



ISSN: 2321-2152

IJMECE

*International Journal of modern
electronics and communication engineering*

E-Mail

editor.ijmece@gmail.com

editor@ijmece.com

www.ijmece.com

<https://zenodo.org/records/14505661>

Pattern Formation and Adaptation in Multi-Robot Systems: A Review

Vijayapuram Keerthi¹, Assistant Professor¹, Department of ECE, Siddhartha Institute of Technology & Sciences, Telangana, India

Sunkara Satya Krishna², Assistant Professor², Department of CSE, Siddhartha Institute of Technology & Sciences, Telangana, India.

Abstract

Recent advances in robotics have started making it feasible to deploy large numbers of inexpensive robots for tasks such as surveillance and search. However, coordination of multiple robots to accomplish such tasks remains a challenging problem. This report reviews some of the recent literature in multi-robot systems. It consists of two parts. In the first part, we reviewed the studies on the pattern formation problem, that is how can a group of robots be controlled to get into and maintain a formation. The second part reviews the studies that used adaptation strategies in controlling multi-robot systems. 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.

Introduction

Recent advances in robotics have started making it feasible to deploy large numbers of inexpensive robots for tasks such as surveillance and search. However, coordination of multiple robots to accomplish such tasks remains a challenging problem. Previous reviews on multi-robot systems (such as those written by Cao et al.[25] and Dudek et al.[7]) have taken a broad view. Different from these, this report has a narrow span and limits itself to the recent literature on pattern formation and adaptation in multi-robot systems. The report consists of two parts. In the first part, we reviewed the studies on the pattern formation problem, that is how can a group of robots be controlled to get into and maintain a formation. The second part reviews the studies that used adaptation strategies in controlling multi-robot systems. 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.

Pattern formation in multi-robot systems

The pattern formation problem is defined as the coordination of a group of robots to get into and maintain a formation with a certain shape, such as a wedge or a chain. Current application areas of pattern formation include search and rescue operations, landmine removal, remote terrain and space exploration, control of arrays of satellites and unmanned aerial vehicles (UAVs). Pattern formation is also observed in various animal species as a result of cooperative behaviors among its members, where the individuals stay at a specific orientation and distance with respect to each other while moving, or fill a specific area as homogeneously as possible. Examples of pattern formation in animals include bird flocking, fish schooling, and ants forming chains[18].

We have classified the pattern formation studies into two groups. The first group includes studies where the coordination is done by a centralized unit that can oversee the whole group and command the individual robots accordingly. The second group contains distributed pattern formation methods for achieving the coordination.

Centralized pattern formation

In centralized pattern formation methods, a computational unit oversees the whole group and plans the motion of the group members accordingly[3, 13, 23, 24]. The motion of each robot is then transmitted to the robot via a communication channel. Egerstedt and Hu[13] propose a coordination strategy for moving a group of robots in a desired formation over a given path. Path planning is separated from the path tracking task. It is done in a centralized way and the tracking of virtual reference points are handled separately. The path for a virtual

<https://zenodo.org/records/14505661>

leader is computed as a reference point for the robots to follow. They applied the method to coordinate the movement of simulated robots in a triangular formation while avoiding an obstacle. In this example, the robots that formed the corners of the triangle, went around an obstacle, which fell in between the robots. The paper proves that, if the tracking errors of the robots are bounded or tracking is done perfectly, then the described method stabilizes the formation error. Koo and Shahrz [23] propose a centralized path-planning method to fly a group of unmanned aerial vehicles (UAVs) in a desired formation. The path of each UAV is computed by a leader UAV, which is more capable than others. Only the leader

has cameras and sensors. It tells the other UAVs, via a communication channel, what trajectories they should track. What UAVs should do is to take off and fly toward their trajectories and lock onto them. Two cases are considered in experiments: the case where UAVs take off one by one, and where they do it simultaneously. Trajectory computation is the main focus of this study. Belta and Kumar [3] propose a centralized trajectory computation scheme that uses kinetic energy shaping. Instead of using a constant kinetic energy metric, they employ a smoothly changing the kinetic energy metric. The method generates smooth trajectories for a set of mobile robots. The proximity between the robots can be controlled via a parameter. However the method does not take obstacle avoidance into consideration and that is not scalable. A target assignment strategy for formation building problem is described by Kowalczyk [24]. Starting with a scattered group of robots, the algorithm first assigns a target point for each robot in the desired final formation. Then it generates necessary priorities and trajectories for the robots to avoid collisions while moving to their target points. Each robot has an area around its path that is forbidden to other robots with lower priorities. If the robot's trajectory crosses a forbidden area of a higher priority robot, the robot waits until the higher priority robot moves out of its way. The method is tested with non-holonomic and holonomic robots. The method assumes the existence of a global sensing ability and a centralized computation. The scalability of the method is not addressed.

Centralized pattern formation strategies rely on a central unit that oversees the whole group and assume the existence of a communication channel between the central unit and the individual robots. Such assumptions make the centralized strategy more costly, less robust to failures, and less scalable to the

control of large number of robots. An alternative is to use decentralized pattern formation strategies

Decentralized pattern formation

Communication and completeness of information known by robots impose a trade-off between precision and feasibility of forming and maintaining the pattern and the necessity of global information and communication. Studies that require global information or broadcast communication [29, 19, 12] may suffer from lack of scalability or high costs of the physical setup but allow more accurate forming of a greater range of formations.

On the other hand, studies using only local communication and sensor data [21, 22, 10, 5, 17, 15, 9, 11] tend to be more scalable, more robust, and easier to build; but they are also limited in variety and precision of formations. Sugihara and Suzuki [12] achieved pattern formation by providing each robot the global positions of all others. In this study, an algorithm is developed for each pattern. The proposed method can uniformly distribute robots creating different pattern formations (circles, polygons, line, filled circle, and filled polygon). It can also split a group of robots into an arbitrary number of nearly equal sized groups. Despite the impressive results obtained by this decentralized algorithm, the global communication required to share information among the whole group, makes it less scalable. Carpin and Parker [19] introduced a cooperative leader following strategy for a team of robots. The robots are able to maintain a specific formation while simultaneously moving in a linear pattern and avoiding dynamic obstacles. The robots use local sensor information and explicit broadcast communication among themselves. The framework handles heterogeneous teams, i.e. comprising of robots with different types of sensors, as well as homogeneous ones.

Two levels of behaviors were implemented for tasks: team-level and robot-level behaviors. Transitions are made when necessary among specific behaviors in these two levels. For example, when a member of the team faces an obstacle, the whole team waits together with that member for it to go away for a certain amount of time. If this time is exceeded that member circumnavigates the obstacle and the team returns to its main task of moving in a formation. Balch and Hybinette [21, 22] proposed a different strategy for robot formation that is inspired from the way molecules form crystals. In this study, each robot has

<https://zenodo.org/records/14505661>

several local *attachment sites* that other robots may be attracted to. This concept is similar to molecular covalent bonding. Possible attachment site geometries include shapes resembling where the robot is the center of the shape and the attachment sites are the ends of the line segments. Various robot formation shapes result from usage of different attachment site geometries just as different crystal shapes emerge from various covalent bond geometries. When a team of robots moving in a formation, they avoid the obstacle by splitting around it and rejoining after passing. This approach is scalable to large robot teams since global communication is not used and that local sensing is sufficient to generate effective formation behaviors in large robot teams.

Another method similar to crystal generation which employs a form of probabilistic control is proposed by Fujibayashi et al. [11]. This study makes use of virtual springs to keep two agents in close proximity. Each pair of robots within a certain range of each other, are connected via a virtual spring. Each agent is classified by the number of neighboring agents within this range (number of connections). The robots form triangle lattices that have random outlines. To obtain a desired outline, the virtual springs among some robots are broken with a certain probability. The candidate springs to be broken are chosen depending on the number of connections the robots it join have. This *breaking preference* and the probability of breaking changes from formation to formation. The algorithm uses only local information and is decentralized. One disadvantage of the method is the difficulty of choosing custom parameters for each formation.

A graph-theoretic framework is proposed by Desai [10] for the control of a team of robots moving in an area with obstacles while maintaining a specific formation. The method uses control graphs to define behaviors of robots in the formation. This framework can handle transitions between formations, i.e. between control graphs. Proofs of the mathematical results required to enumerate and classify control graphs are given. Although the computations for control graphs increase with the number of robots, the fact that these computations are decentralized allows the methods described to be scalable to large groups. Another graph-based approach to *moving in formation* problem is introduced by Fierro and Das [17]. They proposed a four-layer modular architecture for formation control. Group control layer is the highest layer generating a desired trajectory for the whole group to move. Formation control layer

implements a physical network, a communication network, and a computational network (control graph). It maintains the formation by using local communication and relative position information. Kinematics control layer deals with the required linear and angular velocities of robots. Finally, the dynamic control layer handles the task of realizing the necessary speeds given by the kinematics control layer. This four-layer architecture provides an abstraction among tasks required at different levels. For example, a robot with different mass, inertia, and friction can be used only by changing the dynamic control layer. Furthermore a modular adaptive controller is described which can manage control of robots with unknown dynamics and learns the robot dynamics on-the-field. Hence using a different robot requires no change in the system. The method described is scalable (control algorithms scale linearly) and flexible (it allows various formations). Centralized and decentralized versions of control graph assignment algorithm is also described in the study.

only local communication and sensor information. Obstacle avoidance is also provided in this method. It extends ordinary behavior-based approaches with the application of social roles that represent positions in the formation and with the use of local communication to improve performance. As new agents join the formation, the shape is *fixed* by local communications and role changes where necessary. The locally communicated information reaches the leader, i.e. the front most robot, which knows the whole shape of the current formation and which decides on the changes necessary. This information is then propagated to the necessary followers, and the formation is updated. There is no need to predefine social roles or positions for robots. Everything is done dynamically as the formation grows. This method supports various formations and also switching between them, therefore it is flexible as well as being scalable and local. Dudenhoeffer and Jones [5] designed and implemented a tool to model and simulate collective behavior and interactions of a group of thousands of robots. Using this simulation tool, the problem of hazardous material detection by thousands of micro-robots scattered around a region is tackled. Social potential fields are utilized for coordinated group behavior where robots are desired to stay at a specific distance from others to obtain optimum coverage of the area. They are also required to wander in this formation to search other parts. The desired behavior is obtained by using a subsumption architecture. This study also validates the proposed method in cases where it is possible for agents to die and where agents

<https://zenodo.org/records/14505661>

have imperfect sensor readings. The method uses only local information and is scalable to very large groups of robots. Mataric and Fredslund [9] used local information to establish and maintain formations among robots. Each robot has a unique ID and a designated friend robot which it can see through a friend sensor. There is also minimal communication between robots: heartbeat signals (robots broadcast their IDs), swerve signals (changing direction), and formation messages. Each robot can learn the number of robots in formation and the type of formation using broadcasted messages. For each formation, each robot has a specified angle which determines the angle it should keep between its front direction and the direction of its friend. This angle is calculated locally. The details of this calculation are given in [9]. This study accomplishes the task of establishing and maintaining formations using only local information and minimal communication.

However the possible formations are limited to chain-shaped ones that do not make a backward curve. One of the major reasons why multi-robot systems are preferred over single-robot systems is their robustness in performance. The robustness of multi-robot systems can be improved by incorporating adaptation mechanisms that can respond to continuing changes in the environment as well as in the capabilities of individual robots.

Adaptation in multi-robot systems

In this section we review the studies that used adaptation strategies in controlling multi-robot systems. 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 as 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

Individual level adaptation

Reinforcement learning models become useless when the state space is too large. Using multiple learning modules for different states instead of a single complicated learning module is one approach to solve this problem. Takayashi's work [26] is one such study. The problem studied in his work is a reduced

version of robo-soccer challenge. Opponents are assumed to have different modes of operation each with a different policy. Modules consist of predictors and planners. Predictor predicts the next action of opponent based on its previous behavior. Planner on the other hand generates optimal move based on this prediction. Predictors compete for better accuracy and only best predicting module is reinforced. This creates specialized modules for different modes of operation of the opponent. The problem used in this work is ball chasing in presence of a random moving opponent. The results show improvement over single module learning. Reinforcement learning converges to optimal policy given sufficient trials but it is often the case that these sufficient trials are too large to be feasible. Piao [20] proposes an improved reinforcement learning method to improve learning speed of learning. This method is a combination of rule learning, reinforcement learning and *action level selection* which is basically behavior rules for specific states. The rule base consists of instances that are states passed through a fixed interval. These instances are labeled after each epoch using information gathered through the epoch. These instances are then combined to create rules. These rules are used as a prohibitive guide to inhibit useless or harmful actions. *Action level selection* is composed of hard coded rules to govern general strategy of robots. *Action level* is also fed into reinforcement level together with sensor data to generate the state information. Finally reinforcement learning module uses sensory information and *action level* to generate state and learns to generate actions. Piao applies this method to the robo-soccer problem. He assumes only one agent is learning at a given time and reports improved performance on learning with multiple robots over standard Q learning.

Reinforcement learning is intended for single entities, therefore it doesn't have any mechanisms to support cooperative behaviors. Tangamchit's work [16] tackles this problem. This work addresses the distinction between action level and task level systems. To solve problems, action level systems generate reactive behaviors. On the other hand task level systems generate tasks composed of subtasks possibly distributed over multiple agents. Tangamchit defines cooperation as a task level activity, where robots can share resources and duties. Two different schemes of reward are considered: global and local. In the global reward scheme, the reinforcement received by a unit is distributed to the whole group. In contrast, in the local reward scheme the reward is not distributed among the members of the group. Two learning algorithms are considered: Q-learning and Monte

<https://zenodo.org/records/14505661>

Carlo learning. Q-learning uses cumulative discounted rewards whereas Monte Carlo learning uses averaging to assess the value of each action in each state. Reward is same for each state action pair in an episode. This scheme is slower since it disregards the importance of latter actions in episode which are usually more effective in obtaining reward.

The case examined for this study is puck collecting behavior which is a subclass of foraging problem. Robots are required to collect pucks and to deposit them into the bin. Each action has a negative reward except the action of depositing a puck. The field consists of a home region, which doesn't contain any pucks, a deposit bin, and pucks distributed around the region. Two heterogeneous robots are used for this task. The first robot moves and collects better in the region outside the home region. The second robot is limited in movement to home region but can accomplish bin deposition more efficiently. Optimal strategy requires robots to cooperate and first to bring pucks into home region and second to deposit them. This requires task level learning.

Results indicate that task-level cooperation can't be learned well using local rewards or discounted cumulative rewards as in Q learning. In opposition global rewards coupled with average rewards result in cooperative policies for this task. Reinforcement learning only requires feedback for applied sequence of actions to incorporate domain knowledge. This is usually incorporated by choice of reward functions. Mataric [14] discusses reward functions in a foraging task. Although single goals are mathematically simple to analyze, they cause problems through acquisition of behavior.

Especially contingent and sequential behaviors are hard to convert into monolithic goal functions. Instead of this, separate goal functions are used, each describing a subgoal of agent. A second improvement is progress estimators. These estimators give a rough idea of how well a specific goal is going on. These two improvements greatly increase the usage of domain knowledge in the topic (by designing appropriate subgoals and estimating progress of the subgoals). They also give much more reinforcement than standard methods, since not only the final goal but also intermediate steps are reinforced. This improved method is tested on real robots working on a foraging task. Robots are to collect pucks and to deliver them to home. Robots are also responsible to be present at home at certain intervals. Robots are given some simple reactive behaviors to reduce state space of learning

problem to a manageable size. These behaviors are collecting pucks when it is immediately before agent, avoiding obstacles and dropping pucks when at home. Experiments are compared with optimal policy generated by hand. Results indicate the benefit of both improvements purposed. An interesting note in this paper is the interference caused by agents. Increasing number of learning agents has detrimental effect on general learning speed and convergence. Parker's [6] L-ALLIANCE model uses multiple behavior sets and global communication to achieve cooperation. Each behavior set has a monitor. These monitors check required conditions for activation of behavior sets, also assess the capability of agent and other agents. Parker introduces two motivations: impatience and acquiescence. Impatience corresponds to tendency to take a task being done by other robots and acquiescence describes tendency to give up a task to be performed by another robot. L-ALLIANCE architecture changes these motivational parameters during learning. The architecture requires robots to broadcast current actions to other robots. This architecture assumes that when a robot declares an action, the changes in environment that can be caused by result of that action are attributed to that robot. This handles credit assignment problem. L-ALLIANCE architecture can handle heterogeneous groups and can adapt to failures or changes in robot abilities which are desired properties. On the other hand, L-ALLIANCE requires global communication and makes a strong assumption to solve credit assignment problem.

Goldberg et al. [4] propose *Augmented Markov Models* (AMM). AMM is a Markov model improved by additional statistics about transitions. It is designed to learn the statistics of the environment rather than to generate a policy. AMM's assume action being performed can be known perfectly, so it is differentiated from Hidden Markov Models. AMM's are first order Markov models but they are built incrementally. This incremental building gives them ability to better approximate such higher order transitions in the system. Their work combines AMM's with behavior based robotics [2]. Each behavior is monitored using AMM's with different time scales. This allows system to respond both slow and fast changes in the environment.

Group level adaptation

Reinforcement learning is by definition centralized which is inefficient to implement in multi-robot systems. Yanli's study [27] on opportunistically

<https://zenodo.org/records/14505661>

cooperative neural learning proposes a trade-off for centralized versus decentralized learning debate. In pure decentralized learning models each agent keeps its learning experience hidden from other agents. This seriously affects performance of the group since the experience can not be shared. Yanli solves this problem by adding 'opportunistic' search. This strategy is similar to survival of fittest concept in genetic algorithms. Less t networks copy highly t networks to improve their performance.

Yanli reports the comparison of three cases, central, distributed and opportunistically distributed. These cases are tested on searching task where agents are required to cover as much of a given space as possible avoiding multiple passes as much as possible. The best strategy clearly is one that utilizes cooperation. All agents act simultaneously and plan their movements ahead of action. Agents also share their plans with other agents. These plans are used to predict the next action of all other agents by each agent. Learning takes place in these predictors. When the next action of other agents can be predicted precisely reward can be calculated.

Results show that central learning is superior to all this methods in performance. However central learning has many problems in fault-tolerance and communication. OCL (opportunistically cooperative learning) performs almost as well as central learning and both perform remarkably better than distributed only case.

Agah[1] combines both individual and group adaptation in his work. Agah uses so called *Tropism Architecture* to approach multi robot learning problem. Tropism architecture serves as a learning module between senses and actions. Each tropism is defined as a tendency to elicit a response for a given stimuli. Tropism architecture keeps a list of learned tropisms (i.e. state, action, tendency pairs). Agents make decisions based on matching tropisms to current state. A stochastic process is used to determine which actions to apply biased on the tropism values.

Both kinds of learning are applied using this architecture. In individual learning scheme, the list of tropisms are updated based on feedback obtained from environment. These updates include adding a new valid action for current state, increasing tropism value for a pair which has been positively reinforced and changing action when an invalid or negatively reinforced action is encountered. In population learning, the tropism lists for each agent is converted into variable length bit strings. Using these

bit strings, a genetic algorithm is run. The fitness of each individual is calculated based on the rewards it received during individual learning. Results indicate success of this twofold method even in absence of reinforcement propagation as in Q-learning.

It is not always possible to have behaviors beforehand and even behaviors should be learned in certain cases. Hexapod locomotion is such a case. Parker[8] studies on learning a cooperative box pushing task in hexapod robots. The main problem he is facing is the locomotion problem, since moving hexapod robot requires more complicated operations than wheeled robots. For this task Parker purposed Cyclic Genetic Algorithms (CGA), which handles requirements of such complicated control. The motivation behind CGA's is evolving a sequence of operations instead of simple stimulus-response pairs. CGA encodes a series of activations which are to be repeated by the agent. Fitness of each chromosome for a given task is calculated by using a computer simulation where the chromosome to be evaluated is paired with the best known solution to the problem. The success of the group is used as the fitness measure for the chromosome. Results indicate the effectiveness of purposed method.

Cooperation requires coordination among robots, which requires communication. Early approaches to cooperation used peer-to-peer communication models. This, although possibly required for optimal solution, requires increasing computational power and bandwidth for increasing number of robots in the system. Local communication reduces bottlenecks in communication but not totally solves this problem. Stigmergy, that is communication through environment, is a possible solution to communication bottleneck. This implicit communication scheme allows scalability and is observed in social insects.

Yamada's[28] work provides a working implementation of an implicit communication system for cooperation in robot groups. This scheme is applied to the box pushing problem. Goal is identified with a light source and robots are assumed to be capable of the following: detecting whether box being pushed is moving or not, presence of other robots and presence of walls. Here walls are modeled as unmovable boxes so they are ignored in the end. The authors generate situations to solve implicit communication problem. Situations abstract models of state of the world, which are computed using the sensor data and some very crude memory (such as counters for some sensor readings). Robots have sets

<https://zenodo.org/records/14505661>

of rules for each situation. These rules are applied according to sensor readings.

Conclusion

We reviewed the recent studies on the pattern formation and adaptation in multi-robot systems. The pattern formation studies are classified into two groups. The first group includes studies where the coordination is done by a centralized unit that can oversee the whole group and command the individual robots accordingly. The second group contains distributed pattern formation methods for achieving the coordination. The studies that used adaptation strategies in controlling multi-robot systems were classified into two levels: group level and individual level.

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.
- [19] Carpin S. and Parker L.E. Cooperative leader following in a distributed multi-robot system. In *Proceedings. ICRA '02. IEEE International Conference on Robotics and Automation*, 2002.
- [20] Piao S and Hong B. Fast reinforcement learning approach to cooperative behavior acquisition in multi-agent system. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 871. 875, 2002. vol.1.
- [21] Balch T. and Hybinette M. Behavior-based coordination of large-scale robot formations. In *Proceedings. Fourth International Conference on MultiAgent Systems*, 2000.
- [22] Balch T. and Hybinette M. Social potentials for scalable multi-robot formations. In *Proceedings. ICRA '00. IEEE International Conference on Robotics and Automation*, 2000.
- [23] Koo T.J. and Shahruz S.M. Formation of a group of unmanned aerial vehicles (uavs). In *Proceedings of the American Control Conference*, 2001.
- [24] Kowalczyk W. Target assignment strategy for scattered robots building formation. In *Ro- MoCo '02. Proceedings of the Third International Workshop on Robot Motion and Control*, 2002.
- [25] Cao Y., Fukunaga A. S., and Kahng A. B. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7.23, 1997.
- [26] Takahashi Y., Edazawa K., and M. Asada. Multi-module learning system for behavior acquisition in multi-agent environment. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 927. 931, 2002. vol.1.
- [27] Yanli Y., Polycarpou M.M., and Minai A.A. Opportunistically cooperative neural learning in mobile agents. In *Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN*, pages 2638.2643, 2002.
- [28] S. Yamada and Saito J. Adaptive action selection without explicit communication for multirobot box-pushing. In *Systems, Man and Cybernetics, Part C, IEEE Transactions on, Volume: 31, Issue: 3*, pages 398.404, 2001.
- [29] Cao Z., Tan M., Wang S., Fan Y., and Zhang B. The optimization research of formation control for multiple mobile robots. In *Proceedings of the 4th World Congress on Intelligent Control and Automation*, 2002.