TRANSPUTER CONTROL AND SENSING ARCHITECTURES FOR MOBILE ROBOTS

P.J.Probert and H.Hu [1]

1 Introduction

At Oxford we have a programme involving many aspects of mobile robotics including vision, sensor integration, path planning and architectures [1]. The work in architecture is based primarily around a small prototype of a free ranging vehicle for factory automation marketed by Caterpillar, which uses a low level controller developed by GEC. This vehicle, which is on loan from GEC and measures 0.9 x 1.2 m with a platform 0.6 m high, is equipped with a variety of sensors including sonar, trinocular stereo and a sophisticated laser range finder. Our other vehicles include a French Robosoft platform, a tracked vehicle, on loan from RARDE, and a small home-made mobile platform. Although the work described here is aimed primarily towards the GEC robot, it is designed to be applicable to any of the vehicles.

We have adopted a distributed transputer architecture for sensing and control in these projects. The transputer provides us with a powerful on-board facility and adapts easily to a distributed reconfigurable sensing and control strategy. A generic transputer architecture for sensors with local intelligence allows us to build networks of distributed sensors and opens an integrated path for combining the information of disparate sensors [2]. In control a distributed architecture overcomes the bottlenecks associated with a centralised controller. The use of transputers throughout gives a consistent interface for easy communication and interfacing between the various units in the sensing and control strategy.

2 The GEC *Turtle*

The main thrust of the work on control is on the GEC factory robot, the *Turtle*. This vehicle was originally designed to operate through a radio link under the central control of a 'land-based' computer, in a known environment with no sensory control except for odometry. For this purpose it has on-board a powerful low level capability in odometry, path smoothing and motor control. This is realised by a number of microprocessors co-ordinated by a GEM-80 industrial cyclic controller.

At the original on-board level of control, path points are stored in a data structure called the *path buffer*. The buffer is filled by the path smoothing module which is provided with sparse points by the land-based computer. A new point is sent from the buffer to the motors every GEM-80 cycle time (every 80 msec).

3 Layered Architecture

Above this controller and replacing some aspects of it we have implemented three layers of an architecture based on Brooks's subsumption ideas [3]. The layered controller communicates with the low level controller through an intelligent communications interface (Figure 1).

Our controller provides the robot with exception handling capabilities in a factory environment. The scenario is as follows. The robot moves down aisles between from a docking point to pick up a pallet. Although there are a number of alternative routes, they are few in number. Occasionally aisles may be blocked; for example by a pallet shed by an earlier robot. These blockages may be temporary or they may last for some time. The robot is to replan a path to continue its task, taking a safe route past the obstacle. Although a very specific behaviour pattern has been chosen, the same architecture could provide a diverse range of capabilities.

[1] The authors are with the Robotics Research Group University Department of Engineering Science, Oxford OX1 3PJ

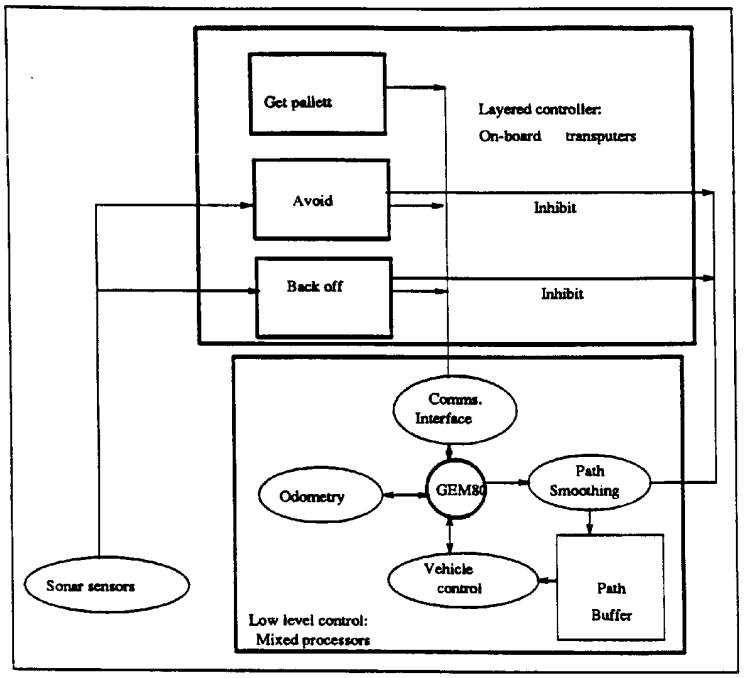

Figure 1: The on-board control

The *get pallet* layer of the controller is simply implemented as a command to tell the vehicle to go to a pre-determined position to pick up a pallet; sparse path points are sent to the communications interface where they are intercepted by the path smoothing module. Eventually more sophisticated path planning based on camera images will take place at this layer. Using a ring of 12 Polaroid sonar sensors, the *avoid* layer provides the capability to move around obstacles and the *backoff* layer provides selfpreservation in backing away from a moving obstacle [4] whilst this route is being followed. The lower layers subsume the higher in writing new path points directly to the *path buffer*, calculating the new path as the robot moves.

Although much of the design philosophy is similar to subsumption, especially in minimising communications between layers and in asserting the independence of layers, our robot has a definite aim in life, provided by the highest layer. We are less interested in the ideas of building up complex from simple behaviours than in realising an efficient control strategy. In addition, in our strategy, the *inhibit* mechanism is passed from the lower layers to the upper ones.

4 On-board Hardware

The current on-board hardware is shown in Figure 2. Two transputers on the robot realise the control and sensing outlined above.One deals with the obstacle avoidance and selfpreservation strategies. The other provides the highest level control and is interfaced to a radio link to allow development of more advanced planning algorithms. Another transputer handles the sonar sensors, both the transducers and the motors on which they are mounted. Further transputers will be added as greater complexity is brought into the algorithms to implement each layer. All sensing and replanning takes place whilst the vehicle is moving at its normal

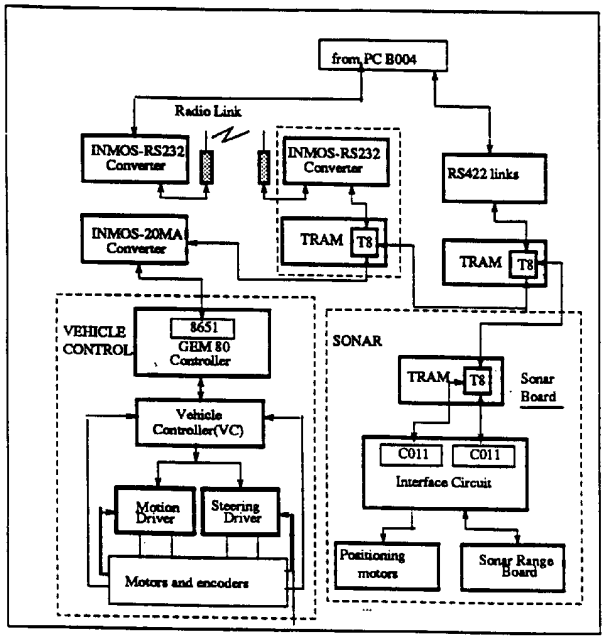

Figure 2: The Hardware Architecture

speed of around 0.5m/sec (the vision systems are not included at present in this real time strategy).

Development has been largely using TDS on PCs to date, although a Niche board installed in a Sun workstation is now being used as well. Occam is used throughout.

5 Conclusions

The project has just completed its first phase in realising a simple on-board control strategy. The use of transputers throughout provides a unified framework for a modular approach to development. Our current work is developing the *avoid* layer of the layered controller. Simple algorithms have been developed as we will describe in the Colloquium. Our next stage of work is in developing the sensing and algorithmic strategies.

References

[1] J.M.Brady, H.Durrant-Whyte, H.Hu, J. Leonard, P.J.Probert, and B.S.Y.Rao. Sensor-based control of agvs. *IEE Computing and Control*, 1989.

[2] H.F.Durrant-Whyte, S.Grime, and H.Hu. A modular, decentralised architecture for multi-sensor data fusion. In *Transputer Applications Conference*, 1990.

[3] R.A. Brooks. A layered intelligent system for a mobile robot. In *Third Intl. Symp. Robotics Research*, 1985.

[4] M.D.Adams, H.Hu, and P.J. Probert. Towards a real time architecture for obstacle avoidance and path planning in mobile robotics. In *IEEE Conference on Robotics and Automation, Cincinnati*, 1990.

[5] T.Hague, J.M. Brady, and S.Cameron. Using moments to plan paths for the oxford agv. In *Proc. IEEE Conference of Robotics and Automation*, 1990.