Synchronization and Fault Detection in Autonomous Robots

Anders Lyhne Christensen, Rehan O'Grady and Marco Dorigo

***Abstract*—In this study, we show a group of robots can synchronize based on firefly-inspired flashing behavior and how dead robots can be detected by other robots. The algorithm is completely distributed. Each robot flashes by lighting up its on-board LEDs and neighboring robots are driven to flash in synchrony. Since robots that are suffering catastrophic failures do not flash periodically, they can be detected by operational robots. On a real multi-robot system of 10 autonomous robots, we show how the group can correctly detect multiple faults, and that when given (simulated) repair capabilities, the group can survive a relatively high rate of failure.**

I. INTRODUCTION

In this study, we leverage some of the high-level principles behind synchronizing systems found in Nature to obtain a robust, simple, distributed approach to fault detection in groups or swarms of autonomous robots. By detecting faults, the robots can leverage their multiplicity and ensure continued operation by reassigning functional robots to the failed robots' task or by taking steps to have the failed robots repaired. Some faults are hard to detect in the robot in which they occur. These faults include software bugs that cause a control program to hang, sensor failures that prevent a robot from detecting that something is wrong, and mechanical faults such as an unstable connection to a power source. Alternatively, a robot might be able to detect a fault, but the fault itself might still render the robot unable to alert other robots or a human operator. The robustness of a multi-robot system can therefore be improved by giving robots the ability to detect faults in one another.

In the accompanying video, we demonstrate a completely distributed approach that builds on the principles behind synchronization observed in fireflies to implement a heartbeat-like fault detection scheme in a group of autonomous mobile robots. For our experiments we use robots from the *swarm-bot* robotic platform [1].

II. SYNCHRONIZATION

Many distributed natural systems can be reasonably modelled as networks of pulse-coupled oscillators. The internal state or *activation* of each oscillator increases over time until it reaches a certain threshold. When the threshold is reached, the oscillator discharges (*fires*) and the activation instantly jumps back to zero – the cycle then repeats. When a nearby oscillator observes a flash it immediately increases its activation by a (small) amount. If this increase causes the oscillator's activation to exceed the firing threshold, the oscillator fires, resets its activation to zero, and commences a new cycle. An example with two oscillators is shown in Fig. 1.

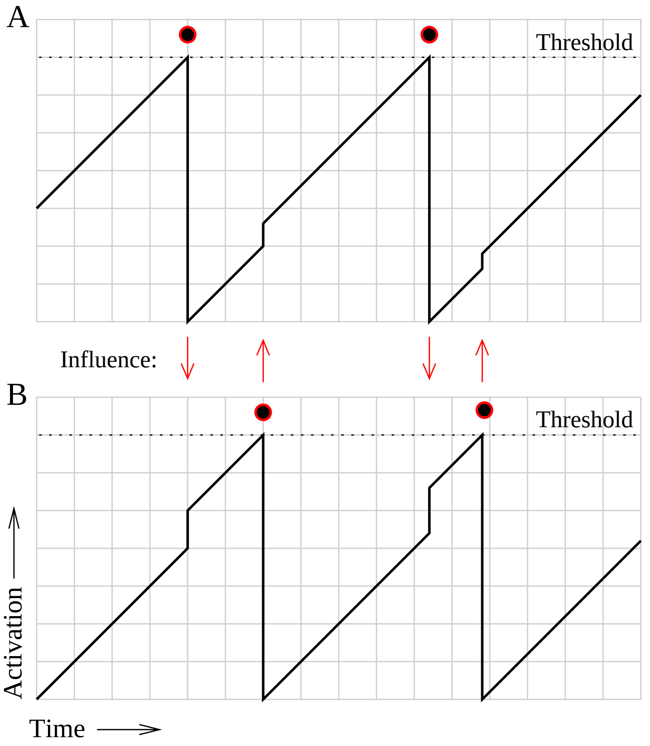

Fig. 1. An example of two pulse-coupled oscillators. Both oscillators increase at a constant rate until the threshold is reached or until one oscillator observes that the other one fires. When an oscillator's activation reaches the threshold, the oscillator fires. If one oscillator observes the other's firing, it increases its own state by $\epsilon\, x$, where ϵ is the pulse-coupling constant and x the activation of the oscillator.

The self-synchronization of pulse-coupled oscillating cardiac pacemaker cells was first described by Peskin [2]. Mirollo and Strogatz later showed that a population of fully connected pulse-coupled oscillators almost always evolves to a state in which all oscillators are firing synchronously [3]. Recently, Lucarelli and Wang [4] showed that a group of pulse-coupled oscillators will eventually synchronize even when each oscillator interacts with only a subset of the population. This holds true for systems with changing topologies as long as the interaction graphs are connected.[1]

Anders Lyhne Christensen is with DCTI-ISCTE, Av. das Forças Armadas, 1649-026 Lisbon, Portugal and IRIDIA, CoDE, Université Libre de Bruxelles, 50 Av. Franklin Roosevelt CP 194/6, 1050 Brussels, Belgium (e-mail: anders.christensen@iscte.pt)

Rehan O'Grady and Marco Dorigo are with IRIDIA, CoDE, Université Libre de Bruxelles, 50 Av. Franklin Roosevelt CP 194/6, 1050 Brussels, Belgium (e-mail: rogrady@ulb.ac.be, mdorigo@ulb.ac.be)

[1] We obtain the interaction graph for a population of oscillators by letting every oscillator correspond to a node in the graph with an edge to each member of its neighbor set.

We propose an approach for synchronization based on local visual communication. The approach resembles behavior observed in fireflies: we let each robot act as an integrate-and-fire oscillator and when the activation of the oscillator reaches a certain threshold, the robot lights up its on-board LEDs in red and resets its oscillator. When neighboring robots (within 50 cm) detect the flash, they increment their own activation. Eventually all the robots are driven to flash in synchrony.

III. FAULT DETECTION

Synchronization can be used as a fault detection tool if the robots assume that a robot that is not flashing has a fault. A robot can stop flashing voluntarily if it detects a fault in itself. In this way, it can implicitly signal that it requires assistance. A robot also stops flashing when it experiences a catastrophic fault (software bug, physical damage, and so on...) which causes the control program and thus the periodic flashing to stop. When operational robots discover a non-flashing teammate they know that a fault has occurred and they can take steps to rectify the situation.

In a normal situation the robots would be operational and synchronized. However, when robots commence a task or when they encounter each other after having been separated for a period of time, their activations are likely to differ. In other words, they are not synchronized. This means that one robot cannot assume that another robot has become non-operational just because the two robots do not flash in unison. To address this issue, a flashing robot does not immediately consider another robot non-operational if the two robots do not flash at the same time. Instead, the flashing robot (F) treats the robot (N) that did not flash when F flashed as a *candidate* robot. We say that F becomes *suspicious* of N. If N flashes before F flashes again, both robots are operational but they are just not (yet) synchronized. However, if F flashes *again before* N flashes, F assumes that N is non-operational. Hence, a robot detects a fault if it flashes twice while observing that another robot does not flash at all.

IV. EXPERIMENTS ON REAL ROBOTS

In order to test our approach in a scenario where more than one robot can become non-operational, we conducted an experiment with a group of 10 robots, in which a fault was injected in an operational robot with a probability of $p = 0.0005$ every control cycle. We simulated a repair mechanism that allowed one robot to "repair" another robot by physically connecting to it and by illuminating its blue LEDs. When a failed robot detected that it had been "repaired", it set its activation to a random value and restarted its controller. We let the experiment run for 12 min. All robots were operational from the start of the experiment and the first fault occurred after 20 s. During the experiment a total of 13 simulated faults occurred. At one point a total of four robots were non-operational, while only one robot was non-operational when the experiment was stopped.

The results suggest that our approach is robust in situations where multiple faults can be present at the same time. Furthermore, when the robots can repair one another, a swarm of robots can survive a relatively high rate of failure.

V. SUMMARY

In this study, we have present a distributed approach for detecting non-operational members in swarms of robots. Our algorithm is inspired by the synchronous flashing behavior observed in some species of fireflies. Robots flash periodically by lighting up their on-board LEDs. Whenever a robot perceives a flash from a nearby robot, it increases its own activation and flashes slightly sooner than if it had not seen a flash. We show that swarms of simulated and real robots following this scheme are driven to flash in synchrony. In our fault detection scheme, the periodic flashes function as a heart-beat mechanism. A failed robot need not explicitly signal other nearby robots that it requires assistance – it only needs to stop flashing. We do not, therefore, need to distinguish between robots that voluntarily have stopped flashing and robots that, for instance, have experienced a catastrophic fault rendering them unable to take any action – including flashing. We showed that real robots are able to detect and respond to faults by detecting non-flashing robots. We also showed that the scheme is robust to multiple faults and that a team of robots with self-repair capabilities is able to survive a relatively high rate of failure. For more detail on the approach and more results see [5].

VI. ACKNOWLEDGMENTS

This work would not have been possible without the innovative robotic hardware developed by Francesco Mondada's group at the Laboratoire de Système Robotiques (LSRO) of EPFL. This work was supported by the *SWARMANOID* project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-022888 and by the *VIRTUAL SWARMANOID* project funded by the F.R.S.-FNRS. The European Commission is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo acknowledges support from the F.R.S.-FNRS, of which he is a Research Director.

REFERENCES

[1] F. Mondada, G. C. Pettinaro,A. Guignard, I. V. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella and M. Dorigo, SWARM-BOT: A new distributed robotic concept, *Autonomous Robots* 17(2–3); pp 193–221, 2004

[2] C. S. Peskin, Mathematical aspects of heart physiology, Courant Institute of Mathematical Sciences, New York University, NY; 1975, pp 268-278

[3] R. E. Mirollo and S. H. Strogatz, Synchronization of pulse-coupled biological oscillators, *SIAM Journal on Applied Mathematics*, 50(6); pp 1645-1662, 1990

[4] D. Lucarelli and I. J. Wang, Decentralized synchronization protocols with nearest neighbor communication, *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM Press, New York, NY; pp 62-68, 2004

[5] A. L. Christensen, R. O'Grady and M. Dorigo, From fireflies to fault tolerant swarms of robots, *Technical report IRIDIA/2008-02 IRIDIA, CoDE - Université Libre de Bruxelles*, 2008, *Submitted to IEEE Transactions on Evolutionary Computation.*