

Holonic Formation Control for Mobile Robots *

Harry Chia-Hung Hsu
Compal Communications, Inc.
Email: harry_hsu@compal.com

Alan Liu, Yen-Ru Cheng
Dept. of Electrical Engineering,
National Chung Cheng University
Email: aliu@ee.ccu.edu.tw,
d95415007@ccu.edu.tw

Abstract

Within the research field of formation control, multiple robots are generally controlled in only one team. Not much research focuses on formation control with multiple teams, and even less research focuses on combining several formation teams into a larger one. To solve the problem of a formation consisting of several sub-formations, we use the concept of holon to propose an approach called holonic formation control. For such control, we describe a method to present the shapes of holonic formations and a method for these holonic formations to maintain formations. In addition, we design the architecture of the robot holon to control each mobile robot. The simulation results demonstrate that mobile robots can maintain holonic formations well and can perform multi-team holonic formations in a pursuit-evasion game.

1 Introduction

Formation control is a research topic within the research fields of multirobot systems [1]. From the Arkin's description [2, p.361], the term formation means that a robot team forms a *specific* geometric pattern (or called formation shape) and maintains it while moving. Researchers have proposed various kinds of methodologies to control formations. Behavior-based control is used for controlling robots to maintain formations, such as line, column, wedge, etc [3]. The main control concept in [4] is to let each robot keep a single friend (a specific neighboring robot) at a desired angle. Robots plan their trajectories according to the reference trajectory of a common reference point [5].

Among these research work in formation, they generally control multiple robots in only one team. Not much research focuses on formation control with multiple teams, but we can still see the work by [6], in which it controls multiple teams through an adjacency matrix. In [7], the robots

use the algorithm FOLLOW to divide themselves into several groups. In [8], a multiagent system called *Virtual Operator MultiAgent System* (VOMAS) is used for controlling multi-team formations with the advantages of multiple teams, multiple tasks, scalability, and flexible task switch.

Other than the work done above, in some situations, the relation between/among formation teams is more complex than only multiple independent teams. Some formation teams may cooperate with each other to form a larger formation. For example, a platoon in the military is generally divided into several squads (described in detail in Section 2). To solve such kind of formation problems, in which a formation is formed by several sub-formations, we use the concept of *holon* to describe this concept [9]. A holon is an autonomous and cooperative entity/system, and "a holon can be part of another holon" [9]. According to this concept, when we view a mobile robot as a holon, a formation formed by several mobile robots can also be viewed as a holon. In addition, a larger formation formed by several sub-formations can also be viewed as a holon. This process can be repeated again and again, so that we can extend a formation to a larger scale. We call such kind of formation a *holonic formation* and call its control *holonic formation control*.

Two important tasks in formation control are formation shape representation and robotic control. For the shape representation, we propose a method to represent the shape of a holonic formation, and we use the concept of desired vector [8] to design the robotic control for each robot to maintain formation. In addition, we design the architecture of the *robot holon* to control each mobile robot, where this architecture is derived from the VOMAS architecture [8]. According to this architecture, each robot is controlled by a pair of a virtual operator and a robot agent, and a robot has the capabilities of high level reasoning and low level reactive control. In addition, dynamic team control can be also achieved based on this control architecture.

*This research was partly sponsored by Ministry of Economic Affairs under the grant 95-EC-17-A-02-S1-029 and National Science Council under the grants 95-2752-E-008-002-PAE and 95-2221-E-194-009.

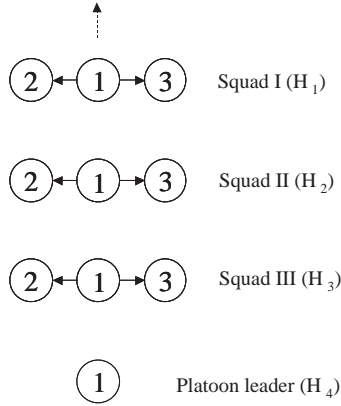


Figure 1: A line-column formation.

2 Holon and Holonic Formation

Before describing the concept of holon and holonic formations, we illustrate an example of a formation consisting of several sub-formations with Figure 1. When a platoon consisting of ten infantrymen is in a combat situation and the enemy is in front of this platoon, these infantrymen may be divided into four groups with three squads and a platoon leader. As shown in Figure 1, each squad consists of three infantrymen in a line formation, and the four groups form a column formation, so that we call this formation a line-column formation, which means several line formations form a larger column formation. From the figure, we can see three squads, marked as squad I, II, and III, and the platoon leader. Each circle designates an infantryman (or a mobile robot), and Arabic numerals and H_1 to H_4 denote identifications of these infantrymen, which will be described in detail in Section 3. The dashed arrow is the progress direction of the platoon, and the solid arrow means that the infantryman in the side with an arrowhead will follow and maintain formation with the infantryman in the line side.

In the following, we will present the concept of holon first and then describe the concept of holonic formations.

2.1 Holon

The term ‘holon’ was first mentioned by Koestler in [10]. The word comes from the Greek *holos*, which signifies whole, with the suffix *on*, which means a particle or part as in *proton* or *neutron*. A holon has two-face meaning: wholes and parts, and it behaves partly as wholes or wholly as parts according to what view we look at it. A holonic robot system views a robot as a holon and lets multiple robots build an HMS (holonic manufacturing system) with implementation in an assembly system [11]. We use the concept of viewing a robot as a holon in the formation con-

trol to propose the holonic formation control in this paper.

2.2 Holonic Formation

We can treat each robot as a holon [11]. According to the concept of “a holon can be part of another holon” [9], a formation can also be viewed as a holon. Similarly, several formations with the same or different shapes can form another larger formation. According to the same concept, the larger formation can also be viewed as a holon. This process can be repeated to extend the number of robots to a larger scale with regular formation structure. We call such kind of formation *holonic formation* and call its control *holonic formation control*.

To describe the concept of holonic formation in detail, we illustrate the concept with the example in Figure 1. In this figure, each circle indicates a mobile robot (an infantryman), and each mobile robot can be viewed as a holon. From the squad I to the squad III, each squad consists of three mobile robots and forms a line formation. The platoon leader also forms a line formation, although the line formation consists of only one mobile robot. These three squads and the platoon leader can also be viewed as holons. A larger column formation is formed by the four line formations, and similarly the line-column formation is also viewed as a holon.

3 Holonic Formation Control

There are two important tasks in holonic formation control, which are to present the holonic formation shapes and to maintain the shapes via control of each mobile robot or via control of sub-formations. We then use the formation representation in [8] as the basis to represent the shapes of holonic formations. In addition, we describe the robotic control for mobile robots and for sub-formations to maintain formation.

3.1 Holonic Formation Representation

In holonic formation representation, each holon can be viewed as a *vertex*, and the relation between two holons as an *edge*. Each holon is given an identification, called *Holon ID (HID)*, and HID determines a holon’s position in a formation. The HID of the *leader* holon in a formation is defined as 1, and the rest of the holons with HIDs of natural numbers are called the *follower* holons. We use the leader-following approach to maintain formation.

A mobile robot can be located by using an *HID sequence*, which consists of several HID items and each HID item determines the position of a holon in its corresponding *formation level*. For example, the HID sequence of the platoon leader in Figure 1 is $\langle 1, 4 \rangle$, where 1 means that the HID

of the robot is 1 (shown as the Arabic numeral inside the circle of the platoon leader) in the formation level of the line formation, and 4 means that the HID of the line formation is 4 (shown as H_4) in the formation level of the line-column formation.

A *successor* holon (a follower holon) will maintain formation with its *predecessor* holon (the leader or local leaders) according to their HIDs and their corresponding edge relations. The form of edge relation is described as follows:

$$E(c, H_{suc}, H_{pre}, l^d, \phi^d), \quad (1)$$

which means when the *condition* c is true, a *successor* holon H_{suc} (may be a robot or a sub-formation) will follow its *predecessor* holon H_{pre} with a desired distance of l^d and a desired angle of ϕ^d .

A formation graph of a holonic formation is formed by a set of edges with restrictions on holons. If all holons in a formation are restricted on mobile robots, we call such a formation as a *simple formation*, which is the simplest form of a holonic formation. For example, the line formation in Figure 1 can be expressed as the following formula (marked as F_L):

$$F_L = \{E(i = 2, H_2, H_1, 50, 90^\circ), \\ E(i > 2, H_i, H_{i-2}, 50, 90^\circ + 180^\circ(i\%2)) \quad (2) \\ |H_i \in \mathbf{R}, i \leq 3\},$$

where the restriction is written after the symbol '|', and the restriction of $H_i \in \mathbf{R}$ means that all holons are mobile robots. In this formula, the limitation of $i \leq 3$ is used for limiting the number of mobile robots in a line formation (a squad) to 3 mobile robots.

The line-column formation in Figure 1 can be expressed as

$$F_{LC} = \{E(i > 1, H_i, H_{i-1}, l^d, 180^\circ) \\ |H_i = F_L\}, \quad (3)$$

where the restriction of $H_i = F_L$ means that each holon H_i is a line formation of F_L in Eq. (2). We call such a formation, which consists of sub-formations, a *compound formation*.

3.2 Holonic Formation Maintenance

A maintained formation means that all members of the formation are in their correct positions according to its formation shape. Before presenting the control for holonic formations, we firstly present the robotic control for simple formations to maintain formation. In [8], we use the concept of *desired position* and *desired vector* to let a successor robot follow its predecessor robot. Considering a robot, R_i , and its predecessor robot, R_j , if we know the position of the

predecessor robot, we can obtain the desired position of the successor robot according to the formation shape or, exactly speaking, the edge relation, such as Eq. (1). Assume that the distance between the actual position and the desired position of R_i is represented as vector $\Delta\mathbf{l}$. The desired vector (or desired velocity) of R_i , \mathbf{v}_i^d , is obtained as

$$\mathbf{v}_{\Delta\mathbf{l}} = \alpha_1 \Delta\mathbf{l} \quad (4)$$

$$\mathbf{v}_i^d = \mathbf{v}_j + \mathbf{v}_{\Delta\mathbf{l}}, \quad (5)$$

where α_1 is a positive constant and \mathbf{v}_j is the velocity of R_j . Two partial vectors of the vector \mathbf{v}_i^d are \mathbf{v}_j and $\mathbf{v}_{\Delta\mathbf{l}}$, where the partial vector \mathbf{v}_j can let the successor robot follow the predecessor robot and the partial vector $\mathbf{v}_{\Delta\mathbf{l}}$ can let the successor robot approach its desired position to maintain formation. We can modify the value of α_1 to adjust the approaching speed. The work of [8] demonstrates by theorems and by simulation results that the control of Eq. (5) makes a successor robot maintain formation with its predecessor robot.

We can extend the concept of desired position and desired vector to the holonic formation maintenance. A successor holon can obtain the relation between it and its predecessor holon from the edge relation such as Eq. (1), and then obtains its desired position by calculation if the successor holon can detect the position of its predecessor holon. The successor holon may be a mobile robot or a sub-formation. If the successor holon is a mobile robot, the successor holon can use the method of Eq. (5) to approach its desired position and to maintain formation. If the successor holon is a sub-formation, the leader holon of the sub-formation is responsible for moving to the desired position of the holon. Because we use the leader-following approach to maintain formation, when the leader holon moves, other follower holons in the sub-formation will also follow the leader holon. Hence, the whole sub-formation will move to its desired position. Since both a mobile robot and a sub-formation will move to their desired positions, the holonic formation is maintained.

4 Robot Holon

Intelligent agents are suitable for implementing the concept of a holon, because first, an intelligent agent is a program with similar attributes to a holon, such as autonomy and cooperation [12]. Second, several agents can cooperate to form a multiagent system (MAS), and several MASs can cooperate to form a larger MAS; this attribute is similar to "a holon can be part of another holon." Many researchers use the concept of agent to implement an HMS [13]. In this paper, we adapt the agent architecture VOMAS [8] to be the architecture of the *robot holon* shown in Figure 2 in order to control each mobile robot to maintain holonic formations.

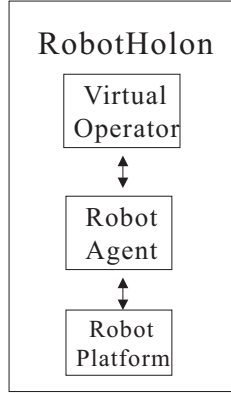


Figure 2: The architecture of a robot holon.

Each mobile robot is controlled by a robot holon, and two main agents in it are the *virtual operator* agent and the *robot agent*, where the virtual operator deals with high level task planning and the robot agent handles low level reactive control.

4.1 Robot Agent

We use motor-schema (a kind of behavior-based control) [2] to design the robot agent, so that the robot agent can control the mobile robot reactively. The behaviors of the robot agent include

- Obstacle Avoidance: Robots can avoid bumping into obstacles.
- Robot Avoidance: Robots can avoid bumping into other mobile robots.
- Goal Seeking: Robots can move to its goal. This behavior is designed for the leader robot to lead the whole formation to a specific goal.
- Predecessor Following: A successor robot can maintain the relation of an edge, such as Eq. (1), to follow and maintain formation with its predecessor robot by using the method described in Section 3.2.

The reader can refer to [8] for detailed design of behavior-based control of the robot agent.

4.2 Virtual Operator

The robot agent can handle reactive control of a mobile robot; in contrast, the virtual operator deals with which robot the predecessor is and the relation between this robot and its predecessor. To implement it, a virtual operator requires the knowledge of an HID sequence and a formation graph, such as those in Eqs. (2) and (3). According to the

knowledge, a virtual operator can know the relation between this holon and other holons, and it can maintain formation with its predecessor holon by using the behavior of “predecessor following” from the robot agent.

Each virtual operator can perform a high level task. The main difference between a virtual operator and another one is their corresponding tasks. We can design several virtual operators with various tasks to perform various roles for mobile robots, such as pursuers and evaders in Section 5.3.

A virtual operator supports four requests for mobile robots to organize multi-team control, and they are the *join*, *remove*, *merge*, and *split* requests. A mobile robot can send a join request via agent communication language (ACL) to a leader robot to join its simple formation. In contrast, the remove request lets a robot remove itself from a simple formation. The merge request can let two simple formations or two compound formations merge into a larger compound formation; in contrast, the split request can split a compound formation into several smaller formations.

5 Implementation

5.1 Simulation Platform

We use Borland C++ Builder (BCB) to build a simulator to simulate mobile robots. We use JADE (Java Agent DEvelopment framework) to build agents, such as the virtual operator and the robot agent. We use Internet to communicate information and commands between agents and the simulator. In addition, we use JESS (Java Expert System Shell) to code planning or tasks for each role (virtual operator) in Section 5.3.

5.2 Results of Holonic Formations

Figure 4 shows three kinds of holonic formations, i.e. the line-column, column-wedge, and column-line formations. The figures in the upper row show the snapshots of robots after executing for a while, and formations in these figures are only roughly maintained. The lower row shows that the formations are well maintained. The line-column formation is the same as the formation shown in Figure 1 and its formation graph is represented in Eq. (3) with l^d of 50 cm. Similarly, the column-wedge and the column-line formations are a wedge and a line formation formed by several column sub-formations. The formation graphs of these two formations are F_{CW} and F_{CL} shown in Figure 3.

5.3 Multi-Team Control

A holonic formation consists of several sub-formations and each sub-formation can be treated as a small team to accomplish a task. In order to test multi-team control for

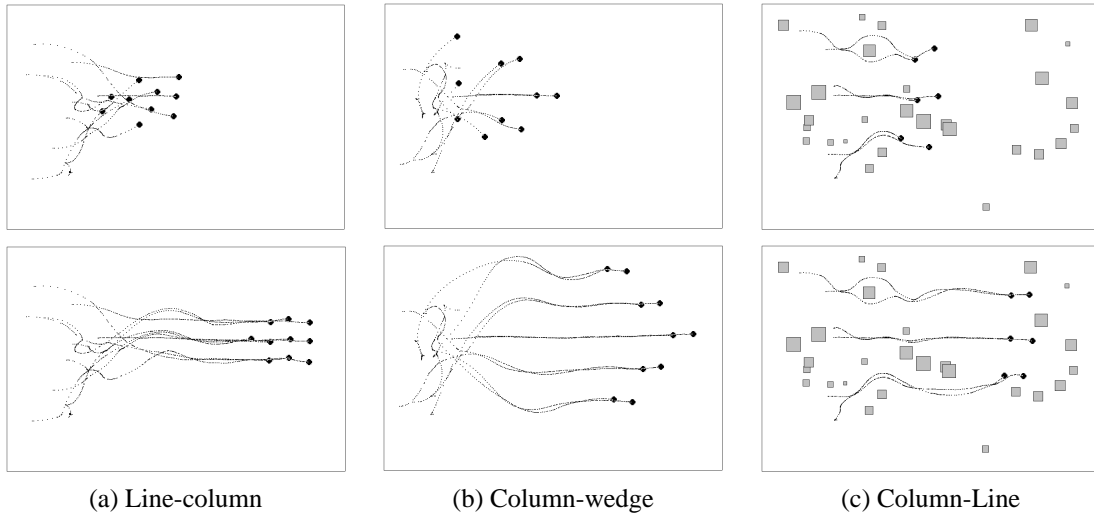


Figure 4: Simulation results of holonic formations.

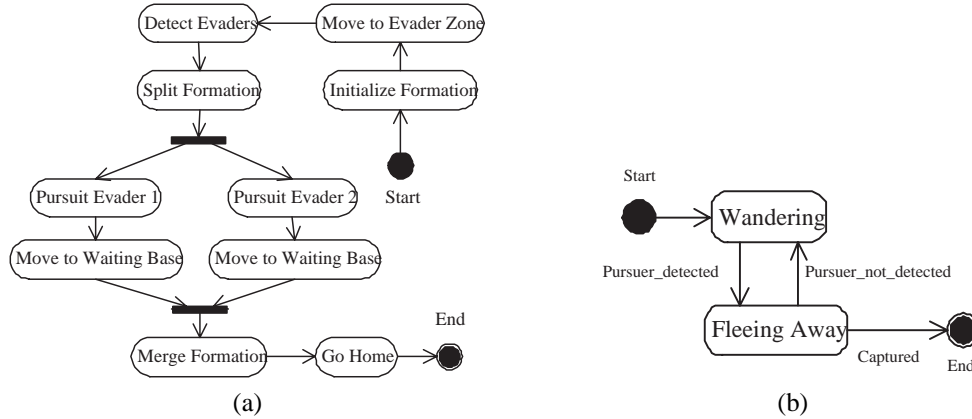


Figure 5: Design of the pursuit-evasion game. (a) Activity diagram of the pursuer teams. (b) State diagram of the evaders.

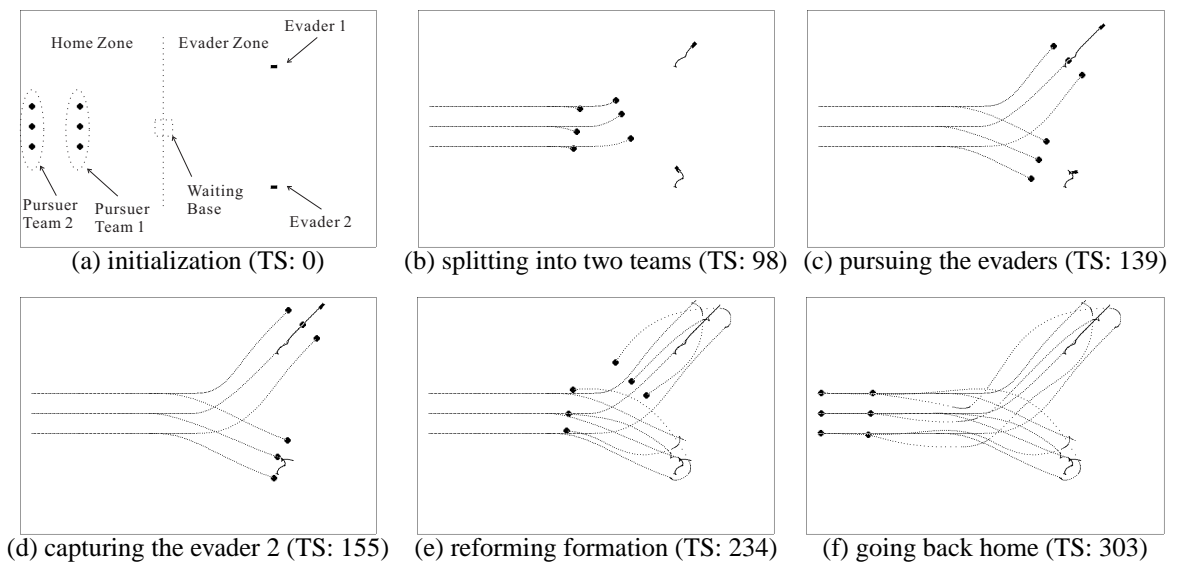


Figure 6: The simulation results of the pursuit-evasion game. TS: time steps.

$$\begin{aligned}
F_C &= \{E(i > 1, H_i, H_{i-1}, 50, 180^\circ) | H_i \in \mathbf{R}, i \leq 2\} \\
F_{CW} &= \{E(i = 2, H_2, H_1, 120, 135^\circ), \\
&\quad E(i > 2, H_i, H_{i-2}, 120, 135^\circ + 90^\circ(i\%2)) | H_i = F_C\} \\
F_{CL} &= \{E(i = 2, H_2, H_1, 120, 90^\circ), \\
&\quad E(i > 2, H_i, H_{i-2}, 120, 90^\circ + 180^\circ(i\%2)) | H_i = F_C\}
\end{aligned}$$

Figure 3: The formulas for the column-wedge and the column-line formations.

holonic formations, we design a pursuit-evasion game to clarify the capability of multi-team control. The scenario of the pursuit-evasion game is described as follows. Six pursuers (unicycle-type robots) form two teams in a holonic formation, and the goal of each team is to pursue a respective evader (car-like robot). In the beginning, the two pursuer teams will move from the home zone toward the evader zone in a line-column formation, and the evaders just wanders. After these pursuers enter the evader zone, they detect their respective evaders and split into two teams to capture their corresponding evaders; meanwhile, the evaders will try to flee away. After they capture their evaders, they then go back to the waiting base to wait the other pursuer team. When the other team arrives, the two pursuer teams merge into and reform a line-column formation and then go back home together. Note that this simulation mainly focuses on feasibility of multi-team control and not on the algorithm of how a pursuer can capture an evader, so that the maximum speed of a pursuer is set three times faster than that of an evader, and a pursuer can easily capture an evader.

From the scenario mentioned above, the pursuit-evasion game requires two kinds of roles (virtual operators): pursuer and evader. In addition, it also requires two kinds of robot agents for the two robot platforms: unicycle-type robots and car-like robots. Based on the requirements, we design two robot holons to control the mobile robots of pursuers and evaders, where the architecture and functions of robot holons have been described in Section 4. As for the high level task of virtual operators, we use Unified Modeling Language (UML) to design the pursuers and evaders, and their activity diagram and state diagram are shown in Figure 5.

For the implementation, we use JADE to implement the functions of virtual operators and robot agents described in Section 4 and use JESS to implement high level tasks of pursuers and evaders as designed in Figure 5. The simulation results are shown in Figure 6.

6 Conclusion

This paper used the concept of holon to solve the problem of holonic formation control, in which a formation consists of several sub-formations. The simulation results val-

idated the feasibility of holonic formations with taking obstacle avoidance into account. In addition, multi-team control is also feasible by the architecture of robot holon, and it is demonstrated by the simulation results of the pursuit-evasion game.

REFERENCES

- [1] T. Arai, E. Pagello, and L. E. Parker, "Guest editorial: Advances in multirobot systems," *IEEE Transaction on Robotics and Automation*, vol. 18, no. 5, pp. 655–661, Oct. 2002.
- [2] R. C. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.
- [3] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, Dec. 1998.
- [4] J. Fredslund and M. J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Transaction on Robotics and Automation*, vol. 18, no. 5, pp. 837–846, Oct. 2002.
- [5] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, pp. 65–78, 2004.
- [6] J. P. Desai, "A graph theoretic approach for modeling mobile robot team formations," *Journal of Robotic Systems*, vol. 19, no. 11, pp. 511–525, Nov. 2002.
- [7] K. Sugihara and I. Suzuki, "Distributed algorithms for formation of geometric patterns with many mobile robots," *Journal of Robotic Systems*, vol. 13, no. 3, pp. 127–139, Mar. 1996.
- [8] H. C.-H. Hsu and A. Liu, "Multiagent based multi-team formation control for mobile robots," *Journal of Intelligent and Robotic Systems*, vol. 42, no. 4, pp. 337–360, Apr. 2005.
- [9] HMS, HMS Consortium Website [Online]. Available: <http://hms.ifw.uni-hannover.de/>.
- [10] A. Koestler, *The Ghost in the Machine*. Arkana, 1967.
- [11] M. Sugi, Y. Maeda, Y. Aiyama, T. Harada, and T. Arai, "A holonic architecture for easy reconfiguration of robotic assembly systems," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 457–464, June 2003.
- [12] M. Wooldridge, *An Introduction to Multiagent Systems*. Wiley Press, 2002.
- [13] K. Fischer, "Agent-based design of holonic manufacturing systems," *Robotics and Autonomous Systems*, vol. 27, no. 1-2, pp. 3–13, 1999.