In ICAR 97

AN INTERACTIVE METHOD FOR CONTROLLING GROUP SIZE IN MULTIPLE MOBILE ROBOT SYSTEMS

Owen Holland and Chris Melhuish
Intelligent Autonomous Systems Engineering Laboratory, Faculty of Engineering,
University of the West of England, Coldharbour Lane, Bristol BS16 1QY, UK
o-hollan/cr-melhu@uwe.ac.uk

ABSTRACT

This paper proposes a method of controlling group size in multi-robot systems with very limited computational abilities, such as micro- or nano-robots.The method is loosely modelled on the signalling behaviour of species such as crickets, frogs, and fireflies, where males attract females by making periodic calls or light flashes. A characteristic of such behaviour is that groups of animals broadcast their signals in synchrony, forming a chorus. In this paper, synchrony is used in conjunction with random deviations from synchrony in order to enable each individual to estimate the size of the group over a period of time; since an individual will approach a group of below the required size, and leave a group which is above that size, group size should remain around the target value. Results from simulations are presented which show that the mechanism is feasible; the development of a robotic system to use this method is described.

MOBILE ROBOTS: MULTIROBOT SYSTEMS: AGGREGATION; BIOLOGICAL INSPIRATION

1 INTRODUCTION

Multirobot systems can often be divided into two broad categories: those using symbolic representations and explicit communication; and those using behaviour based techniques and implicit communication 'through the world'. For those developing the second type of system, there is usually some attraction towards minimalism, and a tendency to explore the use of behavioural strategies known to be used by biological systems such as social insects. The system described in this paper is both minimalist, and inspired by the collective behaviour of biological systems. Before describing the system, it will be convenient to discuss the questions begged by this approach: why should anyone build behaviour based multirobot systems other than out of curiosity, why should minimalism and biological inspiration confer any benefit, and what practical purposes might such systems be useful for?

1.1 Behaviour based multirobot systems

The first question has a power in 1997 that it did not have twelve years ago, when the current wave of behaviour based work began. (The first behaviour based robots were built in 1948 by Grey Walter, but his articulation of the underlying philosophy came to light only recently (Holland [1996a]). In the mid 80s, it was simply not possible to give a mobile robot very much in the way of computational power, or communications; behaviour based robots impressed mainly because they were able to produce good performance within the limitations imposed by the available technology. However, the development and availability of cheap, high performance, low power computational and communications equipment has made it anachronistic to continue to work with 68HC11s and low-speed radio; modern small robots can just as easily use a miniature Pentium-based PC (PC/104) and a radio LAN giving 1.6Mb/s (e.g. Dawkins et al [1997]). If multirobot systems are to use behaviour based techniques, then it should be because of the characteristics of the behaviour of such systems, rather than their ability to run on obsolete hardware.

The main reason for building behaviour based multirobot systems is in fact to discover the characteristics of such systems; we know this is worth doing because the biological equivalents, social insect systems, appear to have abilities which transcend the limitations of the individual agents, and also have characteristics of flexibility and robustness which are extremely desirable in many contexts.

Why should minimalism and biological inspiration be thought to confer any advantage? Minimalism is often though to be attractive from the point of view of engineering. For instance, the less there is, the less there is to go wrong; this applies to hardware, where reliability (excluding considerations of redundancy) may be achieved by reducing the parts count, and also to software, where the probability of a coding error increases with the length of the programme. However, unless we are to resume where Grey Walter left off, and build robot controllers