

## Coordination of Multiple Mobile Robots via Communication

Huosheng Hu<sup>†</sup>, Ian Kelly<sup>⊗</sup>, David Keating<sup>‡</sup>, David Vinagret<sup>†</sup>

<sup>†</sup>Department of Cybernetics, University of Reading, Whiteknights,  
PO Box 225, Reading RG6 6AY, UK

<sup>⊗</sup>School of Computer Science, University of West England,  
Coldharbour Lane, Bristol BS16 1QY, UK

<sup>‡</sup>Telecommunications Engineering, University of Politecnia  
de Catalunya, Barcelona, Spain

### Abstract

*Research on the co-ordination of multiple mobile robots has to address three main problems: (i) how to appropriately divide the functionality of the system into multiple robots, (ii) how to manage the dynamic configuration of the system, and (iii) how to realise co-operation behaviour. This paper will concentrate on the third aspect. More specifically, the aim of our research is to develop a team of coordinating mobile robots via effective communication for real world applications. We will describe the methodology to achieve co-operative behaviour, the experimental mobile robots developed, and potential application areas. The developed system is demonstrated by two examples such as flocking and shared experience learning.*

**Key words:** Mobile Robots, Coordination, Communication, Robot Behaviour, Learning.

### 1. Introduction

During last few decades, major research efforts were focused on improving the performance of an individual mobile robot by using advanced sensors, actuators, and intelligent control algorithms. This was mainly driven by the need to perform increasingly complex tasks that required by real world applications. As a result, individual mobile robot has become very sophisticated. Recently, an alternative approach that uses a number of simpler robots to achieve complex tasks via co-operation has been researched [1][2][10][12]. Such an approach offers potentially significant advantages in terms of flexibility of operation that may be achieved, and fault tolerance due to redundancy in the number of mobile robots available[4].

As the number of mobile robots in a system increases, planning and control of the system becomes increasingly complex [5]. The methods to handle such complexity include a centralised control method and a decentralised control method. More specifically, in a centralised control method all planning and decision-making functions are handled by a single control centre. Each mobile robot contains only sensors for localisation and obstacle avoidance, actuators for movements and manipulation, and communication facility for communicating with the control centre. All the movements of mobile robots in the system are controlled from this centre and conflicts among multiple robots are easily solved. This method has been widely adopted in manufacturing industry and warehouses where multiple mobile robots are used to transfer parts and clean warehouses. One major disadvantage of the system is that whole system will stop functioning immediately if the control centre fails.

In contrast, a decentralised control method is to equip each mobile robot with map building, planning and decision-making capability [3]. In any unforeseen situation, the robot is able to plan a new path or

find a solution without waiting for commands from the control centre. The function of the control centre is only limited to the broadcasting of traffic flow information received from all robots and the allocation of tasks in the system. Inter-robot communication becomes necessary since competition for resources should be avoided and sharing experience could improve system performance.

In a decentralised control system, co-ordination of multiple mobile robots needs to address three main issues: (i) how to appropriately divide the functionality of the system into multiple robots, (ii) how to realise the dynamic configuration of the system, and (iii) how to achieve co-operation behaviour [2]. This paper is focused on the third issue, i.e. co-operative behaviour demonstrated by multiple mobile robots, especially in case of no global information about the robot environment. The key to achieve such co-operative behaviour is the development of inter-robot communication mechanism [10].

In the next section we will give a brief introduction of co-operative behaviours and a team of small mobile robots, which were built at Reading during last few years. In section 3, an infrared communication system is presented to show how multiple mobile robots are able to communicate effectively. Two typical examples are given in section 4, which demonstrate the possible applications in the real world. Finally, section 5 gives brief conclusions and future work.

## **2. Co-operative behaviours and mobile robots**

### **2.1 Cooperative behaviours**

A primary aim in the development of co-operative mobile robots is to synthesis cooperative behaviours of multiple mobile robots. Cooperative behaviours enable a team of mobile robots to accomplish missions that cannot easily be achieved with individual mobile robot [12]. Many cooperative behaviours can be synthesised for the coordination of multiple mobile robots based on the reactive behaviour architecture proposed by Brooks [4]. At this stage of our research, we started with the syntheses of the following cooperative behaviours defined as follows:

- Communication behaviour – to realise inter-robot communication and exchange information
- Safeguard behaviour – to keep a safe distance among mobile robots during coordination
- Flocking behaviour – to be able to gather together for co-operation on a particular mission
- Convey behaviour – to transfer an object in a co-operative way, i.e. leader and follower
- Group learning behaviour – to learn from other mobile robots and to share experience together

The communication behaviour is the behaviour that makes co-operation of multiple mobile robots feasible. To achieve mature understanding, a message protocol is required to define so that every member of teams of robots can exchange data and experience effectively. A communication media needs to be selected from those available today, on the market, such as radio, ultrasonic, infrared, etc. The criterion for selection depends on different applications in terms of the number of robots involved, the range of operation, and the interference from environments, etc.

Safeguard behaviour is to protect each robot itself from colliding with other robots or objects. This is extremely important behaviour needed for cooperative operation of multiple mobile robots in a dynamic environment. With such behaviour, mobile robots are able to work closely in a specific mission. The flocking behaviour, on the other hand, enables multiple mobile robots to form a team where each makes its own contribution towards a particular goal. Such behaviour is commonly found in nature where animals or birds flock for protection and goal seeking.

Convey behaviour is useful when a task needs more than one mobile robot. For instance, it would be much better to deploy two mobile robots to move a long box down a narrow corridor in a warehouse, i.e. one is at the front of the box and another is at the back. Another example is the maintenance of nuclear reactors where one mobile robot leads a way to locate the fault component and another

follows to replace it with a new one. In contrast, group learning behaviour aims to enable mobile robots to learn from each other, in a manner that occurs within human society. One of the useful features of group learning is its high learning speed, comparing with individual learning.

Many other cooperative behaviours will be synthesised during the next stage of our research, for instance homing behaviour and foraging behaviour. These are easy to implement since each behaviour is designed in such a way that it can be easily added into or removed from the system. In this paper, we will mainly focus on three cooperative behaviours: communication behaviour, flocking behaviour and shared experience behaviour.

## 2.2 Mobile robots at Reading

Figure 1 presents a team of autonomous mobile robots constructed at Reading during last few years [8][9][10][11]. The robots are equipped with an infrared communication system as well as ultrasonic sensors for detecting obstacles. The obstacle detection system consists of three pairs of ultrasonic sonar transducers in which one pair looking forward, one pair looking to the front-left and the other pair looking to the front-right. This sonar system measures the range to the nearest obstacle in front of the robot. To detect obstacles closely, echoes have to be detected whilst the ultrasonic pulse is still being transmitted, thus requiring a high detection threshold. For the detection of objects far away, a much lower threshold is required to allow for the large signal loss. Thus in order to detect both near and distant obstacles, a time varying threshold system is used. The threshold is initially large, but decreases with time to a pre-set minimum. A time-out system is used to determine if there are objects within range. Each pair of sonar transducers is scanned at 10Hz, and has a range of 30mm to 1m with a resolution of less than 5mm.

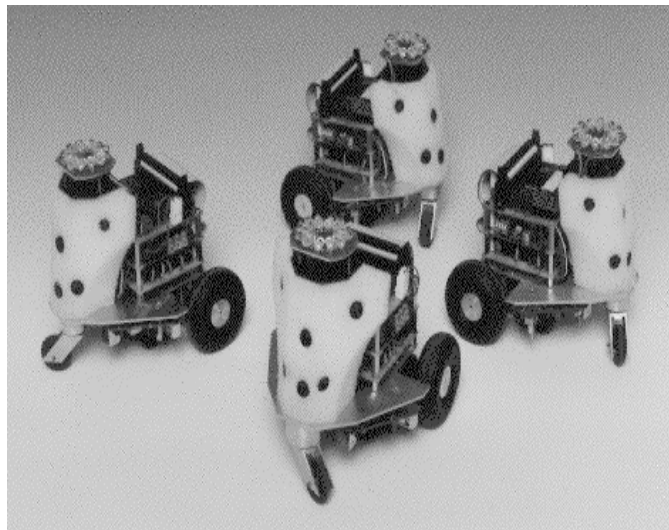


Figure 1 Mobile robots built at Reading

The robot is physically small, having a width of 140mm, a length of 130mm and a height of 140mm. Each robot weighs less than 600 grams. Two back wheels are differentially driven by two small D.C. motors with in-line gearboxes. The front of each robot is supported by a single castor wheel. At present the motor control is open loop with pulse width modulation, providing several different speeds (up to 1m/s) and directional control.

Each robot is controlled by an 8MHz Z80 CPU shown in figure 2. Its communication subsystem has 12 LEDs (transmitter) and 4 photodiodes (receiver). The Z80 receives an interrupt when the communication system receives a byte of data from other robots, which are detailed in later sections.

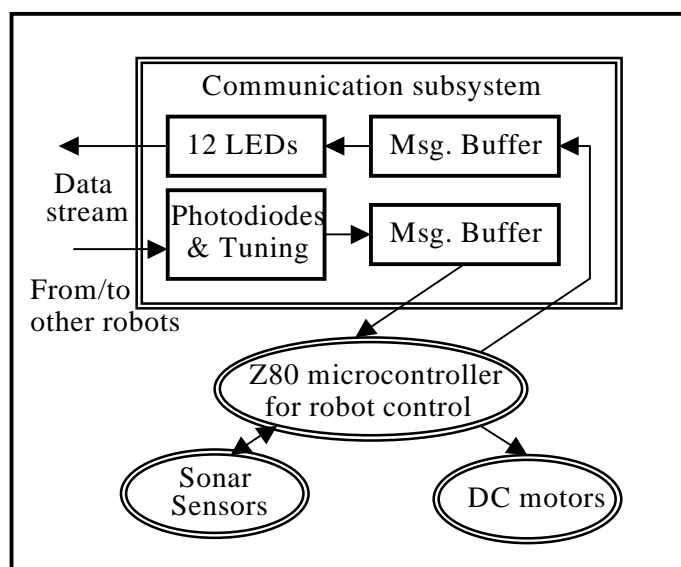


Figure 2 Block diagram of robot controller

### 3. Inter-robot communication

Communication behaviour is a key design issue for cooperation of multi robot systems [15]. The communication may take place directly via explicit communication facility such as radio link or indirect (pseudo communication method) through one robot sensing a change in other robots or its environment. It should be noted that the communication mechanism utilised is critical to successful implementation of coordination of multiple mobile robots. In general, audio or ultrasonic sound region media has rather low data rates. Radio systems has antenna size constrains with inter-robot localisation [9]. Therefore, an infrared communication (and localisation) system with multiple channels was developed at Reading for inter-robot communications of our mobile robots thus achieving high efficiency and reliability [9][10].

To transmit information a ring of twelve LEDs is used each with a half power angle of 60 degrees. They are arranged 30 degrees apart to ensure 360-degree coverage. Figure 3 shows their theoretical light intensity changes with respect to the viewing angle. Four photodiodes has been used in each mobile robot to receive information. They are arranged 90 degrees apart (each with a half power angle of 120 degrees), as shown in figure 4. This combination of LEDs and photodiodes allows communication regardless of the relative orientations of the robots.

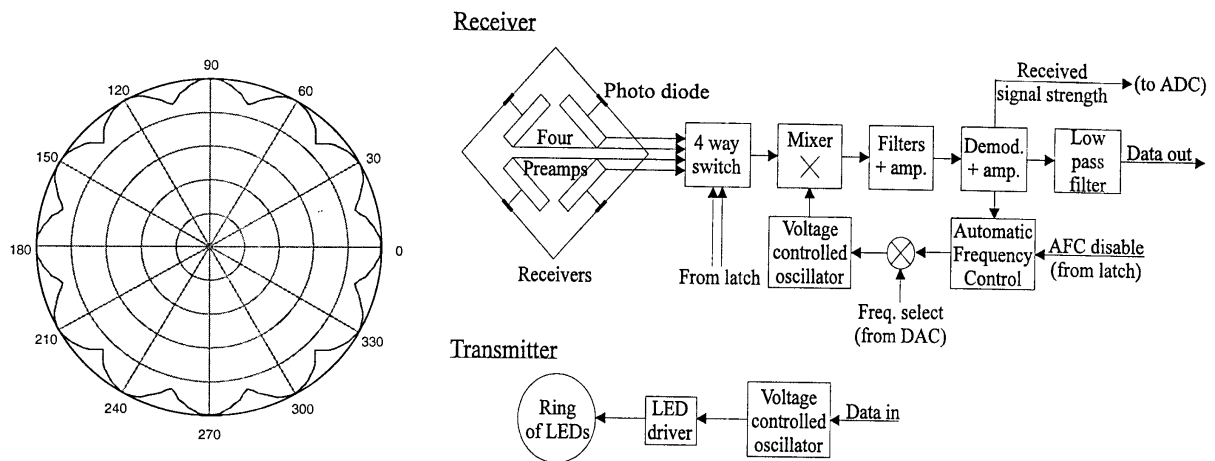


Figure 3 Polar response of 12 LEDs

Figure 4 Block diagram of the communication system

Figure 4 shows a block diagram of the communication system being developed at Reading [7]. Information is transmitted by frequency modulating the IR carriers, with the decoding being carried out using an off-the-shelf radio frequency (RF) integrated circuit. The information is present in the frequency changes around nominal frequency. The transmissions from other robots are received by one of four photodiodes and are then mixed with selected tuning frequency (heterodyning). A filter is used to obtain the information with the correct carrier frequency. The received information is demodulated and passes through a low pass filter to give digital data. Data is transceiver at 1200-baud (120 bytes per second) using differential phase shift keying (DPSK), thus permitting automatic frequency control. The range of this communications system is over 7 metres in the worst case.

The system is frequency division multiplexed, with each robot having its own channel in the range from 220kHz to 400kHz, as shown in figure 5 (a). The message for group learning consists of 16 bytes. As shown in figure 5 (b), the first three bytes, i.e. 255, 255 and 229, are message head used for synchronisation, then 11 bytes are message body, and the last two bytes are a check sum for error verification. A message is sent receptively until a new and updated one is ready. Although the encoded message is sent and received serially byte by byte, a complete message is retrieved only when all 14 bytes have been received (i.e. the 229 through to the second byte of checksum). It should

be notice that the message being sent is no guaranty to be received by its target robots. If the robot fails to retrieve a message, it will simply disregard it and look for a new one.

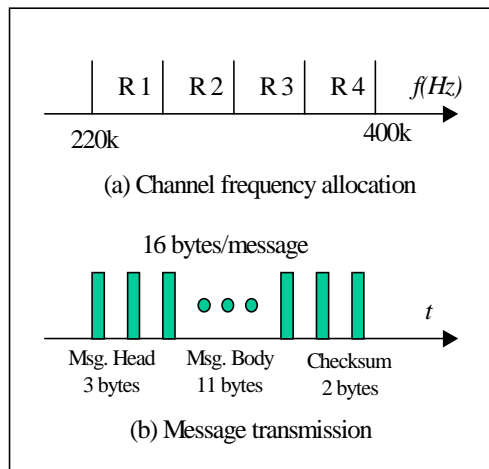


Figure 5 Frequency & message

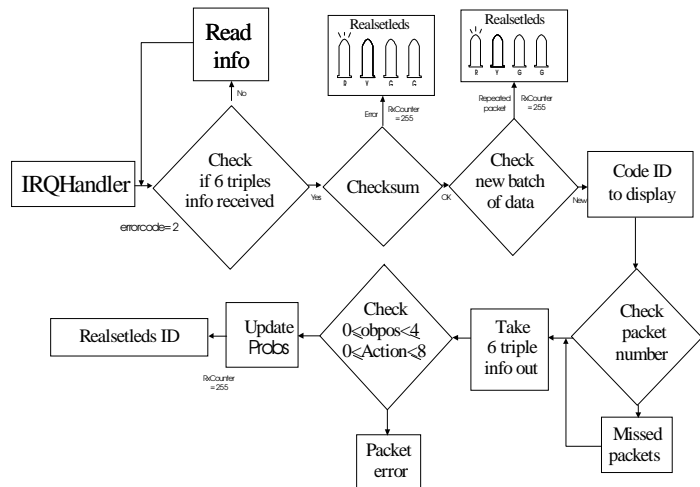


Figure 6 Communication interrupt subroutine

The retrieved message is handled by an interrupt subroutine that operates each time when a complete byte is received by the communication system. Figure 6 shows a flow chart of the interrupt subroutine. If the value 255 is received twice, which is to synchronise the serial port, and then 229 is received. The interrupt routine begins to receive the encoded information consisting of 6 triples. More details about data format can be seen in [9][14]. Note that the visual LEDs have been used to display whether there is an error or a missed packet of data, by switching Red LED on. The identity of the robot that the data was attempted to be received from or successfully received from is displayed on the remaining LEDs.

#### 4. Examples of co-ordination of multiple robots

In this section, two typical examples are given to demonstrate the proposed cooperative behaviours that have been implemented using a team of mobile robots at Reading. One example is to show flocking behaviour and another is group learning behaviour.

##### 4.1 Flocking behaviour

Flocking is a behaviour that is commonly found in nature. Animals sometimes flock for protection from predators. However, flocking presents an interesting problem in the field of robotics, i.e. the robots have to get close to each other but also have to avoid collisions. With many mobile robots in close proximity, their sensors will receive much more noise than when they are spread out. Therefore it poses a challenge which is non-trivial task that needs to be solved.

Since a global destination is not practical in a finite environment, some form of leadership is required during flocking. The selection of the leader has to be dynamic such that the flock should be able to split up to go around obstacles, and rejoin once past the obstacle. There is no predefined leader. If one selected leader fails, one of other robots is able to declare its leadership if it is in a good position. The rule for choosing a leader and the flocking operation of a team of mobile robots are as follows:

1. Avoid obstacle using safeguard behaviour with highest priority, based on sonar data.
2. Become a leader if it is in a good position and no other robots are visible, and wander around.
3. Try to maintain its position if it is in a flock.
4. Speed up and head towards a flock which can be seen in the distance.

5. Return to step 1 to recursively implements this sequence of actions.

The communication system is used by each robot to inform other robots whether it is a leader or a follower. Figure 7 shows a sequence of image frames taken from a video recording of the flocking robots.

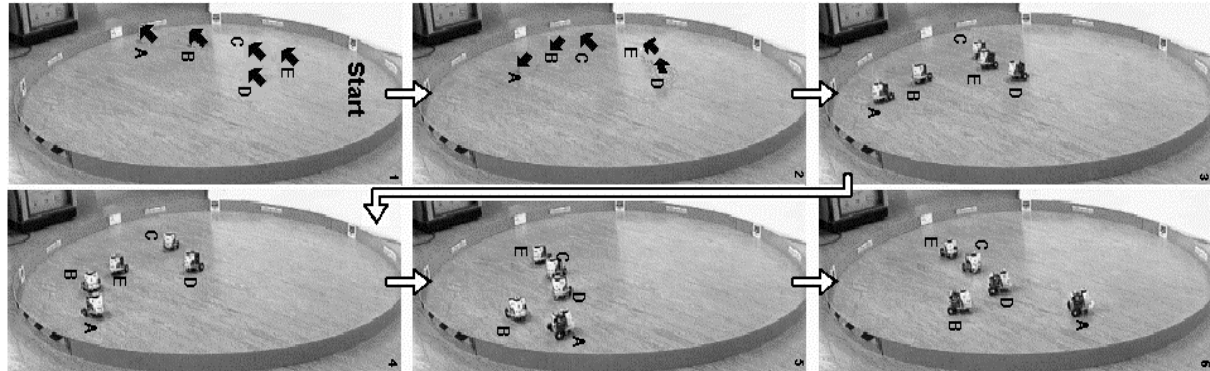


Figure 7 A sequence of 6 frames taken from flocking operation

## 4.2 Shared experience learning

Following the development of co-ordinating mobile robots [7], shared experience learning for multiple mobile robots was investigated during last few years [8][9][10]. In this instance we are investigating the effect of each robot sharing its experiences with the other robots. This should decrease the time taken for the robots to learn how to avoid obstacles as each robot has more data from which to learn.

### The learning algorithm

The learning algorithm is based on sets of fuzzy automata [11]. If the motor speeds are limited to full speed forward, full speed backwards and stop then with two motors there are nine different possible output actions ( $a1...a9$ ). With three channels of sonar each giving an 8-bit range value there are 224 input states. We have mapped these to five input states that represent the different circumstances the robot can find itself in:

- State 1:* no object near robot
- State 2:* obstacle in distance to the right
- State 3:* obstacle in distance to the left
- State 4:* obstacle relatively near the right
- State 5:* obstacle relatively near the left

The robot is required to learn a mapping from input states to output actions which allows it to wander around avoiding obstacles. Each of the five input states ( $s1...s5$ ) has its own fuzzy automaton associated with it. Each automaton, which is effectively a set of motor actions ( $a1...a9$ ), has a set of probabilities of taking the associated action ( $p1...p9$ ). A weighted roulette wheel technique is used to randomly select the most appropriate output action for the given input state. The action with the highest probability is the most likely to be chosen. This method is similar to techniques that are

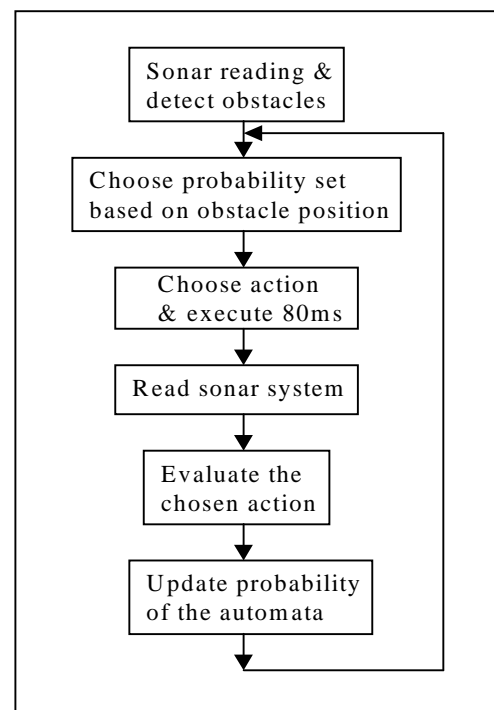


Figure 8 Learning algorithm

used in genetic algorithms [5]. The chosen action is executed for a short period of time and is then evaluated. If the action was successful, its probability of being selected is increased whilst if the action was unsuccessful, its probability of being selected is decreased. In both cases the other probabilities of the chosen automaton are adjusted in order to keep the total probability constant. In the following description the performance value is *Alpha* (positive meaning successful and negative meaning unsuccessful), *n* is the action that was chosen and *j* represents all the actions except for *n*. The rules for adjusting the probabilities is thus:

$$pn = pn + Alpha \text{ (increase/decrease probability of chosen action)}$$

$$pj = pj - (Alpha / 8) \text{ (decrease/increase other probabilities)}$$

Since we are only using a slow processor all of the probabilities are stored as signed 16-bit integers rather than as fractions.

In order to evaluate *Alpha*, a definition of which actions are good and which actions are bad is required. Rules were chosen which are general and hence would not give the robot any information about which motor actions to select. The basic rules are that if there is no object within range it is good to go forward but if an object is relatively near it is good to get further away from it. Thus if the object is in the distance it is still good to go forwards but it is also good to get further away from the object.

The rules used for calculating *Alpha* are:

State 1: (no object near robot)

$$Alpha = (LeftSpeed + RightSpeed) * 12$$

State 2 and state 3: (obstacle in distance)

$$Alpha = ((RightSpeed + LeftSpeed) * 6 + (RightDir + LeftDir) * 2)$$

State 4 and state 5: (obstacle relatively near)

$$Alpha = FrontDir * 4$$

where: LeftSpeed and RightSpeed are the current motor speeds; +1 for forwards, -1 for backwards and 0 for stop; and RightDir, LeftDir and FrontDir are the changes in distance of the nearest object to the relevant sensor. This is positive when the robot is getting further away and negative otherwise. The values are bounded to +/- 9 (approximately +/- 35mm) to prevent a huge value from being returned when an object is suddenly lost or detected by sonars.

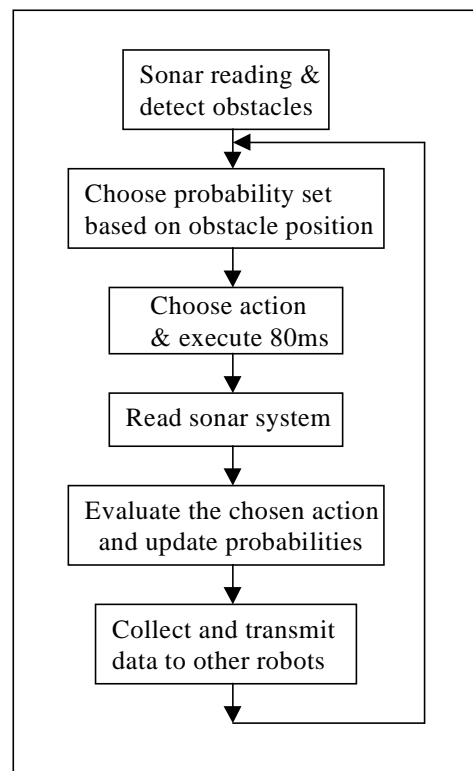


Figure 9 Group learning algorithm

The main adaptive loop of the program is shown in figure 8. The range returned from the front sonar sensor is used for determining whether the robot has successfully got further away from the obstacle when it is close. At this range the front sensor can always see an object anywhere towards the front of the robot. The left and right sensors are used when the object is further away as the front sensor is unable to detect objects to the far left and right.

### Group Learning

For group learning, each robot has to transmit for every learning cycle, its input state (i.e. 1 to 5), the output action that it chose (i.e. 1 to 9) and how good or bad that action was (i.e. the *Alpha* factor). To allow for the low rate of data transmission and the scanning of up to four robots, six sets of this data

are compressed into a packet of data along with start of packet/synchronisation codes, an incremental packet ID and a two-byte check sum. The ID code allows for the detection of missed packets since if the ID of this packet is not one greater than the last packets ID then one or more packets of data have been missed. This loss of packets is typically about 5% for two way learning and 15% for four way learning. The main adaptive loop of the program is present in figure 9.

## Results

The test set-up consisted of four robots each in its own environment bounded by an elliptical wall of 7.5m circumference and 120mm height. Tests were conducted with four robots running the learning algorithm at the same time, both with and without the sharing of experiences. The four robots could thus be acting as four individuals, two pairs or one quadruplet. The 45 probabilities ( $p1...p9$ ) for each ( $s1...s5$ ) were stored in the on board EEPROM every 3.4 seconds. This allowed 10 minutes of data to be stored in the memory space available. Three tests, each consisting of four robots, were performed under each test condition, thus giving twelve sets of data to allow an average learning rate to be determined.

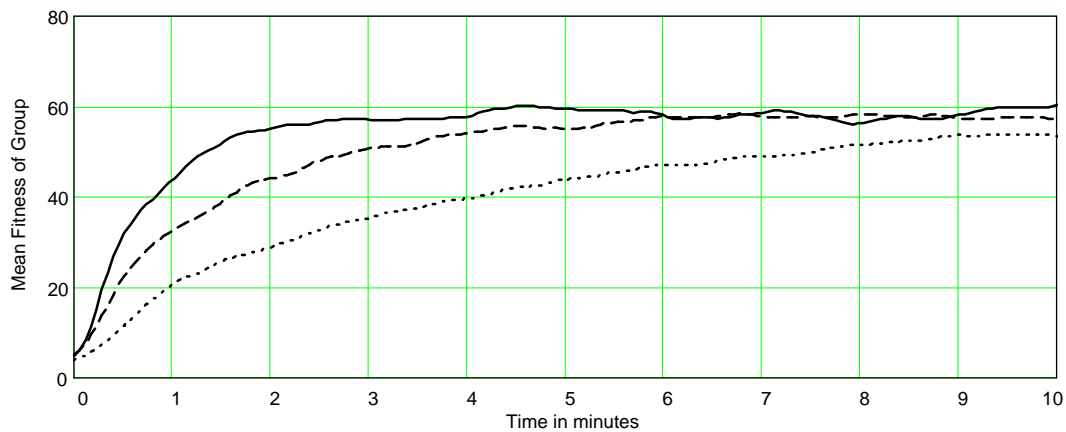


Figure 10. Graph showing the mean fitness of the robots vs. time.

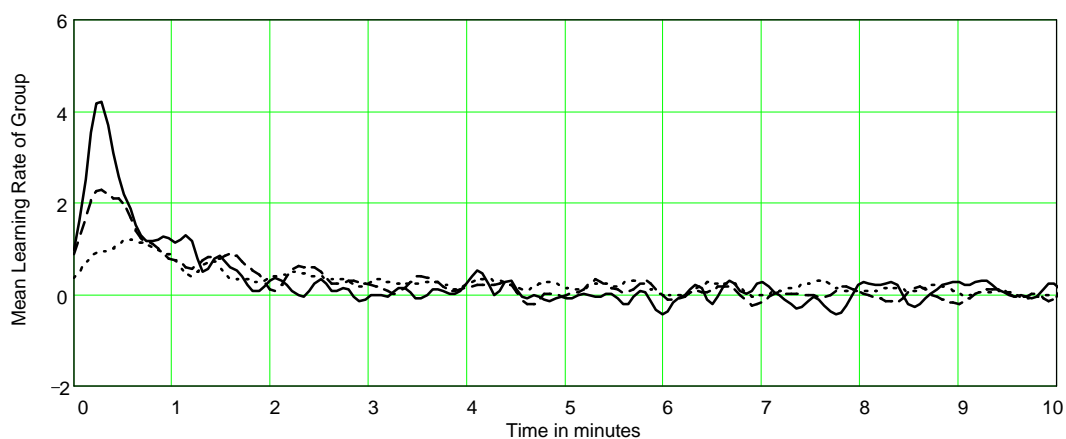


Figure 11 Graph showing mean learning rate of group vs. time

In order for the fitness of the robot to be determined we considered the result of each of the actions ( $a1...a9$ ) for each state ( $s1...s5$ ). We were thus able to determine the approximate *Alpha* factor that each action would give in each state. It was immediately evident that five states were too few even for an initially static robot. For example if the robot were close to a wall on the left and facing along the wall, the action spin right would achieve a high *Alpha*. If, however, the robot were facing the wall almost head on, the action spin right would achieve a much lower *Alpha*. Both of these situations would be considered by the robot to be state 5 (obstacle relatively near the left). The situation

becomes worse if the dynamics of the robot are taken into account. The momentum of the robot may continue to bring it closer to an obstacle even though it has chosen the best action for avoiding the obstacle in that situation. This could result in a negative *Alpha* and hence a reduction in the probability of choosing the best action for that situation in the future. In fact the robot will still learn the best action, as poorer actions would result in an even larger negative value of *Alpha*. The learning will, however, be slowed down. The fitness was calculated by weighting each of the values of probabilities to reflect the expected *Alpha* and then summing all the weighted probabilities. A graph of the mean fitness of the set of twelve tests vs. time for the three cases is shown in figure 10.

The dotted line shows individual learning, the dashed line shows pairs sharing experiences and the solid line shows quadruplets sharing experiences. Although it is subjectively obvious that the four-way learning is faster than the two-way, which is in turn faster than the individual, two methods were used to quantify the learning rates. The first was to numerically differentiate the fitness data with respect to time. This should show the mean learning rate across the group with respect to time. This is shown in figure 11.

The peak values taken from this graph are 1.18 for the individual learning, 2.28 for the two way learning and 4.25 for the four way. As the fitness data had to be smoothed slightly in order for the numerical differentiation to converge and the resultant graph was noisy we decided to try fitting exponentials of the form  $p(1-e^{-t/T})$  to the fitness graphs. A least square method was used to determine the values of  $p$  and  $T$  which gave the best fit. The original fitnesses and the best-fit exponentials are shown in figure 12.

The exponentials are a reasonable fit and gave time constants of 2.675 minutes for individual learning, 1.29 minutes for two way learning and 0.687 minutes for four way learning. These time constants were converted to rates and normalised for comparison with the data from figure 11.

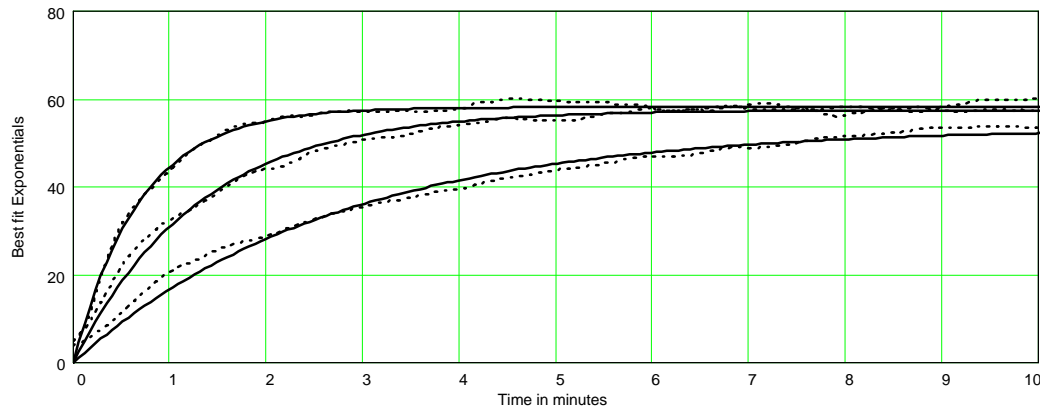


Figure 12 Graph showing best-fit exponentials and original fitness curves

The results clearly show that shared experience learning is faster and more repeatable than individual learning. If the communication system were perfect then the multiple learning robots would effectively be one agent learning with two or four times the learning data rate and the possibility of being in two or four situations at the same time. Each of the robots should therefore just be considered to be part of a single learning agent and the increase in learning rate would thus be as expected. In fact errors in the communication system mean that in practice the robots do differ and each receives slightly less than the expected learning data rate and hence learns slightly more slowly than expected

## 5. Conclusions

The advantages of cooperative mobile robots over single complex mobile robot include the potential for increased fault tolerance, simpler robot design, and wide application domain. However, the use of

multiple robots introduces additional control issues that are not present in single robot solutions. This paper presents a feasible solution for a team of autonomous mobile robots to function in a cooperative manner. To realise coordination, a multi-channel infrared communication system has been developed to exchange message among mobile robots. Two examples of flocking and shared experience learning have been given to demonstrate the system performance. The results clearly show that shared experience learning is faster and more repeatable than individual learning. If the communication system were perfect then the multiple learning robots would effectively be one agent learning with two or four times the learning data rate and the possibility of being in two or four situations at the same time.

The next stage of our research is to investigate other cooperative behaviours towards real world applications. The works will encompass both the developments of sensors and actuators, and the design of intelligent algorithms for cooperative behaviours. More specifically, it will address:

- How adaptation and learning algorithms should be improved to make robotic systems more flexible in a dynamic environment?
- What sensors and data fusion algorithms are required to capture environment features effectively?
- How should the navigation system deal with uncertainties in sensors and actuators?
- Which method the robots can determine for themselves, by communication, an appropriate number of input states?

## References

1. R. Alami, S. Fleury, M. Herrb, F. Ingrand, F. Robert, Multi-robot cooperation in the MARTHA project, IEEE Robotics & Automation Magazine, pages 36-47, March 1998.
2. R.C. Arkin, Integrating behavioural, perceptual and world knowledge in reactive navigation, Journal of Robotics and Autonomous Systems 6, 1990.
3. H. Asama, Operation of cooperative multiple robots using communication in a decentralised robotic system, Proc. Conf. on From Perception to Action, Edited by P. Gaussier and J.D. Nicoud, Lausanne, Switzerland, 7-9 Sept. 1994.
4. R.A. Brooks, A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, pages 14-23, 1986.
5. M. Brady and H. Hu, The Mind of a Robot, Philosophical Transactions: Physical Sciences and Engineering, the Royal Society, Vol. 349, No. 1691, pages 15-28, London, U.K., October, 1994
6. D.E. Goldberg, Genetic algorithms, Addison Wesley Press, 1989.
7. Y. Ishida, *et. al.*, Functional complement by co-operation of multiple autonomous robots, Proc. IEEE Int. Conf. On Robotics and Automation, pages 2476-2481, 1994
8. D.A. Keating and I.D. Kelly, Increasing mobile robot learning rates through sharing of experiences, Proc. IEE Control'98 conference, Swansea, UK, September 1998.
9. I.D. Kelly, The development of shared experience learning in a group of mobile robots, PhD thesis, University of Reading, 1997
10. I.D. Kelly and D.A. Keating, Faster learning of control parameters through sharing experiences of autonomous mobile robots, Int. Journal of System Science, Vol. 29, No. 7, pages 783-793, 1998.
11. R.J. Mitchel, D.A. Keating, C. Kambhampati, Learning strategy for a simple robot insect, Proc. IEE Control'94 Conference, pages 492-497, 21-24 March 1994.
12. M.J. Mataric, Learning in Multi-Robot Systems, In Lecture Notes In Artificial Intelligence (LNAI), Adaptation and Learning in Multi-Agent Systems, Gerhard Weiss and Sandip Sen, eds., Vol. 1042, Springer-Verlag, 1996.
13. K. Narendra and M. Thathachar, "Learning Automata: an introduction." Prentice-Hall, 1989.
14. D. Vinagre, Shared experiences learning in a group of mobile robot, Dept. of Cybernetics, University of Reading, September 1998.
15. J. Wang, On signboard inter-robot communication in distributed robotic systems, Proc. IEEE Int. Conference on Robotics and Automation, pages 1045-1050, May 1994.