Strategies for maintaining large robot communities

Stephen English[1], Jeffrey Gough[1], Alexis Johnson[1], Robert Spanton[1], Joanna Sun[1], Richard Crowder[1] and Klaus-Peter Zauner[2]

[1]School of Electronics and Computer Science, University of Southampton, United Kingdom
[2]Science and Engineering of Natural Systems group, School of Electronics and Computer Science, University of Southampton
kpz@ecs.soton.ac.uk

The confluence of progress in micro-actuators, power sources, and mixed-signal microelectronics have recently moved swarm robotics and robot communities from simulation to reality. Swarms of 20 to 100 robots are in use already, implementations with several hundred robots are practicable, and communities exceeding a thousand robots are certainly conceivable. Such large robotic communities provide platforms for numerous exciting research directions including collaborative swarms and self-reconfiguring structures.

Maintaining hundreds of robots, however, poses significant practical challenges. The literature on strategies for maintaining software and hardware in large robot communities is sparse, even if applicable concepts from wireless sensor-networks are included. Crucial for the viability of any such strategy is its impact on cost per robot.

To provide a realistic setting we introduce a robot platform designed to be fabricated in full on standard printed circuit board (PCB) assembly lines. In this context we introduce a framework for on-line testing and calibration based on code pieces, termed plasmids, that migrate among the micro-controllers of the robots. The proposed approach allows the robots access to a larger library of code then what could be stored locally.

The robot consists of a single PCB that doubles as chassis and contains no custom mechanical components. Inexpensive motors (mass produced to vibrate mobile phones) are directly soldered to the circuit board and used in direct drive. Our prototypes use a 200 mAh rechargeable lithium polymer battery giving the robot over an hour of autonomy while moving at a speedy 1 m/s. An MSP430F2131 microcontroller controls the robot and communicates with neighbouring robots via infrared light. The simplicity of the design allows the entire robot to be assembled with low-cost PCB manufacturing techniques and is well suited for small-scale mass production of several hundred robots.

While this design significantly reduces the current cost barrier to obtaining a robot swarm, it also shifts the attention to the practical problem of maintaining hundreds of robots. Recharging batteries, sieving out robots with worn tyres or accidental damage is one aspect. A second aspect is testing and calibration. It can not be performed in the PCB assembly process and cost considerations prevent proprioceptive sensor. Collaboration among robots to verify performance and provide feedback (e.g., drift direction during a run and return) provide a scalable alternative. A third aspect is the maintenance of software in the robot community.

Our plasmid framework addresses all three aspects with a design that is lightweight enough to run on the microcontrollers. Pieces of code and associated attributes (version, target number for redistribution, lifetime, conditions for transmission) are propagated among robots that meet. For example, the code may perform a test on the robot and require to be forwarded to four other robots that have not encountered this test, before it is deleted. Such test plasmids traverse the robot community which, in its collective memory,contains and executes more code than would fit within the program memory of a single device.