ARCHITECTURE OF AN AUTONOMOUS SYSTEM: APPLICATION TO MOBILE ROBOT NAVIGATION

Jean-Pierre Müller*   Heinz Hügli†

Abstract. We investigate the behavioural approach for building autonomous systems like mobile robots. In this paper we begin by giving a motivation for this approach to autonomy and then describe the architecture selected for its implementation. Finally, we present the experimental mobile robot together with its main features.

1. GOALS OF THE PROJECT

The main goal of this project is to validate an autonomous system architecture in the context of mobile robotics.

This research project is motivated by the central role autonomous intelligent systems will play in advanced robot applications. Its main phases are:

- selection of the behavioural approach and design of the system architecture,
- implementation of MANO, the robot and its distributed architecture.

1.1 Autonomous systems for robot applications

Robot technology has progressed significantly in recent years and has brought a good number of solutions that apply typically to high precision industrial robots that paint, weld, assemble and package products in usually well structured environments.

However, beside these applications, there is an unsatisfied need for robots working in less structured environments like cleaning robots in a supermarket, mail delivery robots in an office or tidy up robots working in a bookstore. These advanced applications require additionnal ability like mastering a changing environment or coping with humans sharing the same environment and simultaneously exploiting efficiently the environment features. Autonomy is the first requirement of a system capable of mastering these new applications.

A second requirement relates to cognitive abilities. Robot tasks can be cognitively demanding. This is especially true for service robots which must cope with a far wider and less controllable range of situations than robots that are confined to factories.

In short, autonomy and sufficient cognitive abilities are basic requirements for advanced robot systems.

1.2 The behavioural approach

The behavioural approach is based on the simple principle of existence of individual behaviours and on the coordination of these behaviours.

We define a behaviour as an independent stereotyped action that is maintained by a specific perceived stimulus. Example of robot navigation behaviours are *Wander Around* or *Go Along*. Each behaviour is implemented as an independant module and is activated by its stimulus. When several stimuli appear simultaneously, several behaviours can be active at the same time. The overall behaviour emerges from the coordination of the various active behaviours.

This general and tempting approach hides a number of unanswered questions we have addressed in the frame of this project in order to select an architecture. Among them we name:

- definition and design of individual behaviours
- compatibility and richness of behaviours

*Institut d'Informatique et Intelligence Artificielle, Université de Neuchâtel, Monruz 36, 2000 Neuchâtel, Switzerland, Muller@info.unine.ch

†Institut de Microtechnique, Université de Neuchâtel, Tivoli 28, 2003 Neuchâtel, Switzerland, Hugli@imt.unine.ch

- reuse of behaviours and design of compound behaviours that make use of several other elementary behaviours
- principle and form of a control structure selecting the behaviours in order to achieve autonomy
- level of performance of the overall behaviour and conditions for its emergence from the individual behaviours
- distribution of knowledge and functionality among the behaviours and the control structure
- pertinence of sensitive, topological and geometric knowledge representations

Finally, we selected our behavioural architecture which builds on the following basic elements

- a set of behaviours: a particular behaviour beeing in charge of deciding what to do according to a set of stimuli
- a set of controllers selecting which behaviours actually control the system at any given time

1.3 MANO

MANO is the implementation of our behavioural approach. It consists of a development and experimentation environment centered around a mobile robot, dedicated vision hardware and a number of interconnected workstations. In addition to offering an implementation of the selected behavioural architecture with three controllers and about twenty behaviours, this environment offers comfort and flexibility. Among others it has the following useful capabilities:

- network-wide development and experimentation capabilities
- virtual robot interface that allows equivalent experimentation on simulator or real robot
- multi-language support

2. STATE OF THE ART

2.1 Behavioural approach

A behavioral approach has already been advocated by many authors including Brooks [1], Agre and Chapman [2] among others, to deal with unpredictable environment. The relationship between behaviors and planning has been explored from different perspectives. Agre and Chapman [3] distinguish plans as programs where everything has to be specified in advance and plans as communication which require a competence in the form of reactive behaviors. Malcolm and Smithers [4] show that a reactive layer is a reliable grounding for actions to be used by a classical planner. Other use of the reactive behaviors-classical planning can be found in [5]. However, as far as we know, only Mataric suggested to use the stimuli themselves to build a representation of the environment in [6] as we describe later in this paper. The originality of our approach is to take this suggestion seriously and explore the consequences of this.

2.2 Vision and behaviours

The behavioural approach sets new challenges to the vision research community: the design of vision-based behaviours. Whereas traditional robot vision represents only one step in a more complete procedure that involves sensing, mapping and navigation [7] vision-based behaviours are complete systems that sense, interprete and act.

Simple vision-based behaviours have been studied and demonstrated in the past involving aspects of control theory, pattern recognition, neural networks, i.e. [8].

But systems developped so far are limited: often they work at a unique level of interpretation, take little advantage of possible interpretations from scene-understanding, use a single behaviour at a time.

The project addresses some of these limitations by developping a large number of behaviours based on several vision devices and by running them concurrently. Also, for exploring more demanding tasks, we introduce compound behaviours which are build upon several simple behaviours.

2.3 Robot system architecture

Numerous architectures can be found in the litterature. Our starting point is clearly the subsumption architecture proposed by Brooks [1]. Performance comparisons of several architectures [9, 10]

3. BEHAVIORAL APPROACH TO BUILD AUTONOMOUS ROBOTS

We assume a task for a mobile robot and introduce basic elements of the behavioural architecture we select for controlling it.

3.1 Robot task and environment

The mobile robot is located in a building, moving horizontally on flat ground. Walls and various obstacles make up its environment. The robot must finally fulfill tasks like:

- exploring an area,
- carrying letters in an office environment,
- tidy up chairs in a room.

To do so, the robot must have basic skills in moving, object recognition, mapping and navigation.

3.2 The animal behaviours

The behavioural approach is inspired to some extend by the animal world in which elementary behaviours can be observed.

As an example, the bee, despite a poor visual system, easily navigates to and from its hive. To reach its hive, the bee could for instance proceed in two steps: first, *Go Along* a characteristic fence, and, as soon as the hive is visible, *Go Towards* it. The bee's goal is reached in two successive behavioural steps which had to be coordinated.

3.3 The artificial behaviours

A behaviour is a stereotyped action that is triggered by a stimulus starting the behaviour and maintaining it as long as the stimulus exists. Typically, the action is maintained by the stimulus in a feedback loop across the environment.

As several stimuli can be active at the same time, several behaviours may become active simultaneously. Fully independent behaviours will then run concurrently while behaviours which share some common resources are incompatible and exclude each other. Among several behaviours competing for a common resource, a single one can be selected. The selection is performed according to a decision scheme dictated by controllers.

Controllers act on the behaviours by allowing them to run or not, and by acting on the decision scheme used for the selection of one among competing behaviours. They do so in a context sensitive manner and in a way to fullfill the robot's task.

3.4 Behavioural architecture

We define autonomy as the capability of a system to use at any time the actual circumstances to serve its purpose (survival for biological systems, any functionality or role for artificial systems). The definition of autonomy requires a compromise between:

- behaving in a situated way, that is in the context of the particular, concrete circumstances, otherwise it would not be able to deal with the dynamics of the world
- behaving in order to ensure its survival or its role otherwise it would just be driven by the environment and, therefore, would not be autonomous

The observed behavior of the system at any given time will be determined both by its contribution to the system survival or role fulfillment (the meaning) and by the environment (the form).

In order to realize this compromise, we propose a distributed architecture (Fig. 1) composed of:

- a set of behaviors (in a technical sense): a particular behavior being in charge of deciding what to do according to a set of features (the stimuli)
- a set of controllers selecting which behavior actually controls the system at any given time.

4. THE MOBILE ROBOT

The mobile robot is composed of the commercially available Nomad 200 [11] and two dedicated vision devices mounted on the robot:

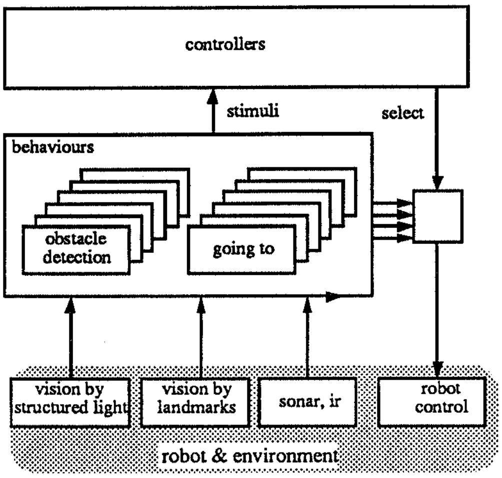

Figure 1: Architecture of the behavioural system

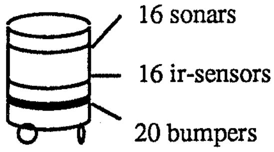

Figure 2: Nomad 200

vision by structured light and vision by landmarks. All three are described thereafter more in detail.

4.1 The Nomad200

Nomad 200 (Fig. 2) is a 1 meter tall mobile robot. Its cylinder-shaped body is moved by a three wheels motion system that allows two degrees of freedom: translation and rotation around its vertical axis. Its sensors are arranged symmetrically around the body, each measuring a given sector of the surrounding environment. They are of three different types: 16 sonars, 16 infrared range sensors, 20 tactile sensors and a digital compass. Proprioceptive sensors include odometry.

4.2 Laser range by structured light

The device uses the principle of structured light to derive the geometry of a profile in front of the robot. A laser device projects light rays oriented ahead of the robot. The system developed uses the profile of laser light to measure the environment 1 m ahead of the robot [12]. The output is a laser range profile given by a list of 3D segments belonging to the ground.

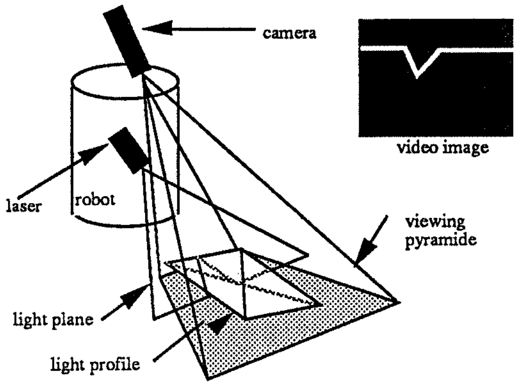

Figure 3: Laser range by structured light

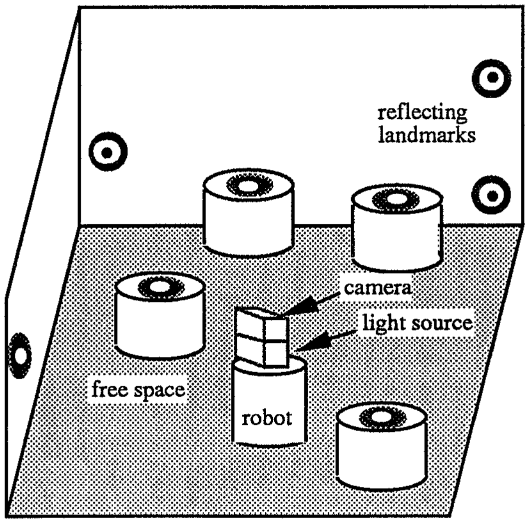

Figure 4: Active vision of landmarks

4.3 Landmark vision system

This active vision system uses a light source coupled to a camera to enhance the detection of reflecting landmarks distributed in the environment. The bright landmarks are detected, labeled and tracked in a dedicated Transputer system that produces the time sequence of labeled landmarks [13] at an approximate rate of 15 Hz.

5. THE BEHAVIOURS

5.1 The set of basic behaviours

We give here a list of behaviours. More on vision-based behaviours can be found in [13, 14].

Follow link north-, east-, south- or westward: The behaviour we are using consists in going in the direction of the general orientation of the free space computed from local sonar readings. The sonar readings are integrated in a local map of the configuration space and a skeleton is extracted based on Voronoi graphs. Coordinated with a behavior to turn at crossings, a complete navigation system has been described elsewhere only using a coarse representation of the free space by a topology of crossings and passages [15].

Obstacle detection and stop: The behaviour stops the robot and is activated when obstacles are found in front of the robot. The stimulus is from range sensing: given shapes of the range profile are interpreted as obstacles.

Wander around: Moves the robot straight ahead and changes the direction only when an obstacle is detected in front of it. The new direction is such that the new move is away from the obstacle. Here, an obstacle is a configuration of radial range field detected by the infrared sensors.

Go Towards: Moves the robot towards a landmark. When several landmarks are visible, the move is towards the landmark just ahead of the robot. The behaviour is no longer stimulated when the landmark is near to the robot. In this behaviour, landmarks are visible spots, detected, labeled and tracked by the landmark vision system.

Go Along: Moves the robot along extended obstacles like walls, keeping a constant distance to them. This behaviour comes in several flavours depending on the sensing device used for its implementation and the prefered wall following direction. Regarding the implementation, a first one is based on the radial range profile from infrared, the other comes from an interpretation of the laser range profile.

Go Along Left, Go Along Right: This behaviours are specialized forms of Go Along. Whereas any direction of following is possible with Go Along, these two new forms have forced following directions.

Obstacle detection and turn: This behaviour turns the robot in the direction of the nearest space without obstacles. It is stimulated by obstacles located in front of the robot and detected by given shapes of the laser range profile.

Move To: Moves the robot to a position. It is stimulated by a difference between the nominal position and the odometric position.

Push Box: This behavior is stimulated by an object near to the robot. It moves the robot towards this object and upon collision, continues its move by pushing the object straight ahead. The robot's moves are controlled to keep the object on the straight line. Here, object detection for moving toward it and for controlling the pushing is based on radial range field detected by the infrared sensors.

Homing: On activation, it moves the robot to a fixed location and orientation with respect to two landmarks. It is stimulated on detection of two appropriate landmarks detected by the landmark vision device.

Position Estimation: The behaviour is stimulated by observed landmarks. It finds the relative position of the robot with respect to a set of three known landmarks.

Free-space mapping: Keeps track of the geometry of the environment as observed when the robot is moving. This is done by reporting successively the laser range profiles into a map.

5.2 The compound behaviours

Compound behaviours take advantage of a specific use of basic behaviours to accomplish more demanding tasks. A compound task is characterised by the set of basic behaviours it makes use of, and by the specific schema it uses to activate them. In addition, compound behaviours can activate parametrized behaviours, allowing more latitude in their use.

Tidy Up Chair: This behaviour tidies up a chair. It looks for a chair, detects it, goes towards it and pushes it to a specified location. This is a compound behaviour built above other behaviours like Homing, Move To, Push Box and Go Toward.

6. THE CONTROLLERS

We call sensory state a set of stimuli and the current sensory state the set of stimuli detected by each of the behaviors. A sensory state defines unambiguously the set of behaviors which are able to control the system when it is in this sensory state. By hypothesis, the sequence of behavior selections and the resulting sensory state changes together with the current sensory state constitute the unique source of knowledge for controlling the system. Based on this source, three controllers have been defined:

- the learner is responsible for finding regularities either structural (recurrent occurrences in the same sensory state) or causal (recurrent sequences in the history). In our case, we only organize the causal information from the history into a topological graph, exploiting sensorimotor neighborhoods as exposed in [16]. So, the nodes are the perceptive states composed of subsequences of sensorimotor states and the edges the selected behaviors. This controller selects the behaviors in order to explore the consequences of selecting a particular behavior in a given perceptive state.

- the localizer is responsible for finding which regularities learned by the previous controller are applicable in the current perceptive state. In our case it is reduced to find in which node of the topological graph the system currently is. When lost, it selects the behaviors in order to experience a known sensorimotor subsequence (path) corresponding to a node of the topological graph.

- the motivator (orientator) is responsible for orienting the system to the desired sensorimotor states. In our case, it localizes and marks the nodes of the topological graph containing the desired sensorimotor subsequences.

- the planner is responsible to exploit the applicable regularities in order to drive the system into a desired perceptive state chosen by the motivator. In our case, it selects a behavior on the path to the desired sensory state in the topological graph.

It is important to make some remarks:

- these controllers have been described in a very general way. For example, to localize itself means to know where we are only in the context of navigation and relatively to a topology which does not need to be isomorphic to any geometric structure of the actual environment. For example, Mataric presents another way to structure the information [17, 18].

- each of these controllers extensively needs the others. For example, the planner and the localizer need the learner but the learner also need the localizer, and so on. However, they do not need to explicitly communicate with each other but through the topological graph.

- but the motivator, all the controllers can potentially choose a behaviour, the learner to experiment to explore new paths, the localizer to find his way and the planner to achieve tasks chosen by the motivator. An arbitration is made such that only one behaviour is selected at any given time by a propagation mechanism.

7. COMMUNICATION

Communication between all the elements given so far is implemented by the blackboard. It plays a central role in the architecture. Both control and knowledge are handled through it.

7.1 Blackboard and control

The blackboard is the communication channel between the planning, the behaviours and the virtual robot. Information can be exchanged

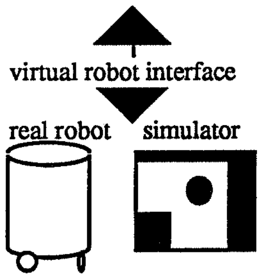

Figure 5: Virtual robot

through it simply by writing and reading a given item, each item being referenced by a key. The information found is of two types: control information for the selection of behaviours and knowledge to be exchanged.

8. Implementation

MANO is the implementation of the architecture on a network of Sun workstations.

8.1 Requirements

Above all, we ask MANO to implement the behavioural architecture and to offer the development and experimentation environment for improved robot skills and tasks. Additionaly, we have following requirements

1. offer virtual robot interface, i.e. the capability to use a real robot with real world interactions as well as a simulated robot in an artificial environment
2. execute in a unix environment, with networking capabilities: behaviours should be able to execute on any node of the network
3. offer monitoring capabilities
4. offer flexibility: changing and adding behavioural modules should be easy

8.2 Virtual robot

By developing a simulator with same behaviour and interface as the real robot, we obtain the advantages of a virtual robot. A simple switch allows to select either the real or the simulated robot (Figure 5).

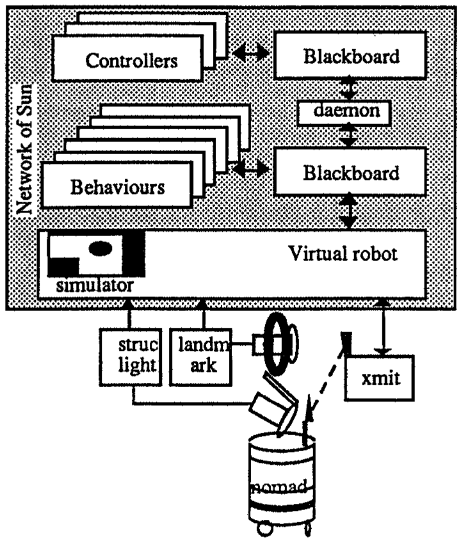

Figure 6: MANO

The capability provides advantages. It speeds up development and makes the robot available to more people.

8.3 MANO

MANO is shown in figure 6. We recognize the Nomad robot , the two vision device mounted on it and connected to dedicated vision processors, and, finally, the network of Sun workstation which hold behaviours and controllers.

Main features thereof are: the implementation of each behaviour as an independent process, making it highly responsive; the use of server processes to implement the virtual robot and the blackboards, defining a client-server relation for the interprocess communication channels is implemented by sockets. The blackboards are implemented by the ndbm Unix tool [19].

8.4 Simulation and monitoring

The nomadic simulator is the equivalent of Nomad 200 in form of a simulator. It includes an artificial environment described by a 2-D map with polyhedral-shaped obstacles in which the robot is moved according to the motion commands given to the robot. The robot movements and interactions with the environment are simulated and reported as the sensed output. The simulation is either exact or corrupted by adjustable noise.

The vision simulator offers the same function as to the two vision systems. It includes an artificial environment described by a 2-D map of obstacles and landmarks. Visual interactions between sensor and environment are simulated according to the sensors models and given as virtual sensed data.

Beside simulation, MANO implements a full range of monitoring tools. To the monitoring tools delivered with Nomad 200, we added equivalent functions for vision. Among them, let us mention the monitoring of

- sensors
- behavioural stimulation and activity
- state of blackboards

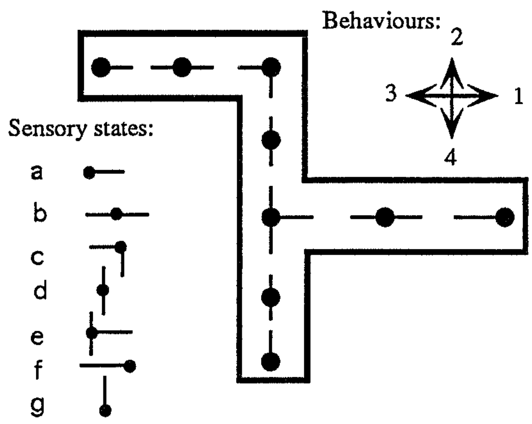

Figure 7: Landmarks for corridors

9. EXPERIMENTS

MANO is now operative in the described form.

9.1 Experiment 1

Using the behaviours described above and based on a learned topological representation of the interaction, our agent is capable of demonstrating self-localisation and navigation between sensory states. Figure 7 shows one environment (corridors in an office environment) and the typical landmarks the robot is expected to observe when using only the sonar sensors. We observe that different zones in the environment may produce the same sensory state for the robot; a graph containing only one sensory state as a node context would be ambiguous. Instead, with sequences of length 3 (two sensory states and the behaviour of the transition) as contexts, we avoid ambiguity and conserve deterministic transitions. Figure 8 shows the *optimal* topological graph of this interaction; nodes are contexts (context 'a1b' means *I am in sensory state 'b' coming from sensory state 'a' through behaviour 1*) and arcs are transitions between contexts through the behaviour contained in the context of the destination node (for more clarity, each bi-directional arc replaces two unidirectional arcs). This graph is almost similar to the result from the learner process. A task is defined by a circular sequence of sensory desired states.

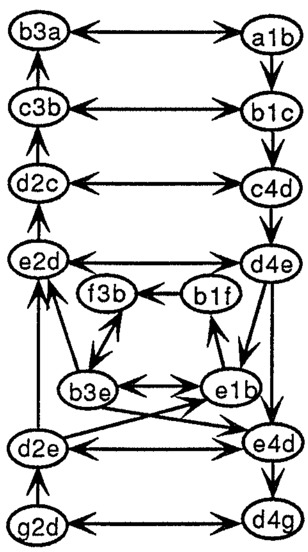

Figure 8: The graph of perceptive states

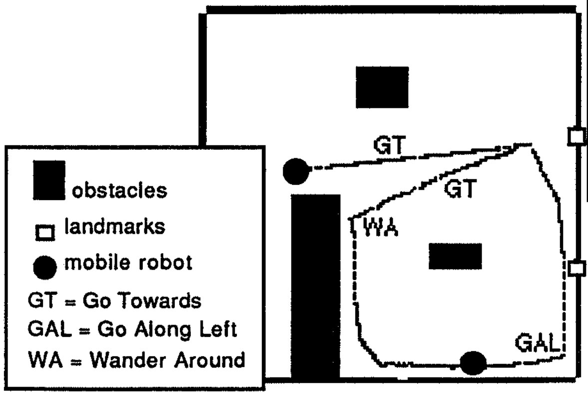

Figure 9: Emergence of a new behaviour

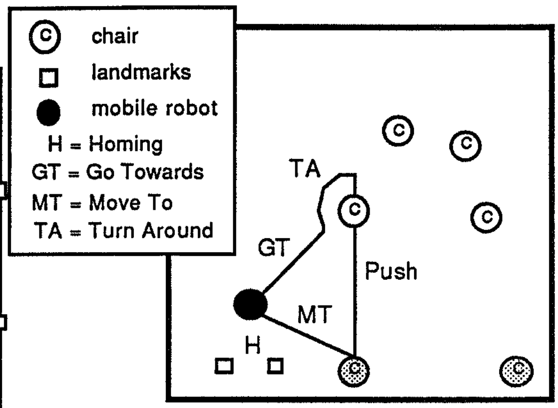

Figure 10: The compound behaviour Tidy Up Chair

9.2 Experiment 2

This second experiment illustrates the combined use of several behaviours. According to the principles introduced above, we expect the several behaviours to cooperate and possibly we expect the emergence of new behaviour in favourable situations.

In the environment of figure 9, the robot navigates under the control of 3 behaviours: Go Towards (GT), Go Along Left (GAL) and Wander Around (WA). The control is static and selects among activable behaviours, the one with highest priority. In the experiment, the order of priority is, from highest to lowest: GAL, GT, WA. This means for example that WA is activated when none of GAL or GT is activable.

Figure 9 shows an example where the described control mode results in a robot trajectory that rapidly enters into a stable loop of sequence GT, GAL, WA. The result can be interpreted according to the bee navigation paradigm: as a successful sequence of elementary behaviours used for reaching a goal. Another interpretation is in term of a new behaviour that now emerges from an appropriate conjunction of environment and behavioural control. Given these three behaviours, this robot environment and this selection rule, a new behaviour emerges that has new features not available in the isolated modules. The whole is more then the sum of its parts.

9.3 Experiment 3

Let us consider the problem of designing a specific task to be fulfilled by the robot: tidy up chairs in a room. Given the rich capabilities of the robot, the question arises if and how the task can be realised. Two fundamental approaches exist for trying to solve the problem: expecting the emergence of this new behaviour in the sense illustrated in experiment 2, or, teaching the behaviour to the robot. In the following experiment, we solve the tidy up chairs task by selecting and teaching sequences of adequate behaviours and storing them into a compound tasks.

The first step is the selection of a set of simple behaviours that possibly solve the task. Here this set is Homing, Go Towards, Push, Go To, Turn Around, Wander Around, Avoid Obstacles. In the second step, various sequences are experimentally tested and the good ones are kept and stored to form the compound behaviour Tidy Up Chair.

Basically the behaviour looks for chairs, repeatedly finds one and moves it to a definite position. It also avoids significant obstacles, finds a way out when lost and repeatedly goes back to a homing position for localisation. Figure 10 illustrates the Tidy Up behaviour. We observe a trajectory resulting from its main sequence (Homing, Go Towards, Turn Around, Push, Go To).

10. CONCLUSIONS

MANO is operational and several experiments show various behaviours, in moving the robot, recognizing objects and mapping the environment. The use of topologies based on the robot/environment interactions has been demonstrated. We also demonstrated the complementarity of means for generating new behaviour: emergence and teaching.So far, our behavioural approach and the derived architecture showed successful solutions to several aspects of autonomous systems.

Future research includes:

- advanced behaviours based on the local map
- improvement of existing behaviours and development of new ones.
- further tests to improve the robustness of the learned topological graph with respect to noisy information.
- the use of local odometric annotation on the topological graph to correct localisation.
- the possibility to express more sophisticated task specifications

Acknowledgement

This work has been supported by the Swiss national research program "Artificial Intelligence and Robotics" (PNR23) under project number 4023-027037. We would like to thank all the team working on this project and particularly Miguel Rodriguez, Yoel Gat from the "Institut d'Informatique et Intelligence Artificielle", Claudio Facchinetti, Franois Tiche, Fabrice Chantemargue et Gilbert Matre from the "Institut de Microtechnique".

REFERENCES

[1] R. A. Brooks. A robust control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.

[2] P.E. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *AAAIU87*, pages 268–272, 1987.

[3] P.E. Agre and D. Chapman. *What are plans for.* In [5], 1990.

[4] C. Malcolm and T. Smithers. *Symbol grounding via a hybrid architecture in an autonomous assembly system.* In [5], 1990.

[5] *Designing Autonomous Agents.* MIT Press, 1990.

[6] M. J. Mataric. Navigation with a rat brain: a neurobiologically inspired model for robot spatial representation. In *Proceedings of the first international conference on simulation of adaptive behaviors.* MIT Press, 1990.

[7] Alberto Elfes S.S. Lyengar. Autonomous mobile robots, perception, mapping and navigation. *IEEE Press*, 2, 1991.

[8] H. Mori and al. A mobile robot strategy applied to harunobu-4. In *Proceedings of the Conference on Pattern Recognition*, 1988.

[9] P. Bonasso T. Dean. 1992 aaai robot exhibition and competition. *AI Magazine*, Spring, 1993.

[10] C. Congdon & al. Carmel versus flakey: A comparison of two winners. *AI Magazine*, 1992.

[11] Nomadic Technologies Inc., Stanford. *Nomadic Robot programming manual*, 1992.

[12] H. Hügli & G. Maître. 3d by structured light: implementation and evaluation of a vision system for small parts. *Optical 3-D Measurement Techniques*, pages 468–477, 1989.

[13] Heinz Hügli Claudio Facchinetti. Two vision-based behaviours for autonomous mobile robots. In *Proc. Conference on Pattern Recognition and Vision*, Jan 1992.

[14] C. Facchinetti H. Hügli, G. Maître and F. Tièche. Vision-based behaviours for autonomous mobile robots. In *Proc. SGAICO 92*, Sept 1992.

[15] Y. Gat and J.P. Müller. Reactive navigation based on simple world modelling. In *Proceedings of the 9th Israeli Symposium on Artificial Intelligence*, pages 43–56, 1992.

[16] J.P. Müller Y. Gat, M. Rodriguez. Enriched sensitive and perceptive localization. In *Proc. SGAICO 92*, Sept 1992.

[17] M. J. Mataric. Environment learning using a distributed representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1990.

[18] M. J. Mataric and R. Brooks. Learning a distributed map representation based on navigation behaviors. In *Proceedings of the USA-Japan Symposium on flexible automation*, pages 499–506, 1990.

[19] F. Chantemargue H. Hügli, F. Tièche and G. Maître. Architecture of an experimental vision-based robot navigation system. In *Proc. Swiss Vision*, Sept 1993.