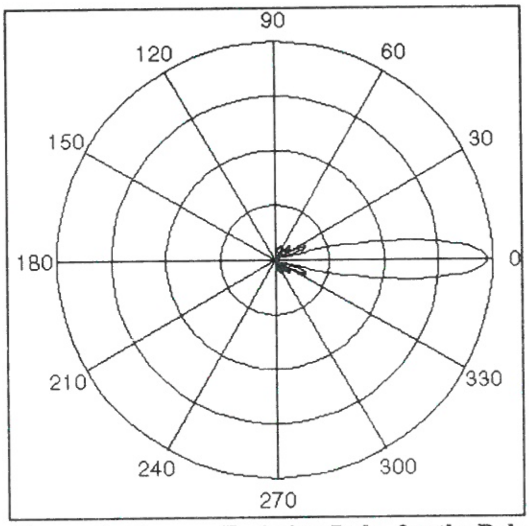


**Figure 6a. Energy Emission Lobe for the Polaroid 7000 Series Transducer**

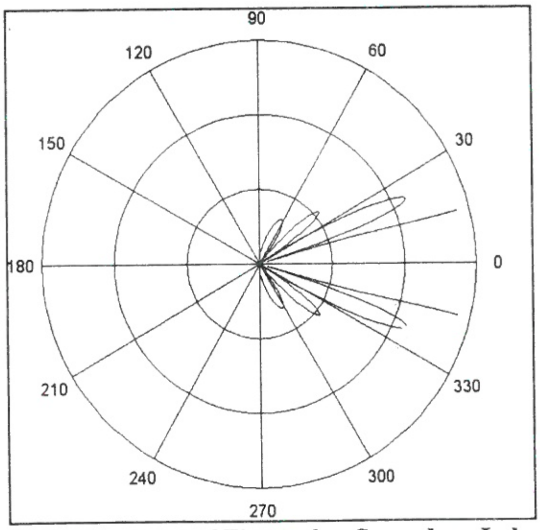


**Figure 6b. Zoom of Figure 6a - Secondary Lobes**

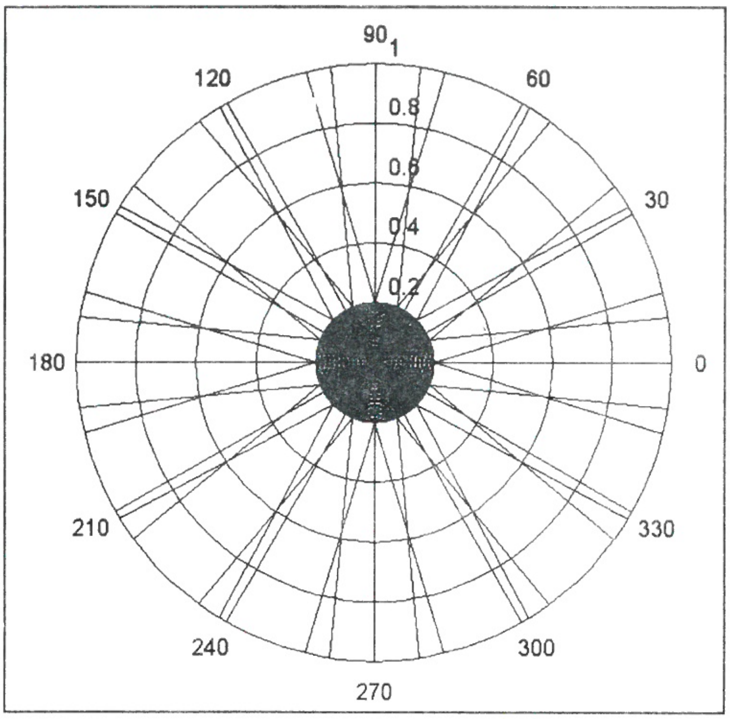


**Figure 7. Coverage Map of the 16 Ultrasonic Transducers Ring**

motor, to sweep the environment around the robot (Tsuzuki, 1990). This technique is useful when the ultrasonic sensing system is used just to locate environmental marks, with the objective of calibrating the localization of the robot in the environment. On the other hand, when the objective is to guide the robot through the external sensors data (such as robot Brutus ), this technique may fail, because it is possible that an object on the robot's trajectory cannot be detected, because during the necessary time for the step motor to complete one turn around its own axis, the robot may walk a considerable distance, and a collision will probably occur.

In this work, a transducer set is distributed and fixed around the robot's round platform (as shown in Figure 1) as a way of sensing the environment in all directions almost at the same time. The transducers are not started simultaneously, and the direction of sensing is changed by software. However, it was necessary to define the number of ultrasonic transducers needed for a suitable space coverage around the robot. This number was found through studying the transducers emission lobe.

As the robot platform diameter is 40 cm and considering the transducer energy radiation lobe aperture equal to 34°, the MATLAB was used to simulate the transducers ring spatial coverage to many transducers sets, each one with a different number of transducers. The set of best cost-effective relationships obtained was the set with 16 ultrasonic transducers.

Figure 7 shows a modeling of the main lobes of the 16 transducers, which are represented by straight lines. The secondary lobes (Figure 6b) are not represented in this Figure. The main lobes intersection point is about 40cm from the transducers ring. Beyond this distance, the obstacles can be detected with an error of ±1% of the measured distance. Obstacles located at distances less than 40cm from the transducers ring, can also be detected by the transducers secondary lobes, but with bigger errors.

A Polaroid board (6500 Series Sonar Range Module) was adapted to be shared by the 16 ultrasonic transducers through a demultiplexer circuit. The Polaroid board provides the transducers excitation pulses and executes the basic processing of the received echo signals (through the demultiplexer circuit).

As a way of compensating for the temperature variation of the robot operation environment (that causes sound velocity variation), a temperature sensor based on the LM35 was used.

## 5. An Agent-based Ultrasonic Sensing System

For the mobile robot under construction at LAI/UFES, a behavior based control system will be provided. So, its ultrasonic sensing system was designed as an agent-based system.

In the agent-based architectures already developed, the sensor agents often are very simple: they work firing behaviors when their outputs exceed a certain threshold. So, the sensor agents are often periodically started. This characteristic limits the sensing system performance.

This paper proposes a non-periodic way of starting the sensor agents. In order to do this, an Object Oriented Concurrent Structure developed by Xavier and Schneebeli (1995), for constructing agent based systems, is used. The transducers are fired according to the priority given to each of them. When the priority increases, the firing frequency also increases. The priority assigned to each transducer ($\varepsilon_k(x)$) varies according to the function:

$$\varepsilon_k(x) = a\left(\frac{1}{x}\right) - b\left(\frac{dx}{dt}\right) + \left[(-1)^n \cos\alpha + c\right] \qquad (1)$$

where:

- $k$ is the $k^{th}$ transducer;
- $x$ is the distance from the transducer to the detected obstacle;
- $c$ is the minimum priority assigned to each sensor agent ($c > 1$);
- $\alpha$ is the angular displacement of the $k^{th}$ transducer related to the reference transducer;
- $n = 0 \Rightarrow$ the robot front is in the same direction of the reference transducer, and
  $n = 1 \Rightarrow$ the robot front is in the opposite direction (the value of n is determined by the low level motor control);
- $a$ and $b$ are adjustable constants.

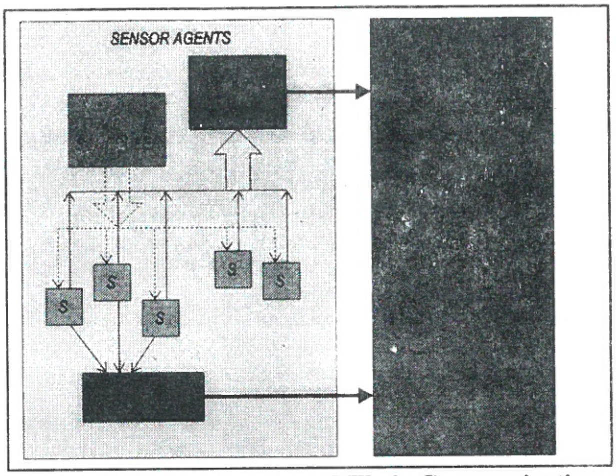


**Figure 8. Sensor Agents and Their Communication Lines**

The first term of Equation 1 means that the priority of the transducer is higher when it detects closer obstacles. The second term guarantees that the priority of one sensor agent increases as its associated transducer gets closer to the detected obstacle. The last term (a constant of which value varies according to α and the direction of the robot displacement) prevents the assignment of very small priorities to the sensor agents. The concept of "small" varies according to the associated transducer location.

Four types of sensor agents have been implemented: one "Local Scheduler" sensor agent, one "Arbitrator" sensor agent, 16 "*S*" sensor agents and some "Discriminator" sensor agents, which are dynamically created whenever they are necessary.

The system scheduler (Xavier and Schneebeli, 1995) starts the Local Scheduler agent. This agent is responsible for determining the more suitable sequence of starting the 16 "*S*" agents, according to the priority (1) assigned to each one (Figure 8).

The $S_k$ agent is responsible for exciting the $k^{th}$ ultrasonic transducer and for receiving the corresponding echo signal, that can be detected as a pulse (the time of flight method for range measurement) or as an envelope (Bastos, 1994). The second case only happens when the distance from a specific sensor to the detected obstacle becomes lower than a certain minimum level. At this moment the $S_k$ agent requests the creation of a Discriminator agent sends a warning to the agents $S_{k-1}$ and $S_{k+1}$, and then these three transducers detect the envelope of the echo signals and send this information to the recently created Discriminator agent (Figure 9). This last agent will define if the obstacle is a part of the environment contour (walls and corners) or some inside object (chairs and tables). If another $S_k$ agent detects an obstacle within a distance shorter than the minimum one, it will repeat the above described procedure, since Discriminator agents are dynamically created. Each $S_k$ agent has the capability of deciding on the relevance of the information and on whether it should be sent to the Arbitrator agent.

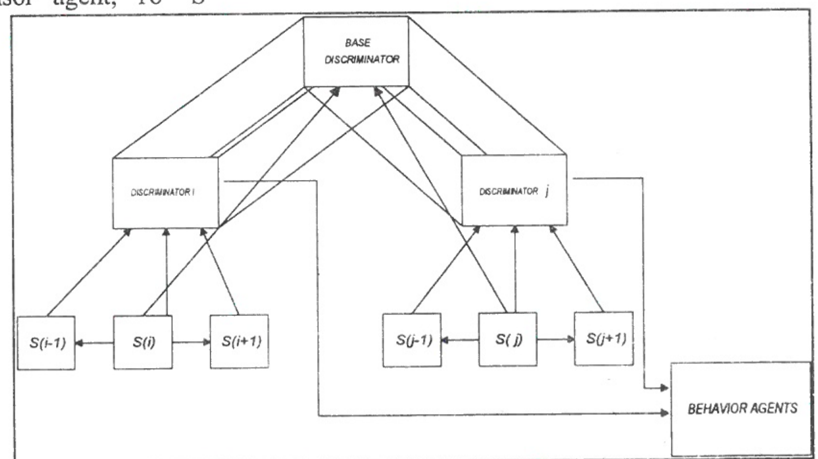


**Figure 9. Creating Discriminator Agents**

The Arbitrator agent has the function of defining which of the messages received from the $S_k$ agents should be forwarded to the behavior agents.

Through the application of the behavior based control model, it was possible to design an "intelligent" ultrasonic sensing system. It is able to define in which frequency each transducer should be fired and to decide on whether a piece of information is relevant or not. In this way, it drastically reduces the generation of redundant or unnecessary sensing information. This capability makes the work of the robot navigation system easier, because it now receives only relevant sensing data.

With the implementation of the Discriminator agent, it is possible to distinguish if an obstacle is a part of the environment contour or if it is an inside object. So, the system navigation also becomes more flexible, because it now receives information about the type of obstacle the sensing system has detected, in addition to the information about the distance from it to the transducers. Discriminator agents can dynamically be created and all of them are executed in parallel, thus allowing the simultaneous processing of data related to more than one obstacle.

The "intelligent" ultrasonic sensing system implemented is able to provide a set of more consistent data for the robot navigation system (behavior agents), thus allowing an improved performance of the navigation system.

## REFERENCES


BASTOS, T. F., **Seguimiento y Análisis de Entornos de Soldadura por Arco Automatizado Mediante Ultrasonidos**, Ph.D Thesis, Universidad Complutense de Madrid, Madrid, Spain, 1994.

BROOKS, R. A., **Achieving Artificial Intelligence Through Building Robots**, A.I Memo 899, MIT, MA, USA, 1986.

COSTE-MANIÈRE, E. , WANG, H.H. and PEUCH, A., **Control Architectures: What's Going On?**, Proceedings of the International Program Development in URIC, Lisbon, Portugal, March 2-3, 1995, pp. 54-60.

JONES, J. L. and FLYNN, A.M., **Mobile Robots: Inspiration to Implementation**, Chapter 9, A K Peters, Wellesley, MA, USA, 1993.

SCHNEEBELI, H. A., **Die Steuerung von Mehrfinger-Greifersystemen**, Ph.D Thesis, Karlsruhe University, Karlsruhe, Germany, 1992.

TSUZUKI, F., **Desenvolvimento de um Sistema Sensorial Baseado em Ultra-Som para Auxiliar o Sistema de Navegação de Robôs Móveis**, Master of Science Dissertation, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil, 1990.

XAVIER, J. E. M. and SCHNEEBELI, H.A., **Estrutura para a Construção de Sistemas de Controle Baseados em Agentes para Robôs Móveis**, Proceedings of the 2nd Simpósio Brasileiro de Automação Inteligente, Curitiba / PR, Brazil, 1995, pp.97-101.

YONEZAWA, A., SHIBAYAAMA, E., TAKADA, T. and HONDA, Y., **Modeling and Programming in An Object-Oriented Concurrent Language**, in A. Yonezawa and M. Tokoro (Eds.) Object-oriented Concurrent Programming, MIT PRESS, MA, USA, 1988, pp. 55-90.