Technical Report DHPC-121

An Architecture for Swarm Robots

K.A.Hawick, H.A.James, J.E.Story and R.G.Shepherd
Computer Science Division, School of Informatics
University of Wales, Bangor, North Wales, LL57 1UT, UK
{hawick,heath}@bangor.ac.uk
Tel: +44 1248 38 2717, Fax: +44 1248 36 1429

The area of collective and cooperative behaviour in swarm robots is an exciting field of study. Recent technological and economic developments have made it possible to build medium to large scale swarms of quite sophisticated mobile robots. We review recent mobile computing technologies and describe our architecture for a swarm of tri-wheeled robots that can exchange information via wireless architecture as well as through their onboard sensors. The main objectives of our system are for educational purposes and as a development platform for an intelligent software middleware to coordinate the activities of a swarm. We also discuss some of the experiments and behavioural goals we are able to study with our system.

Keywords: robot swarm, architecture; mobile computing; wireless communications.

1 Introduction

Although very sophisticated robots have been built by many groups and a great deal of experimentation and work has been done in designing architectures for single mobile artificial intelligent robots[2], less work has been carried out on swarms of intelligent systems[12]. This is partially due to the cost of producing a large number of smart robot systems and partially due to the difficulties in assembling a mobile communications infrastructure for experiments with swarms.

We have attempted to design and build a medium powered mobile robot that is cheap enough to mass produce and hence assemble an interacting swarm. We were also motivated by educational needs and it was useful to employ a very modular design that undergraduate and postgraduate students could work on independently. Some of our early work was based on combining the Lego mindstorms control brick[6] with other simple electronics and sensors to make a prototype swarm robot that could communicate via infra red. We found that the processing power of the Lego brick was adequate for simple experiments but was too simple for the sorts of inter robot interactions we wished to trial. In addition the cost of the Lego control brick makes individual nodes still rather too expensive to mass produce. The infra red communications system was adequate only for very short range communications and we wished our robots to be able to work together over distances of around 100metres (ie over car park sized areas).

In view of these limitations we therefore looked into various mechatronics and electronic commodity components to find a cheap and modular solution. We eventually chose a combination of Microchip's Peripheral Interface Chip (PIC)[14] and Dallas Semiconductor's Tiny InterNet Interface (TINI) control board[5]. We describe these components alongwith our architecture in section 4.

We experimented with various Lego and Meccano and scrap parts for building the mechatronics platform for our swarm robot. These experiments were fun but very labour intensive and something of a distraction from the software design and experiments we wished to work on. We later discovered a toy robot system that offered mass produced parts at a low cost and which was compatible with our electronics drivers. The Cybot [15] toy system is marketed by Eaglemoss Publications and is sold as a kit alongwith the magazine “Real Robots”. The cybot mechatronics is based on work by Kevin Warwick’s group at Reading University on their “Seven Dwarves” robots. We have been able to modify the sensors and augment the electronics control system of the Cybot to arrive at our swarm robot architecture.

We are using two different wireless communications systems for or robot swarm. One is based on IEEE802.11b wireless ethernet[11] and uses a conventional TCP/IP stack for communications between our robot nodes. Another is the emerging Bluetooth standard[1] for which we are developing our own interfaces. We would ideally like to enable our swarm robot nodes with a differential global positioning system (GPS) to allow them to communicate their positions to each other and to a base station. GPS systems are not quite small enough and cheap enough for the sort of accuracy we require. We anticipate this will change in the next few years but in he meantime we are experimenting with a pseudo satellite positioning system based on an overhead camera that uses pattern recognition to identify individual robot nodes and which can broadcast their relative positions via the wireless network.

Figure 1 illustrates our concept for an experimental swarm. We are working with swarm sizes of between eight and sixteen nodes at present. These sizes are constrained by our present budget but we believe our architecture and control algorithms scale to perhaps a factor of four more.

Apart from generating considerable student enthusiasm for control systems and robotics in general our system is proving a useful platform for distributed computing software development. We are investigating the use of various leader election algorithms a and consensus agreement algorithms to allow the robot swarm to cooperate on tasks. In section 2 we describe

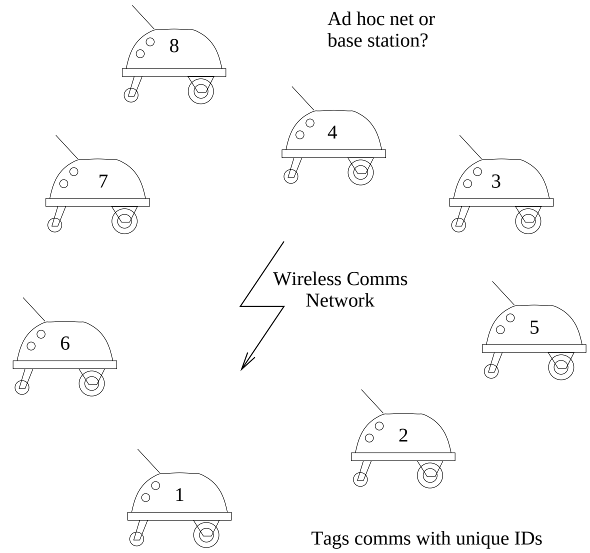

Figure 1: A Swarm of modified cybots communicating using a wireless network.

some of the cooperative tasks and behaviours, such as: searching; retrieval; hunt/defend; and redundant nodes, that we are investigating.

One attractive feature of our swarm node architecture is that each node can be programmed at a relatively high level in Java and that distributed behaviours and control policies can be encoded and exchanged amongst robot nodes. In this paper we concentrate on the overall architecture but do give some details of the software in section 6.

2 Tasks and Behaviours

There are a number of relatively simple but interesting behaviours that can be programmed with a swarm of robots. We have experimented with simple avoidance and attraction behaviour using sensed information from on board ultrasonic sensors. Some interesting clustering behaviour is observed amongst swarm nodes, which we are attempting to understand better by a simulation[8]. Figure 2 illustrates one of our behavioural study goals. We wish to order our swarm to march in formation keeping a precise topological relationship and separation between nodes at all times.

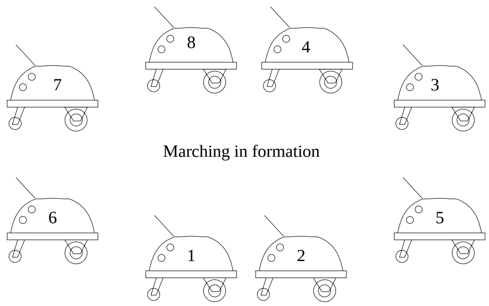

Figure 2: Eight Swarm Cybots marching in formation.

Our early attempts at achieving this through the use of local proximity information were unstable. The use of the pseudo global positioning system information as described in section 5 allow swarm nodes to lock onto specific neighbours and to provide periodic corrections to their positions. Two strategies have been promising to investigate. Firstly a leader node is chosen and other robots in the swarm lock onto that and adjust positions accordingly. Secondly the master satellite/and base station acts as the leader and instructs nodes how to move accordingly.

An associated problem to that of marching in formation is to line up in formation from a random start (as shown in figure 1 for example. We are investigating various stochastic relaxation algorithms for this.

Other behaviours and task we wish to investigate are that of searching and retrieving. We are constructing simple magnetic actuators for our robots using surplus switching relays. Together with magnetic Hall-effect sensors these will allow our robots to find and recover small steel objects. We are also investigating redundancy algorithms to allow a spare node in the swarm to take over if one other fails. This is particularly interesting if the failed node is the leader for some task. We are investigating consensus agreement and leader election algorithms, from our more familiar discipline of distributed computing, to construct behaviours for our swarm[9].

3 Component Modules

Figure 3: Real Robots "Cybot" Mechatronics.

In this section we review some of the component parts we have used for our swarm robot nodes. Figure 3 shows the basic mechatronics unit adopted from the Cybot system. The unit is around 25 cm in length and is based on the classic two drive wheel plus third balance wheel design. In modifying the electronics and sensor configurations we have had to relocate the batteries and use re-chargeable lithium-ion battery packs. The original cybot had two front facing ultrasonic sensor pairs. We have added two more rear facing which gives the unit an almost surround proximity sensor range. We have also added wireless antennae for both WaveLAN and Bluetooth wireless communications systems.

Figure 4: Dallas Semiconductors Tini Board

Figure 4 shows the Tiny InterNet Interface (Tini) control boards. These useful devices have 1 MByte of memory and onboard Ethernet, 1-wire and Inter

IC communications systems. The Tini board runs a small Unix like shell which users can telnet into from the communications ports. A Java Virtual Machine is supported and we employ this for running our high(ish) level control program. This software can communicate with motor and sensor drivers using the Inter IC protocols. We use Peripheral Interface Chips (PIC) for controlling the motors and magnetic actuators. We are also working on a small and cheap camera frame-grabber that will allow the robot nodes an onboard image processing/recognition capability.

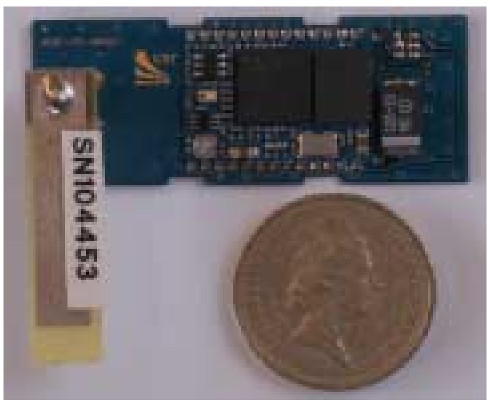

Figure 5: Bluetooth antennae module

Figure 5 shows the Bluetooth radio module from Cambridge Silicon Radio[3]. The device is shown next to a UK One penny coin (diameter 20mm). We are presently working on interfacing the Bluetooth protocol stack to a PIC microcontroller. We have employed preliminary experiments using the Bluetooth communications protocol using a Compaq iPAQ personal digital assistant. These devices come pre equipped with Bluetooth as do many new mobile phones. The Bluetooth protocol is useful as it permits establishment of *ad hoc* communications networks [7]. The WaveLAN IEEE802.11b wireless ethernet technology is somewhat more bulky and power hungry to mount in our small robot nodes but is a useful bridging technology to experiment with.

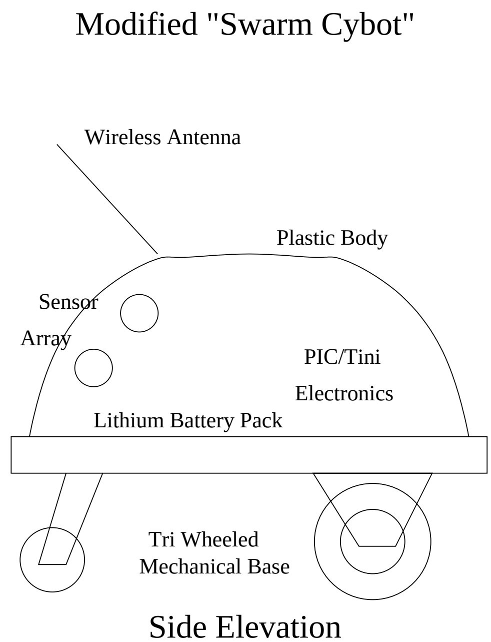

Figure 6: Mechatronics architecture of our modified “swarm cybot”.

4 Swarm Node Architecture

Figure 6 shows the overall architecture of our swarm node mechatronics. Figure 7 shows the block architecture of our swarm node robot. We employ the cheaper series 16 and 17 series PIC chips to drive our motors and magnetic actuators. We are experimenting with the more sophisticated (in terms of memory and interfaces) series 18 PIC chip to interface to a frame-grabber and camera assembly. Although there are several good camera systems available we require one that is both light enough and cheap enough that we can employ one on all our swarm robot nodes. It may be however that we design special purpose “seeing class” nodes that are camera enabled.

5 Positioning System

Figure 8 shows our experimental robot pen with overhead camera that can track individual robot nodes. This is our alternative to a proper global positioning

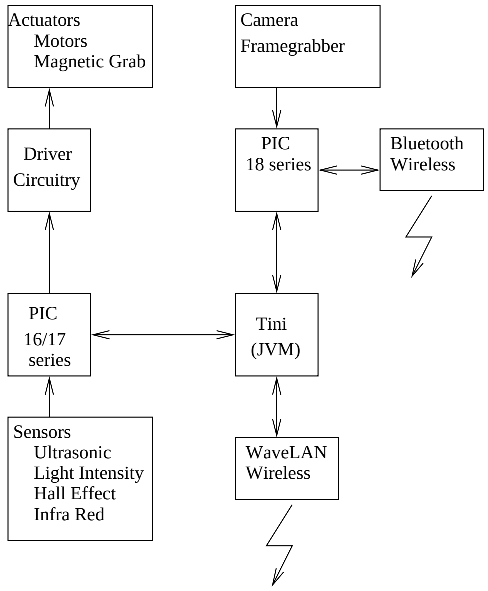

Figure 7: Control system block structure for modified Swarm cybot node.

satellite system receiver that is too heavy, too power hungry and too expensive for the resolution we require. We experimented with a simple rectangular grid patter on the linoleum base of the pen but this proved unnecessary as the robots and pen itself can be readily recognised and a positions to within a few centimetres determined. This is entirely adequate for our present experiments.

Figure 8: Cybot Pen with Webcam "positioning satellite" above.

6 Software Architecture

We describe our software architectures for the robot swarm in more detail elsewhere[9]. In summary, the PIC devices are programmed in what are essentially small finite state machines using C code downloaded to them as firmware. The Tini board can run a small Unix like shell and we used this to run development programs on a local Java Virtual Machine (JVM). The Tini JVM is capable of networking and multi threading code but is not able to run object serialisation nor class introspection codes. This was unfortunate as we hoped to make use of middleware we had developed in Java for remote execution on fixed networks [10, 13]. We have been able to modify our message based system to use very simple point to point algorithms so avoiding the need for object serialisation.

The Tini systems onboard protocol stack allows us to use the wireless ethernet directly from inside the normal Java TCP/IP stack packages. We are presently developing our own interfaces to the Bluetooth wireless protocol stacks however.

7 Discussion and Conclusions

We have described our robot swarm architecture and the relatively commodity priced components we have used. Table 1 summarises the cost per node to build a system like ours.

Component	Cost (UK Pounds)
Cybot Mechatronics	42.00
PIC controller	7.00
Tini Board	70.00
Misc Sensors	12.00
Misc Electronics	6.00
WaveLAN Wireless	110.00
TOTAL Cost per Node	247.00

Table 1: Approximate component costs for our Swarm Robot Node

We believe this is a worthwhile yet affordable design that is enabling us to experiment with distributed algorithms and behaviours in the field of distributed robotics and AI.

We hope to develop a more elaborate teleoperation infrastructure that interoperates with our swarm as well as developing more sophisticated onboard intelligence for individual nodes.

The tradeoff in our design has been that between having a large enough number of nodes and having more sophisticated individual nodes. We believe we have found a useful compromise point but do wish to continue to develop more powerful individual robot nodes. This is dependent upon decreases in price in embedded microcontroller devices or small PCs and improvements in their capabilities and power consumption requirements.

Acknowledgements

The Distributed and High Performance Computing Group is a collaborative venture between the University of Wales, Bangor and the University of Adelaide in South Australia.

References

[1] Bluetooth. The Bluetooth Core Specification. Available from http://www.bluetooth.com/-dev/specifications.asp Last visited August 2002.

[2] Rodney A. Brooks, "Robot - The Future of Flesh and Machines", Pub Allen Lane The Penguin Press 2002, ISBN 0-713-995-017.

[3] Cambridge Silicon Radio. Casira Bluetooth Software and Hardware Development System. Available from http://www.csr.com Last visited August 2002.

[4] Compaq. *iPAQ* Pocket PC. Available from http://athome.compaq.com/showroom/static/-iPAQ/handheld_jumppage.asp Last visited August 2002.

[5] Dallas Semiconductors. "Tiny InterNet Interface" Available from www.ibutton.com/TINI Last visited September 2002.

[6] Benjamin Erwin, "Creative Projects with Lego Mindstorms", Pub Addison Wesley 2001, ISBN 0-201-70895-7.

[7] K.A.Hawick and H.A.James, "Middleware for Wireless Sensors and Robots", Technical Note DHPC-112, August 2002, Submitted to IEEE Int Conference on Wireless Communications and Networks (WCNC 2003)

[8] K.A.Hawick and H.A.James, "Simulating Swarm Behaviour of Robots", Technical Note DHPC-118, September 2002, Submitted to IASTED Conference on Applied Modelling and Simulation, (AMS2002)

[9] K.A.Hawick, H.A.James, R.G.Shepherd and J.E.Story, "Distributed Computing for Robotics", University of Wales, Bangor, Technical Note, DHPC-114, August 2002.

[10] K.A. Hawick, H.A. James, A.J. Silis, D.A. Grove, K.E. Kerry, J.A. Mathew, P. D. Coddington, C.J. Patten, J.F. Hercus, and F.A. Vaughan, "DISCWorld: An Environment for Service-Based Metacomputing," *Future Generation Computing Systems (FGCS)*, 15:623–635, 1999.

[11] IEEE. Standards for local and metropolitan area networks: Standard for port based network access protocol. *IEEE Draft 802.1X/D11*, March 2001.

[12] Andrew H. Jones, Guilherme Nelson DeSouza and Avinash C. Kak, "A Multi-processing Software Infrastructure for Robotic Systems", Proc 2001 IEEE Int Conf on Robotics and Automation, Seoul, Korea, May 21-26 2001.

[13] J.A. Mathew, H.A. James and K.A. Hawick, "Development Routes for Message Passing Parallelism in Java," *Proc. ACM 2000 Java Grande Conference*, San Francisco, June 2000, pp 54-61.

[14] Microchip, "PICmicro microcontroller", Available from http://www.microchip.com Last visited September 2002.

[15] Real Robots. "Cybots" Available from http://www.realrobots.co.uk Last visited September 2002.