ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152www.ijmece .com

Vol 8, Issue 3 Aug 2021

Multi-Robot System Pattern Formation and Adaptation: A Review

Mr.D.VEERANNA[1], Dr.K.AMIT BINDAJ[2],Ms.P.NAGA LAXMI[3],Ms.Y.ANITHA[4],Mr.Ch.GOPALA RAO[5],

Abstract

Recent robotics advancements are making it possible to deploy large numbers of low-cost robots for tasks like surveillance and search. However, coordinating a group of robots to perform these kinds of tasks is still difficult. Recent research papers on multi-robot systems are summarised in this report. It's divided into two sections. The first section covered research into the pattern formation problem; specifically, how robots can be commanded to form a pattern and keep it. The second section examines the research into adaptive strategies for managing networks of robots. In particular, we've looked into (1) how evolution is used to generate group behaviours, and (2) how learning (lifelong adaptation) is used to make multi-robot systems respond to changes in the environment and in the capabilities of individual robots.

Introduction

Recent robotics advancements have made it possible to deploy large numbers of low-cost robots for tasks like surveillance and search. However, coordinating a group of robots to perform these kinds of tasks is still difficult. Previous reviews on multi-robot systems have taken a more generalised approach (see, for example, Cao et al.[25] and Dudek et al.[7]). In contrast to these, the scope of this report is limited to the most up-to-date research on pattern formation and adaptation in multi-robot systems. The document can be split into two sections. In the first section, we looked at previous research into the "pattern formation problem," or the question of how a group of robots might be commanded to form and keep a pattern. The second section examines the research into adaptive strategies for managing teams of robots. We have looked into (1) how evolution is used to generate group behaviours and (2) how learning (lifelong adaptation) is used to make multi-robot systems respond to changes in the environment as well as in the capabilities of individual robots.

The emergence of patterns in networks of multiple robots

.

The pattern formation problem entails getting a group of robots to form and stay in a specific formation, like a wedge or a chain, and maintaining that formation. Pattern formation has many current uses, including in rescue missions, landmine clearance, space exploration, satellite array control, and unmanned aerial vehicle navigation (UAVs). Cooperative behaviours among members of different animal species have also been observed to lead to pattern formation. In these cases, individuals maintain a consistent orientation and distance from one another while moving, or they fill a given area as homogeneously as possible. Flocking birds, shoals of fish, and ant chains are all examples of animals forming patterns[18]. We have divided the research on pattern formation into two categories. In the first category are studies in which coordination is handled by a command centre that has full visibility into the operation and can issue instructions to each robot as needed. In the second category, we find approaches to coordination through the formation of distributed patterns

Professor[2], Assistant Professor[1,3,4,5],
Mail Id : dev.veer57@gmail.com, Mail Id : karpurapu.gavasj@gmail.com, Mail id : nagalaxmi.sbit@gmail.com,
Mail Id : verasangianitha478@gmail.com, Mail Id : chintala.gopal4@gmail.com ,
Department of ECE, Swarna Bharati Institute of Science and Technology (SBIT),
Pakabanda Street,Khammam TS, India-507002

ISSN 2321-2152www.ijmece .com

Vol 8, Issue 3 Aug 2021

Institutionalized patterning

A computational unit coordinates the activities of the entire group in centralised pattern formation methods[3, 13, 23, 24]. The robot's motion is then relayed to it over a data connection. To ensure that a group of robots travels in the desired formation along a predetermined path, Degerstedt and Hu[13] propose a coordination strategy to achieve this goal. The process of path planning is handled independently from path tracking. Both the centralization and the tracking of virtual reference points are handled independently. A virtual leader's trajectory is calculated to serve as a point of orientation for the robots. A trio of virtual robots were guided around a virtual obstacle using this strategy. Here, the robotsthat met at the base of the triangle detoured around something that had landed in the middle of the robots. There is conclusive evidence in the paper that the described method stabilises the formation error if the robots' tracking errors are bounded or if tracking is done perfectly. Unmanned aerial vehicles (UAVs) can be flown in formation with the help of a centralised path-planning method, as proposed by Koo and Shahruz [23]. One UAV, the leader, is more powerful and capable, and it calculates the course for the others to follow. Cameras and sensors are only available to the leader. It uses a communication channel to instruct the other UAVs on what paths to follow. In order to follow their paths, UAVs need to take off and fly in that direction. Experiments take into account both a scenario in which UAVs launch one by one and another in which they launch all at once. Computing trajectories is the main focus of this research. A centralised trajectory computation scheme based on kinetic energy shaping is proposed by Belta and Kumar [3]. They use a kinetic energy metric that gradually shifts over time rather than a constant one. The procedure creates seamless paths for a group of mobile robots to follow. One parameter allows the user to adjust the distance between the robots. The approach is not scalable, however, because it does not account for avoiding obstacles. Kowal czyk [24] details an assignment strategy for the target-based formation-building problem. The algorithm starts with a dispersed group of robots and assigns them each a location on the final formation's target point. Then, it plots out the robots' priorities and paths so that they don't run into each other on the way to their destinations. There is a buffer zone around each robot's path where slower robots can't go. The robot will wait until the higher priority robot moves out of the way if its path takes it through an area that is off limits to it. Both holonomic and non-holonomic robots are used to evaluate the method. The methodology presupposes the availability of a centralised processing power and global sensing capabilities. Lack of consideration for the method's scalability. Strategies for centralised pattern formation presume the presence of a communication channel between the coordinating node and the individual robots, and rely on a coordinating node to oversee the entire group. This centralised approach is less scalable for controlling a large number of robots, more expensive to implement, and less reliable due to the underlying assumptions. Decentralized strategies for pattern formation are another option.

Adaptation in multi-robot systems

In this section we review the studies that used adaptation strategies in controlling multirobotsystems. Specifically we have investigated (1) how learning (life-long adaptation) is used to make multi-robot systems respond to changes in the environment as well in the capabilities of individual robots, and (2) how evolution is used to generate group behaviors. In multi-robot systems, adaptation can be achieved at two levels: group level and individual level. We classify the recent studies into these levels and review them in the following subsections.

Adaptation at the Individual Level

When the state space is too large, reinforcement learning models become ineffective. Rather than relying on a single, overly complex learning module, splitting it up into separate modules for each state can help. The research conducted by Takayashi [[26]] is one example. The issue he focused on was a scaled-down version of the robo-soccer challenge. It's assumed that adversaries employ a variety of strategies. Each module includes its own set of predictors and planners. The predictor makes guesses about the opponent's next move based on the latter's historical patterns of behaviour. Conversely, a planner will use this forecast to generate a set of actions that will maximise success. As a result of this competition, only the best predicting module is consistently reinforced. By doing so, we can develop unique modules to counter various enemy strategies. Ball chasing with a randomly moving opponent is the problem used in this study. Compared to learning

individual modules, the outcomes are superior. Given enough trials, reinforcement learning converges to the best policy, but in practise, these numbers are often unfeasibly high. To accelerate the learning process, Piao[20] suggests a refined version of the reinforcement learning technique. Essentially, the method is a set of behaviour rules for individual states that are the result of a synthesis of rule learning, reinforcement learning, and action level selection.

To form the rule base, we use "instances," which are essentially states that have traversed some kind of predetermined threshold. After each epoch, the instances are given names based on the data collected during that time period. The data from these examples is then used to formulate rules. The guidelines herein serve as a

prohibitive rules intended to discourage wasteful or harmful behaviour. The selection of actions at the action level is governed by predetermined rules that govern the robot's overall tactic. Together, sensor data and action level are fed into reinforcement learning to produce state information. The module that learns to generate actions from sensory data and action levels is called reinforcement learning. In this case, Piao uses it to solve the robo soccer issue. In contrast to conventional Q learning, he claims that learning with multiple robots yields superior results. Since reinforcement learning was developed with a focus on the performance of individual agents, it lacks features that would enable the facilitation of social behaviours. The research of Tangamchit[16] addresses this issue. The paper discusses the split between systems that operate at the activity and task levels. Action-level systems produce reactive behaviours in order to address issues. In contrast, task-level systems create tasks from a collection of smaller tasks, which can then be delegated to multiple agents. When it comes to robots, Tangamchit defines cooperation as a task-level activity in which resources and responsibilities can be shared. Both international and regional incentive plans are taken into account. One's contribution to the group's overall reinforcement is multiplied in the global re ward scheme. The local reward scheme is different because the reward is not shared among the group members. Q-learning and Monte Carlo learning are two of the considered learning algorithms. When determining the value of each action in each state, Q-learning uses cumulative discounted rewards, while Monte Carlo learning uses averaging. Each action taken in a state will earn the same reward regardless of which episode it is. This strategy is less efficient because it fails to take advantage of later actions in an episode that are more likely to result in a positive outcome for the player. Puck collecting behaviour, a special case of the foraging problem, is investigated here. Pucks can only be collected and deposited in the trash can by robots. Every action, with the exception of putting down a puck, has a punishing consequence. There is a puck-free "home" zone, a puck-filled "deposit" zone, and pucks all over the field. In order to accomplish this, we employ a pair of very different robots. Out in the wider world, the first robot's mobility and collection abilities shine.

The second robot can only move around its home area, but it is more effective at the bin deposit action. Collaborating robots are necessary for the best strategy, as they must first bring pucks into the home region and then deposit them there. That sort of learning is reserved for specific tasks. The findings show that local rewards or discounted cumulative rewards, like those used in Q learning, are ineffective for teaching task-level cooperation. However, when average rewards are combined with global rewards, cooperative policies emerge. To incorporate domain knowledge, reinforcement learning only needs feedback for the applied sequence of actions. A common method of incorporating this is through the use of multiple reward functions. When it comes to the role of rewards in a foraging task, Mataric[14] talks about it. Though easy to analyse mathematically, single-goal systems often lead to difficulties in behaviour acquisition. Converting behaviours, especially those that are conditional or sequential, into a single, unified goal function is challenging.

The alternative is to use multiple goal functions, each of which describes a different subgoal of the agent. Estimators of future progress are another development. These approximators provide an approximate measure of progress toward a given objective. Using this area's domain knowledge is greatly enhanced by the aforementioned two enhancements (by designing appropriate subgoals and estimating progress of the subgoals). Because not only the final objective but also intermediate steps are reinforced, they provide much more reinforcement than conventional methods. Robots performing a real-world foraging task are used to evaluate the effectiveness of the new method. Pucks will be gathered and delivered to your house by robots. Also, robots are expected to make regular appearances at home. Some basic reactive

behaviours are taught to the robots in order to make the state space of the learning problem more manageable. Pucks are picked up when the agent is in front of them, obstacles are avoided, and pucks are dropped when the agent returns home. The optimal policy is generated by hand and then compared to the experimental results. Findings support the usefulness of both planned enhancements. The paper makes a fascinating observation about the disruption brought on by agents. The rate of learning and the degree of convergence suffer as the number of learning agents increases.

Cooperation is achieved in Parker's[6] L-ALLIANCE model through the use of various behaviour sets and worldwide communications. A watcher is assigned to each set of behaviours. These watchdogs check the necessary conditions for activating behaviour sets and evaluate the agent's and other agents' abilities. Parker presents a pair of drives, impatience and acquiesces cence. Apathy describes a propensity to allow other robots to carry out a task that one's own robots could do, while impatience corresponds to a propensity to take over a task that's already in progress. The L-ALLIANCE framework modifies these intrinsic motivator settings while the learner is in progress. For this design to work, robots must constantly report their status to one another. This design presupposes that the robot is responsible for any potential environmental changes that result from its declared actions. That solves the issue of giving credit where credit is due. Because of its flexibility and adaptability, the L-ALLIANCE architecture is well-suited for managing diverse teams and keeping up with developments in robot capabilities. To solve the credit assignment problem, however, L-ALLIANCE necessitates global communication and a bold assumption. According to Goldberg et al. [4], Augmented Markov Models are a promising method for achieving this goal (AMM). An AMM is a Markov model with supplementary transition statistics. Not a policy generator, but rather a tool for learning from environmental data. In contrast to Hidden Markov Models, AMMs operate under the premise that the precise nature of an action's execution is known in advance. As a type of Markov model, AMMs are of the first order, but they are constructed in stages. With this enhanced cognitive capacity, they will be better able to approximatively handle systemic high-order transitions. [2] Their research integrates AMMs with behavior-based robotics. AMMs with varying time resolutions are used to keep an eye on each type of behaviour. This enables the system to quickly and precisely react to any changes in its surrounding environment.

Adaptation on a social scale

When applied to multi-robot systems, reinforcement learning is inefficient because, by definition, it is centralised. Opportunistically cooperative neural learning, which Yanli researched in his study [27] proposes as a compromise in the centralised vs. decentralised learning debate. Each agent in a pure decentralised learning model keeps its own learning data private. Since the group won't benefit from everyone's shared knowledge, this is a major setback. By incorporating 'opportunistic' search, Yanli is able to address this issue. A concept from genetic algorithms called "survival of the fittest" is conceptually similar to this approach. To increase their efficiency, less-fit networks often mimic the behaviour of more-fit ones.

Yanli compares three scenarios, including a centralised setup, a decentralised setup, and an opportunistically decentralised setup. All of these scenarios are put to the test on a searching task, in which agents are tasked with exploring as much of a specified area as possible while minimising the number of times they have to make a pass through it. Working together is clearly the most effective tactic. Each agent works in concert with the others and makes preemptive plans for their actions. Plans are discussed amongst agents as well. Using these strategies, each agent can anticipate the subsequent actions of all other agents. These predictors are learning machines. As soon as other agents' next moves can be accurately predicted, rewards can be calculated with greater precision. According to the findings, central learning outperforms all of these approaches. However, there are a number of issues with fault-tolerance and communication that plague central learning. It turns out that OCL (opportunistically cooperative learning) is nearly as effective as central learning, and that both are significantly more so than the distributed-only case. Agah[1] incorporates both personal and social change into his writing. Agah tackles the problem of teaching multiple robots at once with the help of the so-called Tropism Architecture. The tropics architectural style acts as a bridge of comprehension between perception and behaviour. A tropism is a predisposition to react to specific stimuli.

Learned tropes are stored in the tropism architecture (i.e. state, action, tendency pairs). Agents' choices are determined by how well their preconceived notions of the world align with the actual world. A stochastic process is used to determine which actions to apply biassed on the tropism values. This architecture makes use of both supervised and unsupervised learning. In a self-directed learning system, the database of tropisms is continually refined through the incorporation of new information garnered from the surrounding environment. When a state is updated, the action is changed when an invalid or negatively reinforced action is encountered, and the tropism value for a positively reinforced pair is increased. Each agent's list of tropes is encoded as a sequence of bits with varying lengths in order to facilitate population learning. A genetic algorithm is then executed on these sequences of binary digits. Each organism's "fitness" is determined by how much it learned and how many rewards it received. While Q-learning relies on reinforcement propagation, these findings suggest that this dual approach is also effective. It's not always practical to have predetermined behaviours, and sometimes it's necessary to learn behaviours from scratch. In this sense, we can think of hexapod locomotion. Research by Parker[8] on hexapod robots learning to perform a cooperative box pushing task. The primary challenge he faces is the locomotion problem, as hexapod robot movement is more complex than that of wheeled robots. Parker designed Cyclic Genetic Algorithms (CGA) specifically for this task because they can handle the complex control needs. The goal of a CGA is not to evolve a simple stimulus-response pair, but rather a sequence of operations. CGA stores a sequence of actions that the agent must perform repeatedly.

By pairing the chromosome under evaluation with the optimal solution to the problem, a computer simulation can determine how well suited each chromosome is for the task at hand. The chromosome's fitness is calculated based on the collective's success. The outcomes support the usefulness of the planned approach. For robots to work together, they must be able to coordinate their efforts with one another. Peer-to-peer communication models were used in early cooperative approaches. While this may be necessary for an optimal solution, doing so would necessitate ever-increasing computational power and bandwidth to handle the growing number of robots. It's true that establishing channels of communication on a local level helps ease communication bottlenecks, but this is still an issue. An approach to overcoming this communication barrier is stigmergy, or talking to people or things in the environment. Scalability is achieved through an implicit communication system, as seen in social insects. The work of Yamada[28] provides a functional example of an implicit communication system for group robot cooperation. The problem of pushing boxes is used to illustrate this method. A light beacon indicates the location of the goal, and it is assumed that the robots can sense the motion of the box they are pushing, the presence of other robots, and the boundaries of the room. In this model, walls are modelled as rigid cubes that are ultimately disregarded. In order to address the issue of implicit communication, the authors create fictitious scenarios. Conditional abstract world models are computed from sensor data and very basic memory structures (such as counters for some sensor read ings). For every possible circumstance, robots have their own sets of rules. The data gathered by sensors is used to inform these rules.

Conclusion

Recent research on multi-robot system pattern formation and adaptation was summarised. There are two distinct categories within the research on pattern formation. In the first category are studies in which coordination is handled by a command centre that has full visibility into the operation and can issue instructions to each robot as needed. In the second category, we find approaches to coordination through the formation of distributed patterns. Research on controlling multi-robot systems with adaptation strategies can be broken down into two categories: group and individual.

References

[1] Agah A. Phylogenetic and ontogenetic learning in a colony of interacting robots. Autonomous Robots, 4(1)(3):85–100, 1996.

[2] Brooks R. A. Intelligence without representation. Artificial Intelligence, 47:139–159, 1991.

[3] Belta C. and Kumar V. Trajectory design for formations of robots by kinetic energy shaping. In Proceedings. ICRA '02. IEEE International Conference on Robotics and Automation, 2002.

[4] Goldberg D. and Mataric M.J. Maximizing reward in a non-stationary mobile robot environment. invited submission to the Best of Agents-2000, special issue of Autonomous Agents and Multi-Agent Systems, 6(3):67–83, 2003.

[5] Dudenhoeffer D.D. and Jones M.P. A formation behavior for large-scale micro-robot force deployment. In Simulation Conference Proceedings, 2000.

[6] Parker L. E. Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual robot performance. Autonomous Robots, 3(8):239 – 267, 2000.

[7] Dudek G., Jenkin M., and Milios E. A taxonomy for multi-agent robotics. A K Peters Ltd, 2002. Chapter 1.

[8] Parker G.B. and Blumenthal H.J. Punctuated anytime learning for evolving a team. In Proceedings of the 5th Biannual World Automation Congress., pages 559–566, 2002.

[9] Fredslund J. and Mataric M.J. A general algorithm for robot formations using local sensing and minimal communication. IEEE Transactions on Robotics and Automation, 18(5):837– 846, 2002. [10] Desai J.P. Modeling multiple teams of mobile robots: a graph theoretic approach. In Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.

[11] Fujibayashi K., Murata S., Sugawara K., and Yamamura M. Self-organizing formation algorithm for active elements. In Proceedings. 21st IEEE Symposium on Reliable Distributed Systems, 2002.

[12] Sugihara K. and Suzuki I. Distributed algorithms for formation of geometric patterns with many mobile robots. J. Robot. Syst., 13(3):127–139, 1996.

[13] Egerstedt M. and Hu X. Formation constrained multi-agent control. IEEE Transactions on Robotics and Automation, 17(6):947–951, 2001.

[14] Mataric M.J. Reward functions for accelerated learning. In Machine Learning: Proceedings of the Eleventh International Conference, pages 181–189, 1994.

[15] Kostelnik P., Samulka M., and Janosik M. Scalable multi-robot formations using local sensing and communication. In RoMoCo '02. Proceedings of the Third International Workshop on Robot Motion and Control, 2002.

[16] Tangamchit P., Dolan J.M., and Kosla P.K. The necessity of average rewards in cooperative multirobot learning. In IEEE International Conference on Robotics and Automation. ICRA '02., pages (2)1296– 1301, 2002.

[17] Fierro R. and Das A.K. A modular architecture for formation control. In RoMoCo '02. Proceedings of the Third International Workshop on Robot Motion and Control, 2002.

[18] Camazine S., J.-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. Self-Organisation in Biological Systems. Princeton University Press, NJ, 2001.