A unified methodology for swarm algorithm development and evolution

Rene Jespersen - Simon Bjerg Mikkelsen - Trung Dung Ngo

Department of Control - Aalborg University

ABSTRACT: Swarm robotics is a biologically inspired field which inherits significantly from the observation of biological populations such as social insects like ant colonies, termites, bees, and wasps, inspiring the vision of how a large number of simple individuals can collaborate with other members to release intelligent systems. However, it is as a complex system hard to be understood, analyzed, and synthesized and released in our daily life because of the lack of general methodology. To replicate this in the real world, an empirical methodology is required to overcome the practical challenges which arise when working with multiple robots. In this paper, a unified methodology for on the fly programming, debugging, analyzing and synthesizing based on Matlab Simulink and Real Time Workshop will be developed. The kinematics and dynamics of mobile robots as well as swarming algorithms will be virtually implemented in Simulink models which are easily modified and updated. A graphic user interface (GUI) is built up for on the fly debugging and analyzing process with sensory data real-time updated from the robots. Instead of hand-on programming the robots, an automatic code interpreter managed by Real-time Workshop will synthesis functions modeled in Simulink in order to generate and download the code to the robots via wireless communication directly. We expect, that by using the closed loop developing model - a unified methodology - the swarm algorithms will be empirically evolved, and decentralized control of robots will be reliably released.

Introduction

Nature has many fascinating creatures which have adapted to the environment they live in through millions of years, to achieve a better chance of survive. The result is very interesting knowledge from the perspective of an engineer. The shape, physics and behavior of living creatures have been studied to replicate the functionalities with the advantage of improving human made designs. Particularly the swarming behaviors of animals like ants, wasps and birds have given huge inspiration. The usability of a swarm is larger than the sum of the individuals, which can make it more flexible, stronger, more robust and adaptable, etc. An example of this is a flock of birds forming a V-shape to reduce the air resistance and to gain a lift. Hence the individual bird is using less energy to travel a longer distance than it would do on its own. Another example is that ants stand on top of each other to create a bridge for the colony members to cross over gabs much longer than their body length.

These collaborative methods have been replicated by humans for thousands of years but the engineers of today are applying them to robots for use in hazardous environments like a rescue recovery or in war zones etc. One could imagine a swarm of robots send into a crashed building searching for survived people, and by working together they can cover a larger area faster. Animals living in swarms do not have a specific leader to distribute tasks, which reminds of a decentralized system as we know it from distributed system in computers and networks.

Replication of bio inspired formations into algorithms applied on robots is the main topic of our project, and they will be tested in a simulated environment as well as a real world arena. When working with swarms of robots the most important subject is the development of swarm algorithms. Due to the traditional approach, it is a time consuming task including hands-on programming, analysis, synthesis, and even evolution. For this reason a unified methodology is needed to ease such a task when developing and verifying swarm algorithms, which is the topic of this paper.

Related Work

A swarm of mobile robots like social insects can be understood as complex systems because of its properties, e.g. simple and random individual behaviors vs. robustness, adaptability, and scalability of well-organizing group. It is therefore hard to deal with such systems in formal methods. There are many different approaches to understanding, analyzing, and synthesizing the systems with an unlimited range of mathematical modeling, simulation, and empirical experiments.

Mathematical modeling is a formal method of engineering scientists, mathematicians and biologists to have better understanding and prediction about the natural behaviors of social insects. Therefore, it is often used to modeling and analyzing collective actions of robot swarms. There are two tendencies of mathematical modeling named microscopic (bottom-up methodology) and macroscopic (top-down methodology) models. In microscopic models [e.g., see 3,4,5], the swarm model is built up by synthesizing instinct arbitraries of individual robots for a simulation. Although this method might provide prediction of internally sophisticated oscillation of robot swarms, it obtains many drawbacks of neither clarifying influences of external events to the swarm nor indicating the computation cost according to exponentially increased complexity. In contrast, the macroscopic model [e.g., see 2,7] is seemingly preferable because it can overcome the drawbacks of the microscopic one by abstracting the swarm behaviors at very high level while ignoring the basic instinct of individual robots. The model is easier to build up and it can take all external influences into the overall systems without computation cost. However, the model is only used to predict the general properties of the swarm while property of individuals is not traceable. Ultimately, both mathematical models can only be used to get better understanding about the swarm behaviors, it is not clear on how to transfer the simulation results of the models to the real world.

Another methodology of swarm algorithm development is based on simulation evolution [e.g., see 6]. Evolutionary algorithms in association with neutral networks are usually used to improve the fitness function by selecting the best candidates in each generation for the next generation evolution. The overall behaviors of robots are evolved and trained through a computer simulation. It is then synthesized in order to generate the decentralized control for real robots. Although this method has high potential in reducing the computation cost and time consuming in swarm algorithm development, it still has the significant problem of transferring the simulated code to the real robot due to the big gap between the simulation environment and the real world, which is not easy to fill up.

A practical methodology to the swarm algorithm development is based on experiments of real robots [e.g., see 8,9]. The swarm algorithm developed relies on observation and data collection of experiments that makes a true sense about the swarming behaviors in the real world. However, it is not easy to transfer the knowledge synthesized from the observation and data analysis to the swarm algorithms and decentralized control of robots. It is a hard and expensive job due to doing hand-on coding for many robots with heterogeneous bodies.

Although existing approaches contributes to the step of understanding the complexity of robot swarms as well as development of swarm algorithms and decentralized control, none of those are realistic approaches to the swarm algorithm development and evolution.

Methods

This paper proposes a new methodology for developing and testing swarming and formation algorithms. This methodology incorporates many of the good features from the methodologies mentioned in Related Work. It is focused on bridging the gaps between simulation and real the world as well as increasing understanding of swarms by combining the different approaches and streamlining analysis.

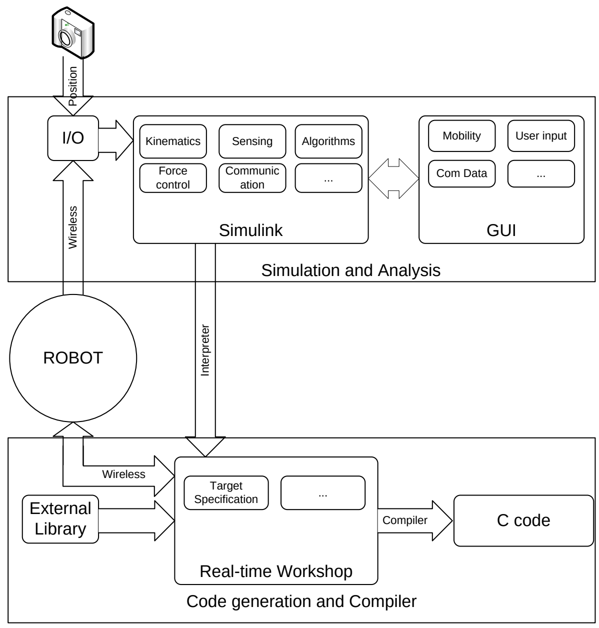

Figure 1. Diagram of the proposed methodology

The proposed methodology involves three main function blocks working in a closed loop for development and evolution: Matlab Simulink, GUI, and Real-time Workshop. On one hand, swarm algorithm can be internally simulated through mathematical models built in Simulink, and shown in GUI for pre-analysis. On the other hand, the sensory data can be directly imported in the same model for the empirical verification. Analysis of data from experiment and/or simulation are presented in a high level GUI making it increasingly easier to compare, and therefore transfer results from mathematical models to the real world through the Real-time Workshop. The Real-time Workshop plays a role as an interpreter that translates model-based algorithm and control to C code before compiling to machine code and downloading to the robots. The wireless connections to and from the robots are imperative to the proposed methodology for two reasons. Firstly sensory data is needed in real-time to be imported into the Simulink for processing and analysis. Secondly in order to automate the expensive job of hands-on-coding of robots, the machine code can be remotely downloaded to the robots. To achieve full hands-off coding of robots and keep focus on the model-based swarm algorithms, the GUI allows to change parameters on the fly. A diagram of the methodology is shown in figure 1.

Development

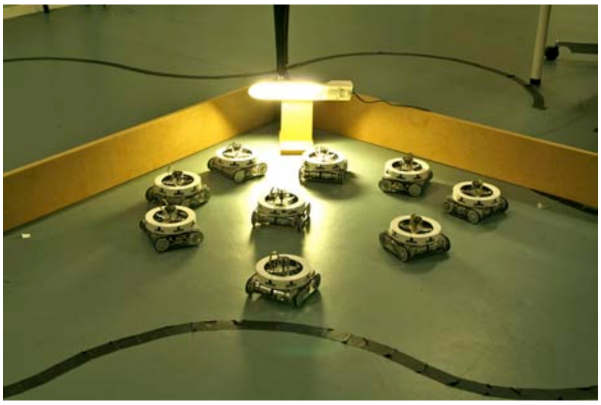

Figure 2. The developed robots.

A set of 10-15 small simple robots is provided to us by the university, from a previous project as illustrated on figure 2. These 13 by 13 cm robots are driven by a tank drive train powered by two small electrical motors. The drive train gives the individual robots great moving ability since they are able to turn around on their own center axis. Hence it is able to change position fast with minimal use of movement. Furthermore they are able to change the drive train between the caterpillar tracks and a four wheel drive, depended on how the tracks are attached. This gives advantages in rough terrains. The main circuit-board of the robots is provided with an ATmega128 microprocessor which is programmed trough an In System Programming port. The microprocessor can however be programmed using any of its communication ports. Furthermore is a set of 8 infrared distance sensors attached around the sides, and one on the top with a rotating mirror, able to scan the surroundings. The robots come with a 300 cm x 270 cm arena with 15 cm high side walls ensuring that no robot can escape. The arena is covered with felt carpet which gives good traction.

The kinematic model of the robots is implemented in the Simulink Model for simulations. In this way the output from simulation and experiments will be directly comparable. Also A communication model is developed for receiving and sending data from the PC's wireless communication port into the Simulink model. Simulation of the swarm will give values and formation to compare to the real world measurements.

On top of the Simulink model a GUI is designed to show all sensor data, as well as the formation of the robots. Furthermore it adds ability for a user to change the parameters of the algorithms on the fly for faster tuning of these. By using Real Time Workshop combined with a GCC compiler the code generation from the algorithms input to the GUI is automated. This and the ability to program the robots through a wireless connection, as demanded in the methods, automate many time consuming jobs.

The existing robots will be expanded with wireless radio communication. To implement the wireless link a new Printed Circuit Board (PCB) will be made for the robots. The PCB will be designed using premade radio modules. This removes the need to design an antenna and communication stack. An ATMega128 Microprocessor will be used on the board. This is necessary since the tasks for the new PCB will be to handle all radio communication, to control the motors and wheel encoders and to detect if there is floor beneath (cliff sensors). All of this cannot be implemented on the existing board since there are not enough serial UARTs, and ports left. The only available communication port is a two wire serial interface. Therefore the new PCB will interface with the Old board using i2c. It would be desirable to have a radio with low power consumption for sensor data and a high bandwidth for programming. Since high bandwidth and low power consumption are conflicting requirements two different radios will be used.

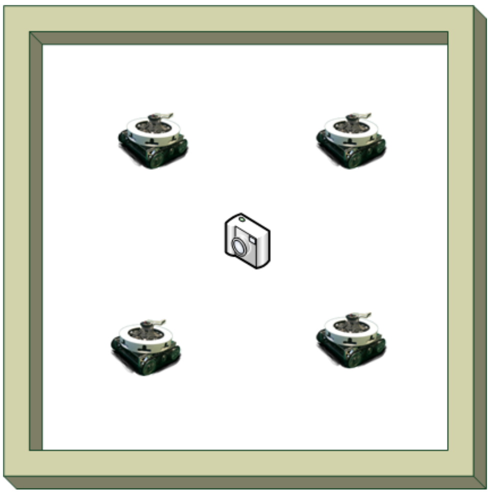

Figure 3. Arena and pGPS camera

The existing arena gives a well defined environment. However consistency in measurements still demands a proper sensor setup. A pseudo-Global Positioning

System (pGPS) in the form of a camera fixed above the arena is implemented as shown on figure 3. The camera will be used for visual as well as position documentation. The pGPS data can be used as input to algorithms on the robots as well. By implementing the pGPS in the arena the verification system is not dependent on the robots. This can be an advantage since data delay and processing is not affected by algorithm implementation.

The precision of pGPS depends on the resolution of the camera compared to the arena size. Software is used to identify robots and track their movement in the picture frame. The picture frame is distorted since the distance from the camera lens to all points on the arena floor is not the same. Software to overcome this is available as open source. [1]

Results

At the time of writing the development is still a work in progress based on the methodology. However the methodology is set and the development is based on this. The results and conclusion are based solely on the methodology.

- The analyses of the swarms are eased by the GUI leading to better understanding. Since the GUI is the only needed interface to the system focus can be held solely on algorithms.
- Real time comparison of sensor data and simulation.
- The task of programming the robots is simplified significantly.

Discussion and preliminary conclusion

This paper is to illustrate the methodology of a larger project. In this paper, the development of a productive methodology has been in focus. It is of most importance when working with a large number of robots that it is easy and fast to try out developed algorithms and non the least to improve on them. The methodology proposed in this paper is model-based, which ease understanding and modification in development pipeline. It is believed that this will save time in the evolution of robot swarms

A wireless communication module for the robots is produced and connected to a Matlab Simulink model and GUI. On one side, it makes the uploading and changing of parameters in algorithms fast and easy, without having to plug and unplug the individuals to program them. On the other side, it allows to download the code to several robots or to control them simultaneously. More specifically, the wireless link opens possibilities for many options which can be operated real time. We are able to read out sensor data, which can save a lot of time in debugging. We can determine whether the robots should be able to communicate with others that are not direct neighbors. The parameters of the algorithms can be tuned or optimized automatically real time, through a real world experiment. It is achieved by using measurements from sensors, to use either in a simulation or in the experiment itself.

By adding a camera above the arena, we are able to locate the robots position and orientation which can be used as a factor when developing the swarm algorithms. It makes it easier to achieve the correct positions. Furthermore it is important that the algorithms can be tested, meaning that the robots are doing what they are supposed to do. The position is useful to test whether the result is accepted or not.

The use of known standards in components is believed to ease the development of the hardware as well as the software. They are all well documented and hold a lot of examples. In time of writing the test setup is not completely finished, but the communication hardware is designed and ready for production, and the software combined with the Simulink model is designed as well.

An analysis on some of the work, which is related to this project, is made. We have taken a look into the state of the art projects which have given us a great insight in the possibilities and limitations of swarm robots.

Now the overall methodology and the progress of the development has been outlined. Next the rest of the work will be discussed, and will mainly focus on the development of algorithms.

Future Work

The hardware and software based on the methodology presented in this paper should be finished next. When it is done the work can begin on the applications it is designed for; namely swarm algorithms. Ideally when this setup is up and running the shift from one algorithm to another is just a click in the Matlab GUI. To outline our thoughts on the rest of the project, some of the future perspectives that could be interesting to look at are described in the following.

- Develop swarm algorithms capable of tuning their own parameters runtime. Algorithms that adjust its parameters to fit the present environment would be an interesting step in the evolution of swarm robotics.
- Test heterogeneous algorithms i.e. not all robots having the same algorithm. Just like ants in nature have different jobs as guards, food finders or food carriers. Challenges would be to get robots with different algorithms to collaborate on the same task.
- Adapt the setup, to work via Internet making a remote lab accessible for anyone around the world that needs to test an algorithm. Considerations about battery time cost and local assistance would be some of the big challenges.

References

[1] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems. In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 4004-4010. IEEE, 2008.

[2] O.Soydal and E. Sahin. A macroscope model for self-organized aggregation in swarm robotic systems. In E.Shain, W.M.Spears and A.F.T.Winfield, editors, Swarm Robotics- Second SAB 2006 International Workshop, volume 4433 of LNSC, pages 27-42, Berlin, Germany, 2007, Springer-Verlag.

[3] Tamas Vicsek, Andras Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-drived particles. Physical Review Latters, 6(75): 1226-1229, August 1995.

[4] Tad Hogg, Coordinating microscopic robots in viscous fluids. Autonomous Agents and Multi-Agent Systems, 14 (3): 271-305, June 2006.

[5] A.Galstyan. T.Hogg, and K.Lerman. Modelling and mathematical analysis of swarms of microscopic robots. In Proceedings of IEEE Swarm Intelligence Symposium (SIS-2005), Pasedena, CA, pages 201-208, Los Alamitos, CA, June 2005, IEEE press.

[6] V.Trianni, Evolutionary swarm robotics – Evolving self-organizing behaviours in groups of autonomous robots. Vol. 108 of Studies in Computational Intelligence, Springer-Verlag, Berlin, Germany, 2008

[7] A.Martinoli. Swarm Intelligence in autonomous collective robotics: from tools to the analysis and synthesis of distributed control strategies. PhD thesis, Ecole Polytechniques Federale de Lausanne, 1999.

[8] D. Payton, M. Daily, R.Estowski, M.Howard, and C.Lee. Pheromone robotics. Autonomous robots, 11(3): 319-324. Nov. 2001

[9] F.Mondada, L.M. Gambardella, D. Floreano, S. Nolfi, J.L.Deneubourg, and M.Dorigo. The cooperation of swarm-bots: physical intefaction in collective robotics. IEEE Robotics & Automation maganize, 12 (2): 21-28, June 2005.

Appendix

A video from a big industrial fair in September 2009 at Messe Center Herning in Denmark was recorded, showing the Aalborg University stand. The swarm robots are presented in their arena, showing them working at a very early stage of development.

http://www.control.aau.dk/~borch/Herningmesse2009/Report/index.htm