WHISKER BASED MOBILE ROBOT NAVIGATION

David Jung and Alexander Zelinsky
Intelligent Robotics Laboratory
Computer Science Department, University of Wollongong
Wollongong, NSW 2500, Australia.
djung@cs.uow.edu.au

Abstract
In this paper we describe the design and implementation of a unique proportional whisker sensor and our general behaviour based software architecture. The Architecture for Behaviour Based Agents (ABBA) is used to realise purposive mobile robot navigation in a cluttered indoor environment. The whisker was developed specifically to meet the requirements of high speed and close wall following. We also discuss the specific architecture we have constructed

Keywords*: Whisker, Sensor, Navigation, Behaviour.*

1. INTRODUCTION

One of the requirements of our research, and a common requirement for an autonomous mobile robots, is the ability to navigate competently in an unstructured environment. This problem has received much research attention in the robotics community. A major difficulty is that there are two competing aspects to navigation. A robot must be able to react very quickly to its environment to prevent collisions with static and dynamic obstacles. Also, in order to achieve useful purposive navigation a robot must use a map-like representation of its environment. This implies some sort of limited cognition (as defined by [McFarland 91]). The former implies minimal processing to meet the real-time requirements, while the later requires more processing than can typically be achieved for real-time response.

The basic control design approaches can be broadly divided into four types. These are defined by [Mataric 92a] and we briefly reiterate them and their shortcomings here. The *purely reactive* approaches use a mapping from sensor sets to associated actions; a set of rules [Brooks 87]. The *planner-based* strategies originated with the symbolic AI community and employ a sense-plan-act cycle. The plan stage uses cognitive techniques to reason about a symbolic world model. There also exist hybrid systems which employ reactive components beneath planner-based systems to provide the benefits of both. Another approach is *behaviour-based*, which uses a set of interacting distributed concurrent behaviours, each of which may incorporate memory and learning of environment representations.

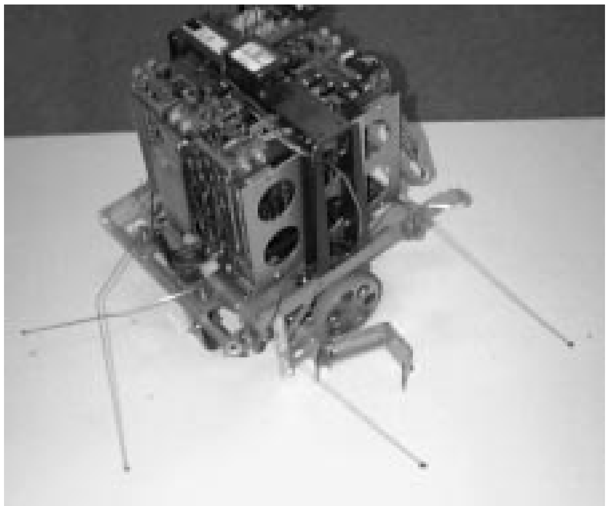

Figure 1 - Yamabico Robot with Whiskers

The purely reactive approach achieves robust real-time performance, but the tasks that can be achieved are limited because of the lack of any cognition. The planning approach suffers from a number of problems including slow interaction with the environment due to slow processing, the frame problem and the symbol grounding problem. The hybrid approaches attempt to marry two incompatible philosophies and still suffer from many of the problems of the planning approaches. For more depth on these problems refer to [Pfeifer 95]. The behaviour-based approach has the potential of real-time response and cognitivistic processing in a uniform manner.

As part of our ongoing research into cooperative heterogeneous multi-robot systems, described below, we require purposive navigation and high speed wall following at close range, up to 1m/sec at 30mm. This immediately precludes the use of purely reactive or planner based architectures. Hence we are adopting the behaviour-based approach. Also we have developed a novel whisker sensor tailored to our task.

The remainder of this paper describes both the whisker sensor and the navigation system based on the whisker we have developed. The following section describes our cooperative multi-robot research project for which the whisker based navigation was developed. Section 2 describes our whisker sensors and how they relate to existing whiskers. The section also describes how the whiskers are used to detect 'natural' landmarks in the laboratory environment. Section 3 details techniques for whisker based wall following and finally the last two sections discuss the general architecture and specific organisation of behaviour units used to accomplish whisker based navigation.

1.1 The Context

This section briefly describes the robots and the research into cooperation between multiple robots for which the whisker based navigation was developed. The aim of the research is to assess the effect on performance of changing some parameters of the cooperation technique employed. The context of the assessment is a specific cooperative task performed by our 'Yamabico' autonomous mobile robots [Yuta 91], one of which is pictured in Figure 1. The task we have chosen is for two Yamabico robots to clean our laboratory floor. Each has different tools such that neither can accomplish the task alone. One robot is equipped with a vacuum cleaner and the other with a sweep for sweeping close to walls, as is explained below. For more information on the cooperation aspect refer to our previous publication [Jung 96].

The robots each have two wheels controlled using shaft encoder feedback, and four ultrasonic range sensors, one facing each compass direction relative to the robot. The robots also have three processor cards connected via a bus, two Motorola 68000's which manage the main application program and the four ultrasonic sensors and one INMOS T805 for the locomotion control.

One of the robots, the *vacuum*, has a vacuum cleaner capable of being controlled via software. It's task is to vacuum piles of litter from the laboratory floor. It cannot vacuum close to walls or furniture. It also has a CCD camera mounted on top which sends video signals via a video transmitter to our vision system.

The MEP tracking vision system was developed by Fujitsu R&D Japan. It is designed for real-time tracking of multiple objects in black and white frames of NTSC video. The hardware consists of a tracking module which can track up to 72 objects at frame rate (30Hz). A processor card running the VxWorks operating system executes the application program and controls the vision card via a VME-bus. The tracking of objects is based on simple 8×8 or 16×16 template comparison using cross correlation which is based on the block-matching system developed by Inoue's Laboratory at University of Tokyo [Inoue 85].

The vision system is connected via ethernet to our host UNIX workstation which has a serial connection to a radio modem. The robot also has a radio modem and using the networking software we developed the visual-motor path is complete. The vision system allows it to 'see' piles of litter, but not fine particles scattered over the floor.

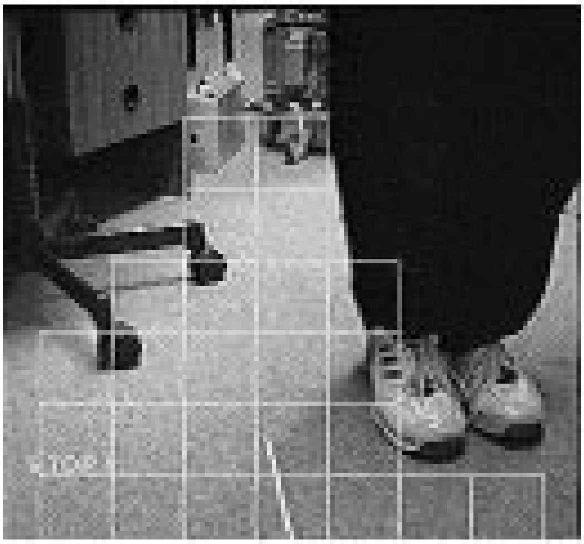

Figure 2 - The vision system segmenting the image into regions of floor carpet and obstacle.

The vision system's matching capability has also been used to implement a robust collision avoidance behaviour [Zelinsky 95b] for the *vacuum*. Figure 2 shows the view from the robot, where the image has been divided into a course grid. Each square in the grid is classified based on how well it matches against a template of the laboratory carpet. There is a separate carpet template for each row in the grid to allow for the texture scaling as the carpet becomes farther away towards the top of the image. For each element of the grid, the vision system gives a distortion value which indicates the confidence of the match between the carpet template and the image. Next the main program thresholds the distortion to classify the grid element as either carpet or non-carpet. The visual behaviour for collision avoidance maps the image space into the floor and physically directs the robot to avoid areas that were not matched as carpet. Although this matching process contains noise, since the classification is re-evaluated at frame rate a robust collision avoidance behaviour results.

The other robot, the *sweep*, has a brush tool that is dragged over the floor to sweep distributed litter into larger piles for the vacuum to pick up. The robot is required to sweep very close to walls were the vacuum cannot reach. This will require basic competence at navigating around our cluttered laboratory, including obstacle avoidance, and the ability to follow along walls closely at high speed. This cannot be achieved using existing whiskers, as is shown in the section to follow.

2. WHISKERS

To realise the sweeping task the robot required sensors to reliably follow walls at close range and high speed and simultaneously avoid obstacles. The robot was initially equipped with four ultrasonic range sensors as shown below.

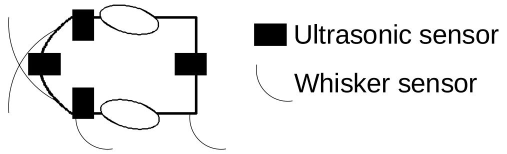

Figure 3 - Schematic layout of sensors

The ultrasonic range sensors proved not to be robust for wall following because they often give incorrect distance readings for specular surfaces that are inclined with respect to the robot. In addition the sensors we have cannot measure distances under the 5cm we require for close following. Infrared sensors were considered, but are typically difficult to make robust for a wide variety of surface material types and colours.

Next we considered contact sensors, such as whiskers. Most contact sensors for tactile sensing in use on robots to date can be divided into two types. Those which measure strain and those that measure deformation. Most tactile sensors that measure the strain are used in systems with manipulators. For a review see [Nicholls 89] and [Fearing 90]. Active whisker sensors that measure strain for recovering object shape have also been reported, for example by [Russell 85], but these are prohibitively expensive for the use of many on a small inexpensive mobile robot and provide more information than we require. Of the tactile sensors that measure deformation, one common type is the skin like sensors, that typically are comprised of an array of simple sensors for measuring deformation at a particular point. These are usually expensive and measure deformations only over a small range. The physical sensors used may be optical, magnetic, resistive, pneumatic or even ultrasonic, for example [Russell 90]. Another common type of sensor for measuring deformation is the mechanical whisker. This type of whisker, as is in use on mobile robots to date, provides only binary deformation information; that is, contact/non-contact information (refer to [Everett 88], [Russell 90] for examples). A common and cheap design involves mounting piano wire such that deformation contacts it with a conducting plate hence making a circuit as shown below.

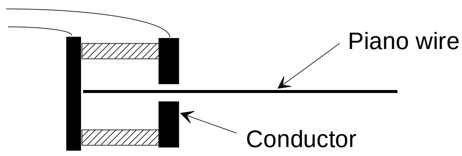

Figure 4 - A binary whisker[1]

For wall following the information required for control of the locomotion is the robot's inclination to the wall and its distance from the wall. If we mounted two binary whiskers on the side of the robot in the positions shown in Figure 3 we would have four sensor states from the whiskers. These would represent following straight too close and too far, and following at an angle too large or too small. Since the *sweep* is required to follow walls to within 30mm to enable the collection of litter near the base of the walls, the information provided by binary whiskers is insufficient.

In nature evolution often evolves specific sensors to provide required information to the organism very directly instead of investing in expensive computational machinery that can extract the same information from other sensors via some higher level perception mechanism. A good example of this type of design is the auditory system of the Noctuid moth as described by [McFarland 93]. Consequently we decided to develop unique proportional contact whiskers and mount them along the left side of the robot as shown in Figure 3.

A number of designs were considered, including using linear variable differential transformers (LVDT's) and strain gauges, but the simplest approach was to use variable potentiometers with flexible wire (Figure 5). A simple circuit that includes an analogue to digital converter allows us to roughly calibrate the whiskers to give 6 bits of information. This is translated into an integer between -32 and +32, where 0 is roughly the centre and negative values represent backward deflection.

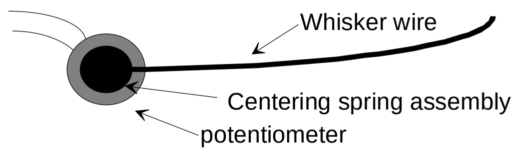

Figure 5 - Proportional whisker using a rotational variable resistor

[1] Partially re-drawn from [Russell 90] pp103.

2.1 Detecting Landmarks

In order to successfully navigate around the laboratory using the scheme outlined in section 5 below, the robot must be able to identify some higher level aspects of its environment. In particular the robot is required to sense a number of landmarks. These include doorways, straight wall sections, convex and concave corners, table legs and chairs.

The architecture utilises a shallow hierarchy of feature detectors which obtain input from the sensors, other feature detectors and behaviour units to determine the presence of a specific feature. The a portion of the hierarchy is shown below. The front and back whiskers each feed sensory data into processes that detect fast deflection and release of the whiskers. A deflection and release in sequence signals an 'impulse', that is, what happens to the whisker as it passes a table leg at speed, for example. An impulse from the front and then back whiskers signals that the leg of a table or other equipment has passed. The expected delay between impulses is dependent on the robot velocity.

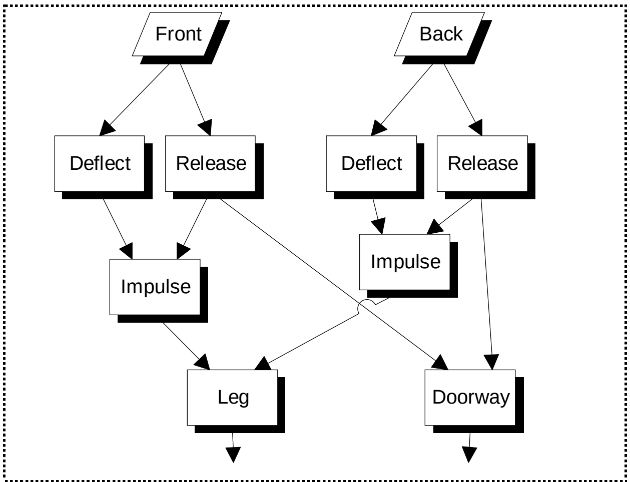

Figure 6 - Part of the landmark detection architecture

When wall following, the release of the front whisker followed by the release of the back whisker signifies the end of the wall segment, for example a doorway or convex corner. The actual hierarchy used in the robot has many more interconnections with other behaviour units and feature detectors for other sensors (odometry and ultrasonic sensors). For example, the doorway detection is active only if the robot is currently wall following.

Figure 7 - Top front view of approaching sweep

Using this arrangement the robot's inclination to the wall is represented very directly by the difference in the readings front and back whiskers, and the distance is proportional to whisker deflection. Hence a feedback controller can be constructed to keep the robot moving straight along the wall at high speed and close range.

Since the robot must avoid obstacles when wall following or navigating the laboratory, it requires some sensors to detect objects in front of it. Again the ultrasonic sensors proved to be inappropriate in all the required circumstances. Objects that are low, such as the wheeled base of office chairs and human feet are not detected by the ultrasonic sensors that are mounted 40cm from the ground. The our solution involved mounting identical whisker sensors on the front of the robot, one on each side. The wire is bent down to give a large vertical cross-section. These work well as sophisticated bump sensors, but also provide extra information about the position of obstacles. The robot can determine, for example, when approaching a wall very steeply, hence deflecting a front whisker on one side, on which side the wall lays, and hence which way the robot should turn to begin following.

3. WALL FOLLOWING

3.1 Straight Segments

In the spirit of reactive behaviour based robotics, we did not implement an algorithm that performs complex trigonometric calculations to determine the amount by which the trajectory should be adjusted, given the whisker deflection. Rather our initial approach was very simple. We maintain two rules that together ensure the robot follows the wall at a specified distance without explicitly calculating the wall range.

$$\theta_{adjust} = p\left(w_{front} - w_{back}\right) \quad (1)$$

$$\theta_{final\_adjust} = \theta_{adjust} + \left(w_{front} - w_{dist}\right) \quad (2)$$

The first rule maintains a zero inclination to the wall while the second rule maintains a specified distance to the wall. The distance it specified implicitly via w_{dist}, which is the whisker value corresponding to the distance to be maintained. The calculation was performed and adjustments sent to the locomotion system at 10Hz. This simple method works surprisingly well, however it is difficult to choose a constant p that is a suitable compromise between enough damping to eliminate oscillations during close following (steady state) and high responsiveness to turn the robot during a sharp approach to the wall before it collides. It would be impossible to achieve even this level of control using simple binary whiskers.

These limitations can be overcome by implementing a simple standard proportional integral differential (PID) controller [Bennett 88]. It has the general form:

$$\theta(t) = K_c T_d \frac{de(t)}{dt} + K_c e(t) + \frac{K_c}{T_i}\int e(t)dt \quad (3)$$

The discrete form is:

$$s_n = s_{n-1} + e_n \quad (4)$$

$$\theta_n = K_p e_n + K_i s_n + K_d\left(e_n - e_{n-1}\right) \quad (5)$$

where e_n is the error in inclination angle, K_p, K_i and K_d are the proportional, integral and differential parameters respectively. This approach overcame some of the limitations of the simple approach.

Access to the Yamabico locomotion system is via a high level command interface, which unfortunately proved to be a limitation for any scheme that requires trajectory corrections to be made at high frequency. For this reason we were forced to implement another control technique that, although not in keeping with the behaviour based philosophy, works well with the limitations imposed by the locomotion interface. The Yamabico locomotion system was designed according to the classical *sense-plan-act* model in that no access to low-level wheel velocity or shaft-encoder information is available to higher level components - such as our wall follower. These are abstracted by an interface that accepts an equation of a line in global coordinates and uses a PID controller to track the robot along that line.

The control algorithm we are now successfully using samples the front and back whiskers at 30Hz. From these values it calculates the distance from the robot centre to the wall. The orientation could also be calculated, however the sensor readings are too noisy for a useful result. A history of these wall distance sample points are used to incrementally update a standard straight line regression fit to the points. This line is then used to apply a translation and rotation correction to the line on which the locomotion system controller is currently tracking, at the slower rate of 10Hz.

There are a number of other control paradigms we could have employed, such as Fuzzy Controllers. Fuzzy controllers are most useful when the system model is not fully understood. In our case we have a good model and the algorithm we are using works within the limitations of the Yamabico locomotion system interface.

3.2 Cluttered Walls

Rarely there are many clear straight wall segments in a typical office or laboratory environment. Hence, when the robot is following along the wall it will approaches obstacles in front of it that the wall following behaviour will not negotiate, such as chairs, tables, boxes, people and other equipment. In this case the front whiskers make contact first. As will be described below a number of interacting behaviours are utilised to negotiate obstacles in this situation, but each relies on the information from the front whiskers. In particular the 'stop reflex' causes the robot to stop with maximum deceleration as soon as either of the front whiskers are deflected.

4. THE ARCHITECTURE

In this section we discuss the general architecture we are developing for supporting the construction of behaviour based agents and the current implementation of this architecture. The following section details how this architecture was used to organise a specific system of behaviour units to navigate using whiskers.

As mentioned in the introduction we have selected the behaviour based approach because complex behaviour is very difficult to achieve using the purely reactive approach and the planner based approach doesn't have the reactivity we require.

There are a number of key properties we consider important in an architecture at all levels.

- *Prediction* - a system should always predict future utility of action and hence sensory input.
- *Learning* - a system should modify behaviour to maximise perceived utility when it encounters novel stimuli (stimuli not predicted).
- *Memory* - a system should continue to improve predictive ability based on learned information.

We are currently designing and implementing an architecture based on these principles called ABBA (*Architecture for Behaviour Based Agents*).

The current implementation consists of three levels of abstraction. Each is implemented as a C++ language class hierarchy building on the previous. The first level is the platform abstraction layer (PAL) which simply serves to provide an object oriented and platform independent interface to the underlying operating system services. It provides an abstraction called an 'ActiveObject' which is essentially an object which executes its own thread of control. The PAL also provides I/O, uniform networking, ActiveObject management and inter-ActiveObject communication and synchronisation. The PAL interfaces a custom operating system called MOSRA on the robots and UNIX on the host system.

The second layer provides facilities for creating, connecting together and composing ActiveObject's into a distributed concurrent network of communicating behaviour units. A behaviour unit can also consist of a compound of nested behaviours. The third layer adds the ability to compose behaviour by interconnecting units in more abstract ways. For example, by providing activation potentiation signals with sensory data flow, as described briefly in section 5.2 below, and enforcing potentiation currency conservation [Gallistel 80]. The details of this layer are still subject to change as we are experimenting with different paradigms. More information will appear in future publications.

5. NAVIGATION

The implementation of a specific organisation of behaviour units to realise a whisker based navigation behaviour is being conducted in two steps. In the first step was to construct a reactive type system that is capable of simple 'navigation' around the perimeter of the laboratory. This system has no internal map, no memory and doesn't learn. It simply uses the wall following behaviour until an obstacle is contacted, in which case the robot reverses and turns right. If the 'wall' ends the robot naively attempts to turn hard left. This unintelligent behaviour will following the perimeter in an inefficient manner, bumping all obstacles with the

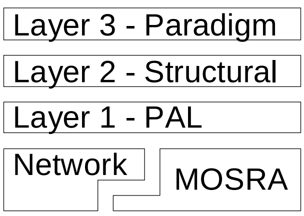

Figure 8 - ABBA layers on the Robot

front whiskers and also entering into doorways. The second step is to add the ability to learn an internal

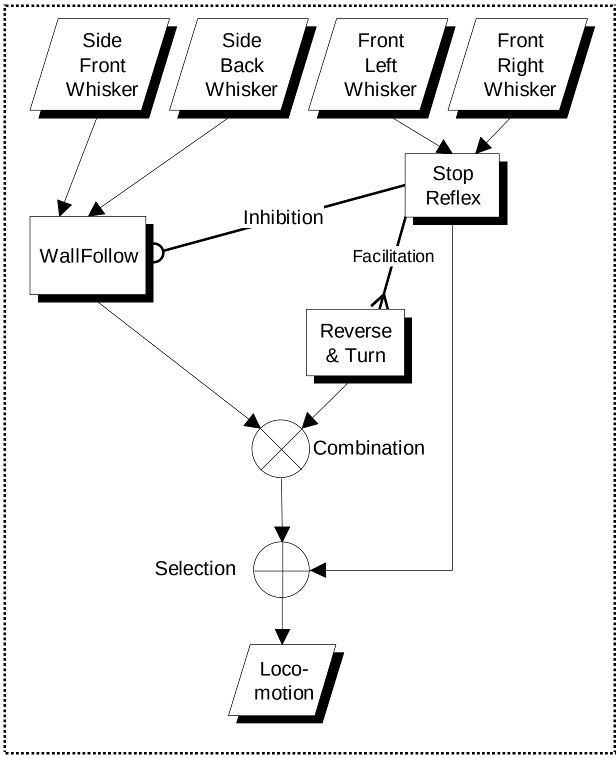

Figure 9 - Behavioural organisation for simple perimeter wall following

representation of the laboratory (a map) and purposefully navigate using it.

5.1 Reactive navigation

This system has initially been implemented on layer 2 and partially layer 3 of ABBA. The implementation is being used as a valuable exercise to provide information for input into the design of layer 3.

Figure 9 shows a simplified subset of the behaviour units and their interconnections used for the reactive component of navigation. This architecture enables the *sweep* to follow around the perimeter of the laboratory. It follows straight wall segments until the front whiskers are contacted. It then stops due to the triggering of the 'stop reflex'. This in turn activates the 'reverse & turn' unit while simultaneously de-activating the wall following behaviour. The result is that the robot reverses and turns right (since the wall following whiskers are on the left of the robot). It then proceeds forward again. This usually causes re-collision with the same obstacle, but repeated reversing and turning eventually negotiates the obstacle. This is a simple non-cognitive behaviour that can then be built upon to provide purposive behaviour as described in the next section.

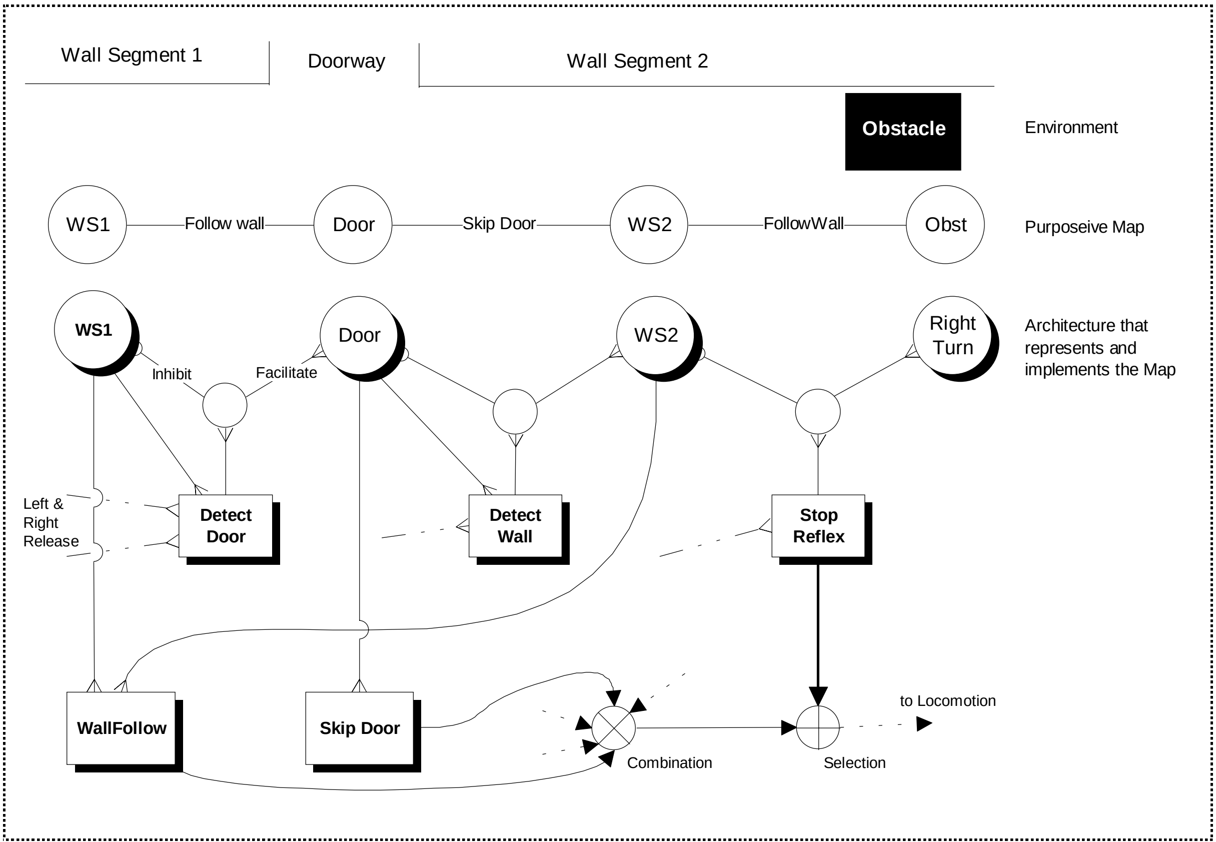

Figure 10 - Portion of an example architecture for purposive navigation along the wall shown

5.2 Purposeful navigation

The reactive part of our navigation system does not always navigate the whole perimeter of the laboratory; for example if placed on an interior isolated wall segment, the robot will forever follow around it without crossing the room to sweep the inside of the exterior walls. The 'reverse & turn' behaviour for negotiating obstacles is also inefficient. For this reason the robot requires the ability to learn a representation of its environment.

In previous work Zelinsky reported how this may be achieved using a resource called a *Purposive Map* (PM) [Zelinsky 94]. The PM incorporates navigation and learning of behaviour based landmarks. In this research we use a variation of the PM. The PM is a graph where the nodes represent points in the environment that can be sensed. In this case we can sense doorways, convex and concave corners, wall segments, chair and table legs and obstacles along a wall using only the whisker sensors. These landmarks are sensed using a feature detector hierarchy of which a subset was shown in Figure 6 for detection of legs and doorways. The links in the graph represent the behaviour that should be activated to proceed from the current node to the next node closer to the goal. The purpose can be achieved, for example, by selecting a landmark as the goal to navigate toward. The behaviour outputs are arbitrated by selecting the behaviour associated with the current link. This simple system can learn the environment by adding new nodes and links to the graph as new landmarks are detected.

One undesirable feature of the PM is that is employs a centralised arbiter for behaviour selection. For the current research we are experimenting with distributed implementations of the PM idea in layer 3 of the ABBA architecture. The diagram above shows an example of a portion of the environment, the PM that represents it for the *sweep* and the architecture used to navigate using it.

The details of the general ABBA architecture are beyond the scope of this paper. Briefly each behaviour unit, the shadowed squares and circles in the diagram, has a potentiation currency it. It may receive currency from other units via facilitation or loose currency via inhibition from other units. It may also spend its currency by facilitating other units or gain currency by inhibiting other units. The combination operator simply takes the weighted sum of the actuator signals in some appropriate way based on the level of potentiation. The selection operator gives a hard-wired preference to one input, the stop reflex in this example. The diagram does not show all the complex connections between these and other units in the system that are not shown.

6. RESULTS

Using the proportional whiskers and the wall following algorithm presented above, the current implementation can wall follow at 500mm/sec, 30mm from the wall.

The architecture developed for navigation currently adds the ability to purposively navigate a predetermined path around the laboratory while detecting and avoiding table and chair legs.

CONCLUSION & FUTURE WORK

We have shown that high speed and close wall following can be achieved using our unique proportional whisker sensor. We have also shown that high reactivity can be combined with cognitive level purposive navigation using the whiskers in a uniform behaviour based architecture. This is achieved by using a distributed, learnt and physically grounded representation of our laboratory environment rather than a symbolic or geometric model. Future work will concentrate on improving the navigation capabilities and the free space obstacle avoidance behaviour based on previous work by Cheng and Zelinsky [Cheng 95]. This experimental work has been used to determine the requirements for a general behaviour based architecture. The ABBA architecture is still evolving.

ACKNOWLEDGMENTS

This work was made possible by the generous support of Fujitsu Japan who manufacture the MEP tracking vision system and Wind River Systems suppliers of the VxWorks® operating system. Thanks are extended to Michael Milway for designing the whisker electronics and building the hardware, and Andrew Russell of Monash University for helpful ideas.

REFERENCES

[Bennett 88] Bennett, Stuart, "**Real-Time Computer Control: An Introduction**", *Prentice Hall International series in systems and control engineering*. Ed. M. J. Grimble. 1988.

[Brooks 87] Brooks, Rodney A. and Connell, Jonathan H., "**Asynchronous Distributed Control System for a Mobile Robot**", *SPIE's Cambridge Symposium on Optical and Opto-Electronic Engineering Proceedings*, Vol. 727, October, 1986, 77-84.

[Cheng 95] Cheng, Gordon and Zelinsky, Alex, "**A Physically Grounding Search in a Behaviour Based Robot**", *Proc. Eighth Australian Joint Conference on Artificial Intelligence 13-17 November,1995, Canberra*. Ed. Xin Yao. ISBN 981-02-2484-2.

[Everett 88] Everett, H. R., "**Survey of Collision Aviodance and Ranging Sensors for Mobile Robots**", *Tech. Report No. 1194, Naval Command Control and Ocean Surveillance Center*, San Diego, CA, 29 March, 1988.

[Fearing 90] Fearing, R. S., "**Tactile Sensing Mechanisms**", *The International Journal of Robotics Research*, Vol. 9, No. 3, pp 3-23, June 1990.

[Gallistel 80] Gallistel, C. R., "**The Organization of Action: A New Synthesis**", *Lawrence Erlbaum Associates*, Publishers, Hillsdale, New Jersy, ISBN 0-89859-009-4.

[Inoue 85] Inoue, H. *et. al.*, "**A Flexible Multi-Window Vision System for Robots**", *Proceeding of 2nd International Symposium on Robotics Research*, MIT Press, 1985.

[Jayawant 89] Jayawant, B. V., "**Tactile Sensing in Robotics**", *Journal of Physics*. E: Scientific instruments, Vol. 22, pp. 684-692, 1989.

[Jung 96] Jung, David and Zelinsky, Alexander, "**Cleaning: An Experiment in Autonomous Robot Cooperation**", *Proceedings of the 1996 International Symposium on Robotics and Manufacturing* (ISRAM '96).

[Mataric 92a] Mataric, Maja J., "**Behavior-Based Systems: Key Properties and Implications**" in *Proceedings, IEEE International Conference on Robotics and Automation*, Workshop on Architectures for Intelligent Control Systems, Nice, France - May 1992, 46-54.

[McFarland 91] McFarland, David, "**Defining motivation and cognition in animals**", *International Studies in the Philosophy of Science 5*, pp153-170, 1991.

[McFarland 93] McFarland, David and Bosser, Thomas, "**Intelligent Behavior in Animals and Robots**", *The MIT Press*, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142, ISBN 0-262-13293-1.

[Nicholls 89] Nicholls, Howard R. and Lee, Mark H., "**A Survey of Robot Tactile Sensing Technology**", *The International Journal of Robotics Research*, Vol. 8, No 3, pp. 3-29, June 1989.

[Pfeifer 95] Pfeifer, Rolf, "**Cognition - Perspectives from autonomous agents**", *Robotics and Autonomous Systems 15*, pp47-70, 1995.

[Russell 85] Russell, Andrew R., "**Object Recognition Using Articulated Whisker Probes**", *Proceedings of the 15th ISIR*, Tokyo, pp. 605-11, 1985.

[Russell 90] Russell, Andrew R., "**Robot Tactile Sensing**", *Prentice Hall*, ISBN 0 13 781592 1.

[Yuta 91] Yuta, S., Suzuki, S. and Iida, S., "**Implementation of a small size experimental self-contained autonomous robot - sensors, vehicle control, and description of sensor based behavior**", *Proc. Experimental Robotics*, Tolouse, France, LAAS/CNRS (1991).

[Zelinsky 94] Zelinsky, A., Kuniyoshi, Y. and Tsukune H., "**Monitoring and Coordinating Behaviours for Purposive Robot Navigation**", *Proceedings of Intelligent Robotic Systems Conference (IROS)*, Munich, Germany, Sept. 1994.

[Zelinsky 95b] Zelinsky, Alex and Cheng, Gordon, "**Incorporating real-time vision into robotic tasks**", *Proceedings of the Joint Australian-Korea Workshop on Manufacturing Technology (KOSEF)*, Published by the Australian Academy of Science, November 1995.