Survey of Swarm Robotics Techniques – A Tutorial

Angie Shia
Special Topics Chair
IEEE RAS Region 6

Abstract

The purpose of this paper is to present a survey of different techniques in coordination, planning and formation to readers that are new to the field. It is meant to give the reader a quick overview of the various approaches that researchers have made to address some problem. Instead of presenting *yet another-paper-on-this-approach*, we put similar papers all into one category, and then present other different, refreshing, or unusual way to resolve the same problem. Under each category, there is a suggested reading section. It lists either some of the other researchers with similar work or suggest the reader that wants more information some work that may be of interest to them.

I. Introduction

The purpose of this paper is to present a survey of different techniques in coordination, planning and formation to readers that are new to the field. It is meant to give the reader a quick overview of the various approaches that researchers have made to address some problem. The work chosen for the case studies are mainly to represent a "category of technique" for some coordination, planning or formation problems. It may not necessarily be the most representative paper in that area. Aside from the obvious difficulty in sorting hundreds and thousands of literature on this topic and wondering how to fairly present them; the main goal of our categorization is to prevent a myopic viewpoint of how to solve a problem. Instead of presenting *yet-another-paper-on-this-approach*, we put similar papers all into one category, and then present other different, refreshing, or unusual way to resolve the same problem. Under each category, the paper either lists some of the other researchers with similar work or suggests the reader where to find more information.

II. Coordination

Coordination is a tricky subject for multi-agents. We have two issues to deal with – First issue, what kind of communication is available? We identified three types- full, limited and none. Full communication means the agents can freely communicated and exchange information. Limited means the communication channel is either unstable or very limited. None means the agents are not able to communicate. The second issue, are the agents distributed or centralized? This affects how they work together. Distributed means the agents are mainly governing themselves. Centralized means there is a leader that is giving orders or making plans for the other agents. In this section, we put the emphasis on communication type. The rationale is that communication issues plague teams of either centralized or distributed alike. Though in theory, distributed teams are supposed to be more resilient to communication failure, but a team is no longer a team if they are not communicating anymore. At that point, the distributed robots are no different from lone robots. Furthermore, while we can always modify a centralized team to a distributed style or vice versa, communication reliability, cannot be controlled. For every new environment the robots are deployed, we do not know what to expect. There can be anything from signal interference to potholes, where a robot falls in and breaks its communication capabilities (like Mars Rover).

However, zero or limited communication makes coordination very difficult, whether amongst human or robots. If we have a known environment that generally would not have communication issues, for example, forest wild fire, we could have certain level of confidence, in using full communication based techniques.

A. Full Communication

I. DISTRIBUTED COORDINATION USING LOCAL PLANNING

Case Study: The MARTHA project

The intention of the MARTHA's project [10], as proposed by Alami et al, is to deploy large number of robots to be used in place such as airport and harbors. It uses local, fluid(continuous) planning, distributed communication. The paper presented the idea of Plan-Merging paradigm. Basically, the paradigm tried to address the issue of resource contention. When a robot wants to elaborate or execute a plan, it will validate with only nearby robots and create a group plan. The paradigm assumed that the chances of a remote robot fighting over the same resource are slim. This operation is called Plan-Merge-Operation (PMO). The authors then suggested a Plan Merge Protocol on how to perform PMO. Each robot requested a PMO to local robots. If the other robots accepted it, robot x will receive the plan of local neighbor and produce a DAG of everyone's plan. If there was a resource contention, it will flag as a need for synchronization and notify those affected robots. The PMO did not alter the plan of any robot; it just changed the time of execution to avoid resource conflict.

Figure 1 shows the global plan similar to the one in Alami et al's paper. The white circle represents original states in the plan, the gray circle represents events with added synchronization and the black circle represents a new plan after synchronization.

Alami et al acknowledged that there were situations where there could be a PMO deadlock between two robots contenting for a resource. The authors did not explicitly address how to solve the deadlock problem. They did mentioned that their technique was safe, but did not address all planning problems. Their paper suggested a robot to try to take over the planning of all affected robots. However, if this keeps happening and propagates upwards to more and more robots, it would eventually lead to centralization.

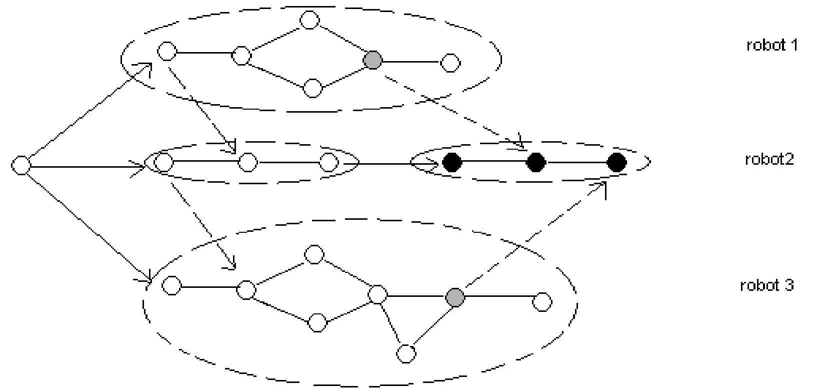

Figure 1. MARTHA Global Plan

Suggested Readings

Readers interested in local based coordination or planning can review Zu et al[25]. They also proposed a local planning for motion and path.

II. DISTRIBUTED COORDINATION USING DYNAMIC ROLE ASSIGNMENT

Case Study: The Azzura robot team

Iocchi et al [6] presented coordination through dynamic role assignment by communication between robots. The robots had both the capabilities of autonomy coordination (in situation where communication link is broken) or communication coordination (planning based on coordination) or a mixture of both. The architecture also allowed heterogeneous robots (different processor or platform) to join in the team.

There were two layers in the architecture – coordination protocol and communication layer. The communication layer enabled inter-robot data exchange. It was a type of Blackboard architecture. The experts controlled real time processes in the robot and also facilitated communication between robots. The coordination protocol was used in robot formation. Each robot could vote for a formation (a process that decomposes tasks into roles). The formation with the most votes would be used. Then, for every role, a utility function computed a utility value of each robot for that role. The robot that had the highest utility value for a particular role would be assigned that role. There may be a situation where a role was not assigned or assigned to more than one robot. This was usually due to communication failure or two robots having the same utility computed for a role. This was not a huge issue since formations were reevaluated every x period of time and the role assignment could change; either because new strategies or some robot cannot fulfill a role. The authors also proposed their own communication protocol based on UDP called EEUDP.

Suggested Readings

Tan and Lozano [9] presented a dynamic scalable and fault tolerant algorithm for robots to autonomously reconfigure itself in case of hostile environment.

B. Limited Communication

III. DISTRIBUTED COORDINATION USING BIDDING AND MAPS

Case Study: Bidding and Map Synchronization

According to Sheng et al [7], bidding algorithm had shown to be very effective in reducing time and distance traveled in robot exploration. One of the problem the authors addressed in which other researchers just assumed, is full communication. The authors believed that chances are, partial communication was a more likely scenario and subnetworks would emerge. Their paper focused on that angle and also tried to address two questions: 1) design a totally distributed, bidding based coordination algorithm 2) address communication limitation while maintaining performance.

For the first question, Sheng et al proposed a “Map-Bid-Travel” scheme. According to their scheme, when an agent was traveling, it would not get involved in bidding. The authors did not explicitly specify why. Presumably, it was because they want to maintain map consistency during the bidding process since the map of a traveling robot kept changing. In one of the issues they raised (pertaining to question 2), bidding should only occur when all the robots in the same subnetwork had the same map. Mapping happened when a robot moved around and acquired new information. It would then notify the other robots in the same subnetwork. Bidding happened when a robot wanted to move to some frontier cell. A frontier cell was a free cell that was next to an unknown cell. The robot would compute a bid B that was based on Djjkstra's shortest distance algorithm on all the frontier cells. The cell with the best value would be the bid B. B is maximum of net gain $g_{i,}$ where g_i is a weighted combination of I - information gain, D - distance to cell , and lambda, the nearest measure of a robot to the cell.

$$B = \max g_i$$
$$g_i = \omega_1 I_i - \omega_2 D_i \_ \omega_3 \lambda_i$$

Next, the robot sent its bid to all the robots in its subnetwork. If no other robot submitted a bid or a higher bid, the robot would then declare itself the winner and notify the other robots in its subnetwork. The other robots would then update their maps. The winning robot then moved to the last step of the Map-Bid- Travel scheme - it moved to that cell. This process would then start all over. The second question the paper tried to address, was how to deal with limited communication. During the bidding, network delay could cause inconsistency map exchange and hence produce incorrect bidding winners. The authors proposed a “Grace Window” in which robots could transmit map information. As long as the grace window time *Tgrace* is greater than communication delay *td*. The maps would stay consistent. In situation where robots from two subnetworks met and wanted to merge their subnetworks, map exchange process can be extremely inefficient. Since these robots had to propagate the information to other robots in their subnetwork, redundant information could be passed around if the mission was almost complete and most of the area had been explored. To address this, the authors proposed a mapping table that kept track of the last sequence of areas explored. Thus, when robots from two different subnetworks meets, they could just find out what was the last sequence number on other's map and send them the one they did not have. Once this information was exchange, the robots could then propagate the new information to the other robots in their subnetwork.

Suggested Readings

Though there are many papers on robot negotiation and bidding algorithms, such as Bererton et al[5], Diaz and Stenz[14], Zlot et al[16], Sheng et al[8], Calliess and Gordon[18], and Pongpunwattana[19], auction and negotiation is a very old strategy in economics. Beaudoin et al [11] presented the classic topic for wood procurement planning.

C. No Communication

A. COORDINATION THROUGH OBSERVATION

Case Study: IDEAL Belief Network

Traditional agent coordination techniques involved an agent acquiring another agent's model or plan. This required explicit communication or inference of other agent's goal based on known belief system that they would do the right thing (reflected in highest utility). Explicit communication required agents to be able to fully communication and exchange their plans. In situations where there were communication breakdowns, whether due to signal interference or range limitation, agents would not be able to coordinate. Utility required agents to do the right thing, but that could not be guaranteed among foreign or heterogeneous agents.

Huber & Durfee [2] presented an alternative method to infer other agents plan. They suggested an agent infer another agent's plan or goal based on observing its “actions”. For example, if an observing agent watched another agent move in some direction, it could give the observing agent a “guess” on where they were going. The observing agent could then decide based on their heuristic plans what they would do. The authors used a Bayesian type network called the IDEAL belief network Every new observation was added to the IDEAL belief network. A current location is set to the probability of 1 and a goal is set to 0. Every time there was a new observation, the new observation node was computed with some probability and the other nodes' probabilities were updated.

One problem with the authors' suggestion was that it involved heavy visual observation; which meant it relied heavily on sensors, and the accuracy of the sensors. Also, authors acknowledged that the belief network could be fooled if the other agent made an evasive maneuver, miscalculated, or, was trying to avoid an obstacle. There was another issue the authors did not address: if there were many agents clustering together, in which one or more of the observed agent was inside that group of agents, how would the visual sensor be able to differentiate which was the agent they were looking for. This was called the "hiding in the crowd" problem.

The authors also brought up the question of early or late commit. That is, when does the observing agent decide that they had figure out what the other agents' intentions were. The problem with early commit was the risk of miscalculation or misinterpretation of the other agent's intent or goal, and thus causing the observed agent to make unnecessary steps or moves. The other extreme end was very late commit – The observing agent could wait till the observed agent arrived at their goal and then compute a path plan and move to its destination. The issue with late commit was that the author did not specify how the observing agent would know that the other agent had arrived. In their testbed, all the agents were in a small controlled grid environment. Hence, it was possible they all had full visibility. In a real environment however, obstruction, distance and so forth could hinder an agent's view of the other agents. It would seem that late commit would not be possible unless the observing agent was following the observed agent. If so, the observing agent would wound up incurring extra steps when following the other agent. This seemed to be what the authors were trying to avoid.

Suggested Readings

Current literature specifically proposing robot coordination without communication is not very common. Here we suggest reviewing literature discussing specific sensors, computer vision (used in Unmanned Aerial Vehicle (UAV), NASA Mars Landers), heat, infra-red, or sonar. Readers can then decide how to integrate it into the robot decision making processes.

III. Planning

Like coordination, planning is a very broad term. The current literature generally use the term to mean path or goal planning. In this section, we present three categories of planning techniques. The first category is probably the most commonly seen planning approaches – using some form of constraint, heuristics, or a mixture of such techniques weave together. The second category, introduces us to a different angle of looking at the problem. It treats robots as particles. The third category is more rarely seen approach – using neural network.

I. Planning Using Constraint, Heuristic, or Hybrid Methods

Case Study: Mars Rover and Lander, JPL

In the paper by Chien et al [12] of Jet Propulsion Laboratory (JPL), they examined three different strategies to handle multiple rovers in Mars rock sampling mission – centralized planner, centralized goal allocation with distributed planner on each rover, and fully distributed planner. For starter, they treated the multiple lander mission as a Multiple Saleman Traveling Problem (MSTP). For the centralized planner, they used an extension of ASPEN planner

(Fukunaga et al. 1997) on the rover. The planner delegated the subgoal to each lander. The problem the lander faced was that it could not handle unpredictable environment.

The second method, part central, part distributed, allowed for more advanced mission. The disadvantage was that because there was still a central planner, the landers could not change their subgoal if they could not fulfill it.

In ASPEN there was a technique called "iterative-repair" that resolved conflicting schedule. For the fully distributed planner, the JPL team used Contract Net Protocol, which was basically a bidding or auction technique. The drawback was the large computation required for the landers, as it constantly had to "repair" its schedule (that is, trigger the repair algorithm).

Rabideau et al presented a empirical study that showed the results of all three approaches. The Contract Net protocol accomplished more goal using about comparable distance traveled than the central and hybrid, but it came with a high CPU time price. However, since newer hardware and faster processors were introduced frequently, the CPU time cost would decrease and should be less of an issue as time went by.

Suggested reading

Ephrati and Rosenschein[27] proposed a bottom-up approach. The robots solved their subgoals and merged them to an eventual global plan. desGardin and Wolverton [13] has a similar idea of using subgoals and merging. Chakraborty et al proposed using differential evolution algorithms for both central and distributed planning. Clark , Rock and Latombe[17] also merges central and distributed in their proposed framework.

II. Planning Using Particle Physics Theory

Case study: Swarm intelligence theory

Rigatos [26] proposed the Swarm Intelligence theory. He used concepts in statistical physics to mimic biological systems. Each robot was treated like a particle. Each "robot particle" had mass, position, velocity, friction, attractive force (to other robots) and repulsive force (to prevent collision). The potential of each particle can be computed. The new position of time t of the "robot particle" $x_i(t+1)$, was computed based on its moves from its current position and the best moves of the other robots from their position. The desired effect was to have each particle should move in the direction of decreasing cost as time went by, with the velocity of the particle approaching 0 as time went to infinity. To address this, Rigatos used ordinary differential equations to find the dynamic behavior of the swarm: $V^i(t) = c^i{}_1 e^{\rho 1t} + c^i{}_2 e^{\rho 2t}$

The initial test result was satisfactory, showing an avoidance of local minima problem. However, the attractive and repulsive force had to be loose to prevent collision. Also, the coefficient of the differential equation had to be adjusted to prevent explosion (of velocity).

Suggested Readings

Kolushev and Bogdanov [21] also proposed treating robots as dynamic obstacles and used graph optimization techniques.

III. Planning Using Neural Network

Case Study: Neural Network based Planner

Li et al [23] proposed adding neural network to the Intelligent Planner component of the Coordinated Hybrid Agent (CHA) framework to handle moving obstacles.

The CHA consisted of 5 components – Communication (standard), Coordination states (represents how much the agent has completed the task), Coordination Rule Base(social laws), Intelligent Planner(generate actions) and Implicit Communication (communication with other agents for special tasks). The neural network planner was a topologically organized neural network. It had only local connection to simulate real life neural network. The neural dynamics was characterized by the Shunting equation. A state of the robot was represented by a neuron. The neuron only had local and lateral connection to other neurons (neighbors). The paper presented an equation to compute the dynamics of the jth neuron of a robot, taking into account items such as: passive decay rate, excitatory inputs, recovery rate of the network (from inhibitory stimulus of other agents), and obstacles. Weights were added to the connection between two neurons. In the neural network landscape activities, targets were set at peak while static and moving obstacles were set in the in the valley. A robot was able to avoid the moving obstacles as it kept moving upward.

Suggested Readings

Currently, there are not a lot of literature using neural network for mobile robotics, even though the ancestor of swarm robotic - Distributed Artificial Intelligence (DAI), can trace its roots to it [30], but more commonly seen, are linear or graph theory, and Bayesian type algorithms. However, neural network has been around for a long time and there are many algorithms available. Interested reader can review neural network literature for other discipline, such as medicine and engineering.

IV. Formation

Formation suffers from similar communication as coordination in Section 2. However, in this section, we will just treat formation as a subset of coordination and assume that the robots can communication to the level to achieve formation.

A. Formation through Leader-Follower and Consensus Based Approaches

Case Study: Formation of nonholonomic robots

Desai, Ostrowski and Kumar [3] had written multiple papers on this topic. The authors used graph and control theory in their proposal for formation of non-holonomic robots. They employed a leader follower type of formation style. Each of the robots was represented as a node on a acyclic graph. Then, using non linear control theory, they first computed the position and trajectory of the leader, and then they computed the position of each robot with respect to the leader and the other neighboring robots. In general, the formation model was defined as *(g, r, H)*, where *g* is gross position and orientation of the leader, *r* is the relative positions of the robot in the team, and *H* is a control graph used by a robot

A robot could follow one or two leaders and they were computed differently. Following one leader was termed *l -ψ* and following two leader was *l - l* , where *l* represents length between robots and *ψ* was their relative angle. The authors pointed out the short coming of using dependent coordination strategies; specifically, that if all the depending robots' actuators were considered together, the problem would be exponential as the number of robots increases. The paper proposed a decentralized control laws for each robots. It allowed changes in the formation shape. The control laws (strategies) were modeled as digraph. A transition matrix allowed a robot to transition from one control graph to another.

Suggested readings

Many papers investigate formation through algorithms. Desai et al had written another paper of similar work [20]. Some other papers include Ballard[1], he uses harmonic oscillation to control formation.

B. Formation Frameworks and Architectures

Case Study 2: Architecture for formation control

In this case study, Fierro and Das [28] extended the above work and proposed a layered architecture for the control strategies. The control architecture had four modules: group control, formation control, kinematic control and dynamic control.

The first module - group control, computed the group trajectory g(t). The trajectory was represented as a directed graph. The second module - formation control, consisted of three networks: 1) physical network pertaining to sensors, 2) communication network for inter-robot dialog, 3) computational network that reflected the resource available for each robot. The physical network was a directed graph showing the flow of sensory data. The communication network was represented as a undirected graph, where the edges were communication channels. The computational network was a directed acyclic graph. The nodes represented robots and the edges were control policies. The third module - kinematic control, were based on feedback linearization [29]. The fourth module - (adaptive) dynamic control, enabled the system to learn new robot dynamics in real time. This controller was necessary as different robot dynamic could cause degraded performance in a close loop system.

In order to manage the state explosion of the control graph, Fierro and Das proposed some constraints, such as, robots could only follow robots they could see (any robot they could not see, even if they could communication with them, could not be considered) . The control graph assignment algorithm could be used both in centralized or distributed mode.

Suggested Readings

Some authors, such as Ren and Sorenson[4] presented a unified, distributed architecture for formation control.

C. Other

In this section, we briefly mention some other work does not fit into the main categories of swarm techniques but is still quite interesting.

I. MIMICKING BIOLOGY - ROBOT PHEROMONES

Werfel [22],[24] used a mixture of formation, planning and coordination strategies to addresses the topic of self-assembly robots. He introduced the concept of *extended* stigmergy. Stigmergy was a concept from biology that claimed that insects such as termites or ants, passed information to one another by leaving something behind a scene, such as pheromones. Werfel extended this concept and enabled robots to pass information to one another.

II. MODELING AND WORKFLOW

Research was only relevant if it could be used eventually. Moreno et al [15] argued that Business Process Management (BPM) systems were very similar to AI Planning and Scheduling (P&S). The authors introduced a workflow modeling tool SHAMASH. The SHAMASH architecture had two components – knowledge base and subsystems. The knowledge base was a rule based system that held domain information. The subsystems had four subcomponents: Author, Simulator, Text generation and Workflow generation. The author subsystem was a user interface that enabled users to enter model information. The simulator checked to see if the model would work. The text generation subsystem converted the visual model to text and the Workflow generator produced a Workflow Process Description language (WPDL) file for input to workflow engines.

D. REFERENCES

[1] L. Ballard, “Experiments in Distributed Multi-Robot Coordination”, M.S. Thesis, Utah State University, 2008.
[2] M.Huber, E. Durfee, “Deciding When to Commit To Action During Observation-based Coordination”, proceedings of the First Conference on Multi-Agent Systems, 1995.
[3] J. Desai, J. Ostrowski, V. Kumar, “Modeling and Control of Formation of Nonholonomic Mobile Robots”, IEEE Transactions on Robotics and Automation, vol. 17, no. 6, 2001.
[4]W. Ren, N. Sorensen, “Distributed coordination architecture for multi-robot formation control”, Robotics and Autonomous, 2007.
[5] C.Bererton, G. Gordon, S. Thrun, P. Khosla “Auction Mechanism Design for Multi-Robot Coordination”, Proceedings 17th Annual Conference on Neural Information Processing Systems, 2003 .
[6] L. Iocchi, D Nardi, “Distributed Coordination in Heterogeneous Multi-Robot Systems”, Autonomous Robots, vol. 15, pp. 155- 168, 2003.

[7] W. Sheng, Q. Yang, J. Tan,N. Xi, “Distributed multi-robot coordination in area exploration”,Robotics and Autonomous Systems, vol 54, pp. 945-955, 2006.
[8] W. Sheng, Q. Yang, J. Tan, N. Xi , “Risk and Efficiency: A Distributed Bidding Algorithm for Multi-robot coordination”, Proceedings Of the 5th World Congress on Intelligent Control and Automation, 2004.
[9] J.Tan, O. Lozano, “A Distributed Graphs Model for the Coordination and Formation of Multi-robot Systems”, Submitted to International Journal of Robotics Research.
[10] R. Alami, S Fluery, M. Herrb, F. Ingrand, F. Robert, “Multi-Robot Cooperation in the MARTHA Project”, IEEE Robotics and Automation Magazine, 1998.
[11] D. Beaudoin, J.M. Frayret,L. Lebel “Negotiation-based distributed wood procurement planning within a multi-form environment”, CIRRELT-2007-10, 2007.
[12] S. Chien, A. Barrett, T. Estlin, and G. Rabideau, “A Comparison of Coordination Planning Methods for Cooperating Rovers”, Proceedings of the fourth international conference on Autonomous agents, 2000.
[13] M. desJardin, M Wolverton, “Coordinating Planning Activity and Information Flow in a Distributed Planning System”, AI magazine, Winter 2000.
[14] M. Diaz, A. Stentz, “A Market Approach to Multi-Robot Coordination”, CMU-RI-TR-01-26, Robotics Institute, Carnegie Mellon University, 2001.
[15] M Moreno, D. Borrajo, A. Cesta, A. Oddi, “Integrating planning and scheduling in workflow domains”, Expert Systems with Applications, vol. 33, pp. 389-406, 2007.
[16] R. Zlot, R. Stentz, M. Diaz, and S. Thayer, “Multi-Robot exploration Controlled by a Market Economy”,IEEE International Conference on Robotics and Automation, 2002.
[17] C. Clark, S. Rock, J. Latombe, “Dynamic Networks for Motion Planning in Multi-Robot Space Systems”, proceedings Of the 7th International Symposium on Artificial Intelligence, Robotics and Space, 2003.
[18] J. Calliess, G. Gordon, “No-Regret Learning and a Mechanism for Distributed Multi-agent Planning”, Proceedings of the 7th International Conference On Autonomous Agents and Multi-Agents Systems, 2008.
[19] A. Pongpunwattana and R Rysdyk, “Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments”, AIAA JACIC, pp. 580-604, December 2004.
[20] J. Desai, J. Ostrowski, V. Kumar, “Controlling formation of multiple mobile robots”, Proceedings Of the IEEE International Conference of Robotics and Autonomous, vol4. Pp 2864-2869,1998.
[21] F. Kolushev, A. Bogdanov, “Multi-agents Optimal Path Planning for Mobile Robots in Environment with Obstacles”, LNCS, vol.1755, pp. 503-510, 2000.
[22] J. Werfel, “Anthills Built to Order: Automating Construction with Artificial Swarms”, Ph.D. Thesis , Massachusetts Institute of Technology, 2006.
[23] H. Li, S. X.Yang, Y. Biletskiy, “Neural Network Based Path Planning For a Multi-Robot System with Moving Obstacles”, 4th IEEE Conference on Automation Science and Engineering, 2008.
[24] J. Werfel,Y . Bar-Yam, D. Rus, R. Nagpal , “Distributed Construction by Mobile Robots with Enhanced Building Blocks”, Proceedings of IEEE International Conference on Robotics and Automation, 2006.
[25] L. Zu, L. Chen, Z. Liu, P. Yang, “Research on Path Planning Method of Multi Mobile Robot in Dynamic Environment”, Proceedings of the 7th World Congress on Intelligence Control and Automation, 2008.
[26] G. Rigatos, “Multi-Robot Motion Planning Using Swarm Intelligence”, International Journal of Advanced Robotic Systems, vol. 5, no. 2, 2008.
[27] E. Ephrati, J Rosenschein, “A heuristic technique for multi-agent planning”, Annals of Mathematics and Artificial Intelligence, vol. 20, pp. 13-67, 1997.
[28] R. Fierro, A. Das, “A Modular Architecture for Formation Control”, Third International Workshop on Robot Motion and Control, 2002.
[29] A. Isidori,W. Kang, “H_∞ control via measurement feedback for general nonlinear systems”, IEEE Transactions on Automatic Control, vol 40, no. 3, pp. 466-472,1995.
[30] E.H. Durfee,”Distributed Artificial Intelligence”, University of Michigan, NSF research paper, 1994.