A Mechatronic System Design Case Study: Control of a Robotic Swarm Using Networked Control Algorithms

Peymon Gazi, Mo Jamshidi
Dept. of Electrical Engineering, College of Engineering
University of Texas at San Antonio
San Antonio, USA

Aleksandar Jevtić, Diego Andina
Group for Automation in Signals and Communications
Technical University of Madrid
Madrid, Spain

Abstract— **This paper describes the use of networked control algorithms in designing a robotic swarm. The main goal of a robotic swarm is to divide one task into multiple simpler tasks. Have we designed a swarm this way, the main challenge would be the problem of delay in communication between individual robots. This paper also goes through the Swarm Intelligence concept and proposes the Network Formation Control algorithms to control a group of robots.**

Keywords- swarm robotics; swarm intelligence; networked control systems

I. Introduction

Swarm Robotics is a new approach to the coordination of multi-robot systems which consist of large numbers of mostly simple physical robots. It is supposed that a desired collective behavior emerges from the interactions between the robots and interactions of robots with the environment. This approach emerged on the field of Computational Swarm Intelligence, as a result of biological studies of the living colonies in nature, where swarm behavior occurs.

Swarm Robotics is often used in contrast with service robotics. Service robotics, as it is widely performed today, usually assumes a given service to be carried out by a single robot or, at most, by a small group of them working together. In any case, though, the concept of cooperation is intended more in the sense of a relay race, than in the sense of an actual team effort for achieving each single task. Each robot, in other words, is assumed to be able to cope with the basic problems of autonomy alone, i.e. locating itself, navigating within its environment, and in case also planning its own future actions.
A new and totally different way is the so-called Swarm Robotics that, as opposed to the more traditional approach, does not necessarily assume each robot as a stand-alone independent unit. On the contrary, Swarm Robotics assumes that a given mission is the result of a joint action of a swarm of simple units. Such units, in theory, might even be unable to perform the bare locomotion without the aid of others of their kind. This approach finds its theoretical roots in recent studies on swarm intelligence, i.e., in studies of self assembling and self-organizing capabilities shown by animals such as social insects [1]. Figure 1 depicts the main concept of swarm robotics.

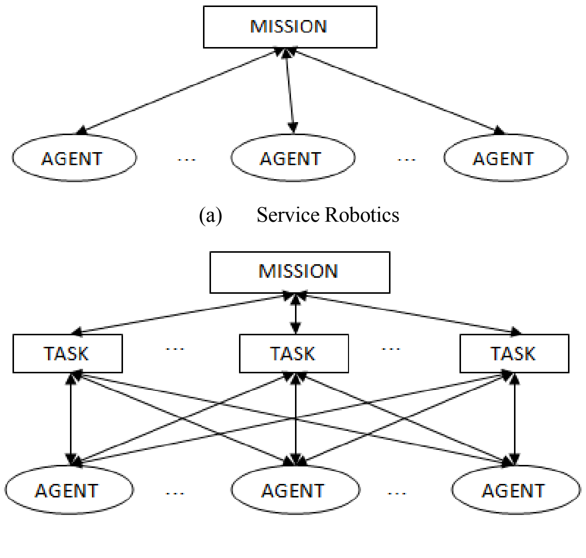


Figure 1. . Service Robotics vs. Swarm Robotics

The key to understanding a swarm is to always remember that each and every member of a swarm is in charge of a small portion of the whole mission. If we let the autonomous robot (member of a swarm) be modeled at the kinematic level as a unicycle

$$\begin{aligned}\dot{x} &= v\cos\varphi \\ \dot{y} &= v\sin\varphi \\ \dot{\phi} &= \omega\end{aligned} \qquad (1)$$

where *(x,y)* denotes the position of the robot, and φ denotes its orientation, a behavior is characterized by the way sensory data is mapped to the control input ω and v, corresponding to angular and translational velocities, respectively. Now relative to this robot model, a way of specifying the effect of individual behaviors is to let the behavior define a vector

$$B = r_B \begin{pmatrix} \cos(\varphi_B) \\ \sin(\varphi_B) \end{pmatrix} \qquad (2)$$

where r_B is the magnitude of the behavior vector and φ_B is its orientation. This vector formalism allows us to map behavior vectors to control values using some appropriate map $F(B) = (v, \omega)^T$.

Formulating multiple behaviors gives each member of the swarm to combine different behaviors to achieve a more complicated behavior [2]. For example one can break down the obstacle avoidance behavior, B_{OA}, to small vector summations, $B_{OA,1\ldots k}$.

$$B_{OA} = B_{OA,1} + B_{OA,1} + \ldots + B_{OA,k} \qquad (3)$$

where the behavior vectors are given by

$$r_{B_{OA,j}} = \begin{Bmatrix} 0 & \text{if } d_j > D \\ C_{OA} \dfrac{(D - d_j)}{(d_j^{\,3})} & \text{otherwise} \end{Bmatrix} \qquad (4)$$
$$\varphi_{B_{OA,j}} = \pi + \varphi_j$$

where $C_{OA} > 0$ and D is the safety distance at which the obstacle-avoidance behavior starts affecting the system. Combining such a behavior with an "approach target behavior" like (6),

$$r_{B_{OA,j}} = C_{AT}$$
$$\varphi_{B_{OA,j}} = \arctan\left(\frac{(y_g - y)}{(x_g - x)}\right) \qquad (5)$$

where $C_{AT} > 0$ is the constant magnitude, and the goal is located at (xg,yg), results in the swarm moving toward target while each member of the swarm tries to avoid the possible obstacles.

II. Overview of the Swarm Intelligence Algorithms

Swarm Intelligence based systems typically consist of a population of simple agents interacting locally with one another and with their environment [3]. The benefits of cooperation can be significant in situations where global knowledge of the environment does not exist. Individuals within the group interact by exchanging locally available information such that the global objective is achieved more efficiently than it would be if performed by a single individual. The group of individuals acting in such a manner can be referred to as a swarm. Problem-solving behavior that emerges from the interaction of such agents is called Swarm Intelligence. Computational Swarm Intelligence refers to the algorithmic models of that behavior, though the term Swarm Intelligence is most commonly used.

A. *Nature-inspired Algorithms*

Examples of emergent behavior in nature are numerous and they gave inspiration to some of the most popular algorithms. Particle Swarm Optimization (PSO) was originally inspired by the social behavior of bird flocking or fish schooling [4]. Ant Colony Optimization (ACO) is based on the foraging behavior of ant colonies [5]. BEES algorithm is a model of the colony of bees in their search for the richer and closer food source [6]. Among others, these algorithms have been applied to optimization problems and many problems that can be converted to optimization problems.

One of the most common discrete optimization problems is the Travelling Salesman Problem (TSP). It consists of finding the shortest path to connect a certain number of cities, where the cities are represented as nodes in two-dimensional space. The TSP is often used as a testbed for newly developed optimization algorithms. Various swarm-based algorithms have been used for solving the TSP in different configurations. One such configuration is a 30-city TSP, Oliver30, and the result of applying an ACO algorithm to this problem is shown in Fig. 2.

Artificial ants, unlike their biological counterparts, move through a discrete environment defined with nodes, and they have memory. In ACO, an artificial ant builds a solution by traversing the fully connected graph. When moving from one node to another, artificial ant leaves a pheromone trail on the route. The amount of the deposited pheromone may depend on the quality of the solution found. The pheromone trail attracts other ants, which by positive feedback leads to a pheromone trail accumulation. Negative feedback is applied through pheromone evaporation which, importantly, restrains the ants from taking the same route, i.e. prevents the algorithm stagnation. The algorithm, which is an iterative process, stops when one of the conditions is met; either when a satisfactory solution is found or when the maximum number of iterations is reached. The ACO metaheuristic is shown in Algorithm 1.

Algorithm 1 The Ant Colony Optimization Metaheuristic

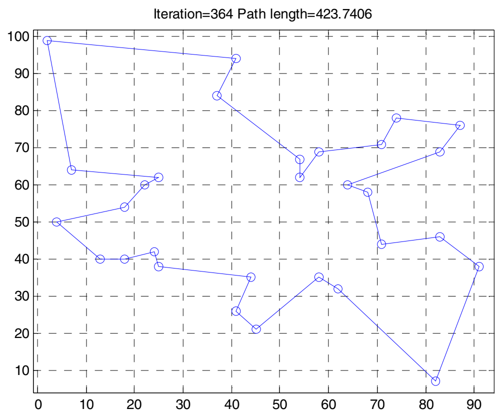
```
Set parameters, initialize pheromone trails
while termination condition not met do
        ConstructAntSolutions
        ApplyLocalSearch (optional)
        UpdatePheromones
end while
```

ACO algorithms have been used to successfully solve many complex problems, such as quadratic assignment problem [7], data mining [8], data clustering [9], image retrieval [10] and image processing for edge detection [11] and broken-edge linking [12]. In [13] the authors applied the swarm cognitive map formation to digital images to investigate adaptation and robustness of the ant-based algorithms to any type of digital habitat.

B. *Motivation for Swarm Robotics*

Swarm Robotics is a new but rapidly growing field. It is an interesting alternative to classical approaches to robotics because of some properties of problem solving by social animals, which is flexible, robust, decentralized and self-organized. Swarm Robotics is, as such, one of the newest and the most interesting applications of Swarm Intelligence.

Identify applicable sponsor/s here. *(sponsors)*



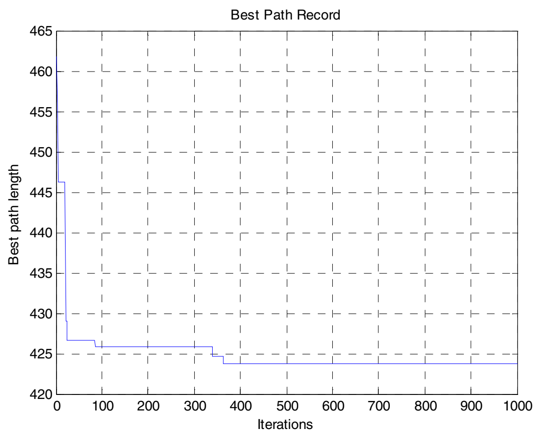


Figure 2. ACO results on Oliver30 TSP: a) best solution; b) best path convergence.

The term "swarm" in robotics is sometimes used to refer to any multi-robot system. In order to define a system as a robotic swarm, certain criteria should be fulfilled:

- Autonomy – It is required that the units that make up the swarm-robotic system are autonomous robots. They are able to physically interact with the environment and affect it.
- Large number – A large number of units is required so the cooperative behavior may occur. The minimum number is hard to define and justify. The swarm-robotic system can be made of few homogeneous groups of robots consisted of large number of units. Highly heterogeneous robot groups tend to fall outside swarm robotics.
- Limited capabilities – The robots in a swarm should be relatively incapable or inefficient on their own with respect to the task at hand.
- Scalability and robustness – A swarm-robotic system needs to be scalable and robust. Adding more robots will improve the performance of the overall system and on the other hand, losing some units will not cause the catastrophic failure.
- Distributed coordination – The robots in a swarm should only have local and limited sensing and communication abilities. The coordination between the robots is distributed. The use of a central unit for the coordination would influence the autonomy of the units.

Though these criteria are not to be used to determine whether a system is a swarm or not, they can be used to measure the degree to which the term "swarm-robotic" might apply.

We are reaching a stage in technology where it is no longer possible to use traditional, centralized, hierarchical command and control techniques to deal with systems that have thousands or even millions of dynamically changing, communicating and heterogeneous entities. The type of solution Swarm Intelligence offers is a way of moving forward when it comes to design and control of complex distributed systems.

III. Network Formation Control of a Group of Robots

Networks are the first widely applicable means of including multiple cooperating computers in the same control system. The advantages of networking in control systems are so strong that any systems above some minimal level of complexity are likely to utilize networking. However, networking in control has operational requirements that quite different from networking in a general computing environment. Not only is the selection of network type important, but the design and configuration of the network are crucial for achieving satisfactory performance.

An important concept in designing a swarm is "Multi-rate Control". Control systems may be represented in a layered manner. At the bottom is the controlled process. Next comes a first control layer in which each control loop acts on a single variable of the process, the axis level. On top of this layer comes a second layer which coordinates the actions on two or more control loops of the first layer, the axes level.

Current candidate networks for NCS implementations are DeviceNet, Ethernet and FireWire. Each network has its own protocols that are designed for a specific range of applications. Also the behavior of a NCS largely depends on the performance parameters of the underlying network, which include transmission rate, medium access protocol, packet length, and so on. There are two main approaches for accommodating all these issues in NCS design. One way is to design the control system without regard to packet delay and loss but design a communication protocol that minimizes the likelihood of these events. The other approach is to treat the network protocol and traffic as given conditions and design control strategies that explicitly take the above-mentioned issues into account. In the following section authors consider the first approach as the main approach and try to analyze the fundamental issues in using an NCS in designing a robotic swarm [14].

IV. NCS IN SWARM DESIGN

In this section, we will analyze some basic problems in NCS-based swarm design, including network-induced delay, single-packet or multiple-packet transmission of swarm inputs and outputs, and dropping of network packets.

Depending on the medium access control (MAC) protocol of the control network, network-induced delay can be constant, time varying or even random. MAC protocols generally fall into two categories: random access and scheduling access [12]. Carrier sense multiple access (CSMA) is most often used in random access networks, whereas token passing (TP) and time division multiple access (TDMA) are commonly employed in scheduling networks. In designing the swarm one can use TP or TDMA approaches and reach the same results. It is mostly recommended to use TP in a swarm. One should always remember that the main purpose of a swarm is to eliminate the role of a coordinator. In applying the TDMA on a swarm, the problem is that a synchronizing signal should be introduced in the swarm which eventually destroys the main concept of swarm design: independency of the members of a swarm from any cooperative knowledge [15].

Another issue with regarding to the design of a swarm using NCS is dropping of network packets. Network packets drop occasionally happen on an NCS when there are node failures or message collisions. Although most network protocols are equipped with transmission-retry mechanisms, they can only retransmit for a limited time. After this time has expired, the packets are dropped. Furthermore, for real-time feedback control data of each member of the swarm, such as sensor measurements and calculated control signals, it may be advantageous to discard the old, untransmitted message and transmit a new packet if it becomes available. In this way the swarm member always uses the fresh data for control calculation.

One simple solution for the packet dropping issue is to acknowledge the sending-member for receiving the data. This could add to the network traffic but in implementing the complicated behaviors, like follow-the-leader behavior, this could result in a relatively good response from the system.

Another approach in dealing with packet loss is to send each packet twice. Each member of the swarm has a responsibility to follow a common behavior. In a period of one second, a member may send multiple packets to the other nodes. Based on the dynamics of the system, one could impose a delay to the data transmission from each member. Obviously this delay should be small enough not to interfere with the dynamics of the swarm which may lead to failure of the whole mission, in other words the behavior.

Another issue is the network-induced delay when transmitting sensor data and control data. Depending on the control network protocol employed, the delay can be either constant or time varying. In case of a time varying delay, an adaptive modeling of the system (each member) should be implemented. In other words, the member should be smart enough to recognize the varying delay and try to adjust the previously-defined model of the swarm. This could add up to the overall calculations which may lead to an overhead in the CPU of the swarm member which is not desired. A time-varying delay is the most significant problem in designing a robotics swarm.

Computational complexity versus real time: In robotics, a distinction is made on internal representations of the deliberative and reactive behaviors, where the former relies on internal representations of the environment, which facilitates the use of planning of optimal paths, etc. Given constraints on the reaction time, can a trade-off be achieved between the complexity of the control algorithm and the time in which the algorithm has to terminate with an answer?

V. CONCLUSIONS

In this paper the main steps of designing a controller for a robotic swarm within a network have been described, the different approaches to deal with the delay in such system has been addressed and as an example, a TSP has been studied.

REFERENCES

[1] Giovanni C. Pettinaro and others, "Swarm Robotics: A Different Approach to Service Robotics", Proceedings of the 33rd ISR (International Symposium on Robotics), October 2002.

[2] K.N. Krishnanand and D. Ghose, "Glowworm swarm based optimization algorithm for multi-modal functions with collective robotics applications," Multi-agent and Grid Systems, Issue 3, Volume 2, 2006, pp. 209 - 222.

[3] Jevtić, A. and Andina, D. Swarm Intelligence and its Applications in Swarm Robotics. *Proceedings of the* 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics 2007 (CIMMACS '07), pp. 41-46, ISBN: 978-960-6766-21-3, Tenerife, Canary Islands, Spain, 14 -16 December, 2007.

[4] Kennedy, J. and Eberhart, R. C. Particle swarm optimisation. *Proceedings of the* IEEE international conference on Neural Networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.

[5] Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Computational Intelligence Magazine*, vol.1, no.4, pp.28-39, Nov. 2006.

[6] Pham D.T., Ghanbarzadeh A., Koç E., Otri S., Rahim S., and M.Zaidi. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. *Proceedings of* IPROMS 2006 Conference, pp.454-461.

[7] M. Dorigo and T. Stutzle. ACO algorithms for the quadratic assignment problem. *New ideas in optimization*, McGraw-Hill NewYork, page 33-50, 1999.

[8] A.A. Freitas, R.S. Parpinelli and H.S. Lopes. Data mining with an ant colony optimization algorithm. *In IEEE Transactions on Evolutionary Computing*, special issue on Ant Colony Algorithms, 6(4):321-332, 2002.

[9] J. Handl and B. Meyer. Ant-based and swarm-based clustering. Swarm Intelligence 1(1): 95-113, 2007.

[10] P. Pina, V. Ramos and F. Muge. Self organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies. *In 2nd International Conference on Hybrid Intelligent Systems*, vol 87, ISBN: 1-5860-32976, pp 500-509, Santiago, Chile, 2002.

[11] A. Jevtić, J. Quintanilla-Dominguez, M.G. Cortina-Januchs and D. Andina. Edge Detection Using Ant Colony Search Algorithm and Multiscale Contrast Enhancement. *2009 IEEE International Conference on Systems, Man, and Cybernetics* (SMC2009), San Antonio, Texas, USA, October 11-14, 2009.

[12] A. Jevtić, I. Melgar and D. Andina, "Ant Based Edge Linking Algorithm", *The 35th Annual Conference of the IEEE Industrial Electronics Society* (IECON 2009), Porto, Portugal, 3-5 November 2009.

[13] V. Ramos and F. Almeida, Artificial ant colonies in digital image habitats - a mass behaviour effect study on pattern recognition. In Dorigo, M., Middendorf, M. and Stuzle, T. (Eds.): From Ant Colonies to Artificial Ants - 2nd International Workshop on Ant Algorithms, 2000.

[14] By Wei Zhang, Michael S. Branicky and Stephen M. Phillips, “Stability of Networked Control Systems”, IEEE Control Systems Magazine, Feb. 2001, pp 84-99.

[15] J. Colandairaj, G.W. Irwin and W.G. Scanlon, “Wireless networked control systems with QoS-based sampling”, IET Control Theory Appl., Vol.1, No. 1, January 2007, pp 430-437.