S^2BHCA

— Multiple AUVs Cooperation Oriented Control Architecture

You Guangxin
Harbin Engineering University, Harbin, P.R.China

Abstract:

Real oceanographic survey, or other similarly applications should be applications of multiple AUVs. The reasons are efficiency, cost, robustness etc. Well-defined AUV control architecture is one of the key technologies for satisfying these demands. In this paper, we proposed the S^2BHCA (Skill & Simulation Based Hybrid Control Architecture) architecture for these applications, which is a kind of multi-robots cooperation oriented intelligent control architecture based on hybrid ideas. The S^2BHCA attempts to incorporate the virtues of the reactive controller and of the deliberative controller, plus the additional online task simulation ability for cooperation and the learning ability. Simple task expression, fast reactiveness, dynamic cooperation and cooperation mechanism independency, learning ability, robustness and modularization can be achieved by realizing the AUV controller based on S^2BHCA

1 Introduction

Future's applications of AUV system must involve those of multiple AUV system. The reasons are efficiency, cost, robustness etc. Especially in the fields of large-scale oceanographic survey (such as AOSN [*Bellingham1997*]), minefields sweeping, high reliability underwater autonomous system, the multiple AUV system plays the irreplaceable roles in these applications. There will be many new problems in front of us, when building such systems, for not only the increased number of the AUVs, but also the cooperation and coordination between AUVs should be considered at the same time. Dynamic environment, dynamic AUV team itself and multiple requests of operators also make the control of such a distributed system harder.

In this paper, we focus on the control architecture of AUV in a coordinated team. Most existing AUV controllers are designed for single robot application and not very suitable for the cooperation or coordination tasks; little efforts for the multiple AUV applications are made.

Many AUV designers accept the deliberate hierarchical control architecture based on NASA's NASREM standards [*J. Albus etc.1987, Kimon etc. 1997*]. The main idea is the principle of increasing intelligence with decreasing control precision; higher-level layer will make the planned and sequenced subtasks for low-level controllers to perform. But how can these controllers deal with the tightly coupled real-time coordination between AUVs, while they are lack of flexibility and have longer response time? And for multiple robot control, the hierarchical architecture tends to create a centralized cooperation layer, which will be impossible when large number of AUVs involved.

Reactive or behavioral approach does not use planned agenda for vehicle control, they believe that the intelligence arise from the process, in which agents adapt to the environments based on deferent behaviors [*Brooks1986, Arkin 1989*]. These approaches are quite simpler and require less computational resources. Coordination either happens between coordination behaviors such as formation, or represented as the "emergent" behavior based on none coordination behaviors [*Maja1997. Parker1998.*]. One of such approach is Kumar's control architecture [*Kumar etc. 2000*] for AOSN application. But this approach also face problems: how to explicitly express task and how make the long-term decisions in a life long mission?

Hybrid approaches maybe the best, which attempt to incorporate the virtues of both deferent control ideas. Phoenix , Ocean Voyager II , ODIN [*Kimon etc. 1997*] belong to this category. As designed for single robot application, they are lack of completely supports for coordination between vehicles. In this paper we proposed a new AUV hybrid control architecture: S^2BHCA (Skill & Simulation Based Hybrid Control Architecture) [figure1], which is specially designed for the multiple AUV system applications. Simple task expression, fast reactiveness, low costs, well robustness, modularization can also be achieved by realizing the AUV's controller based on S^2BHCA.

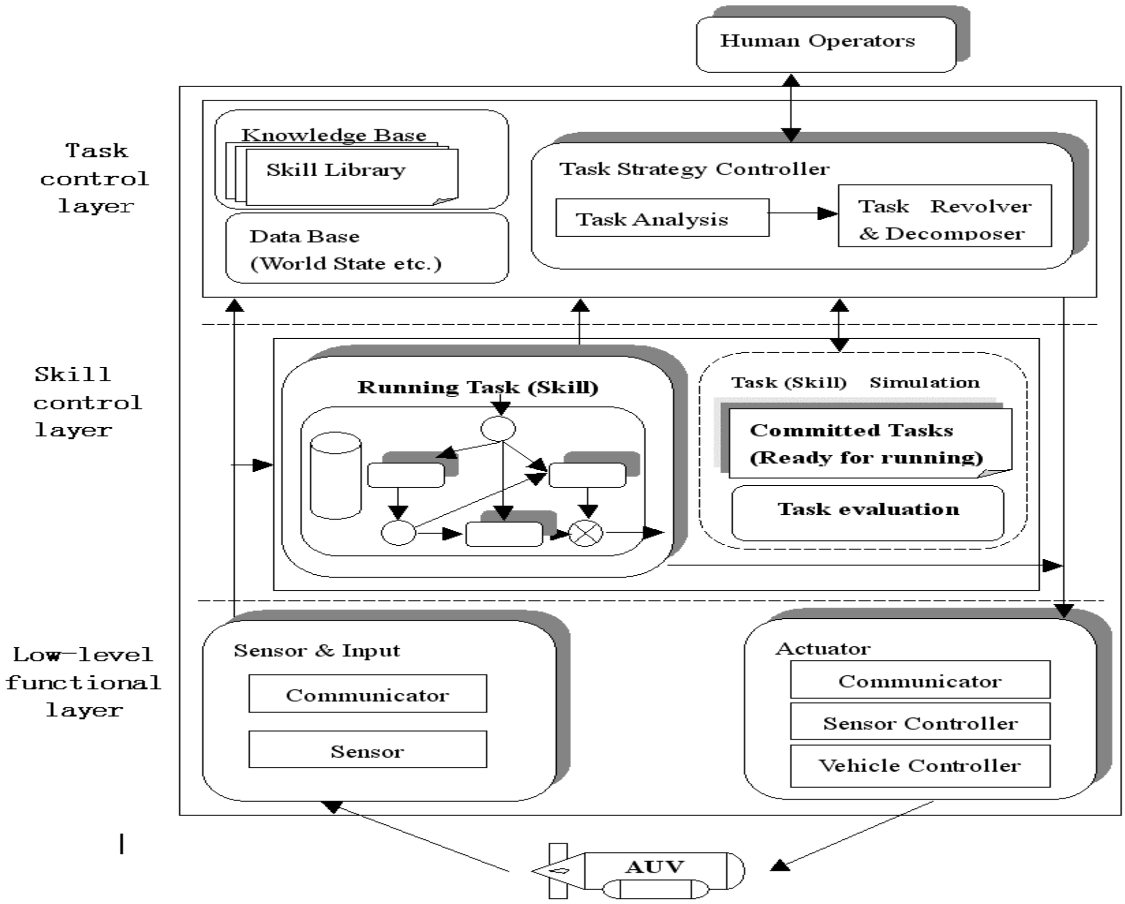

Figure 1 The S^2BHCA control archtecture

2 The S^2BHCA Solution

The S^2BHCA is a three-layered controller [figure1] including the low level hardware functional layer, reactive skill control layer and the task control layer standing for hardware control, behavioral reactive control and deliberative control respectively. The core idea is the concept of "Skill", which aim at bridging the gape between reactive and deliberative control. The concept of "Skill" resides in both task control layer and reactive skill control layer, having deferent form of same meanings in deferent layers. In task control layer, a skill is a cell of functionality for either building complex missions or performing coordination among AUVs, while in reactive skill control layer i is a behavior-based reactive controller, which gives the AUV fast reactiveness and "emergent" intelligence. The granularity of concept "Skill" is between "Behavior" and "Task": it is built atop of the concept of "Behavior" and is the cell for building tasks.

2.1 Extended Behaviors

Maja introduced more internal states to robot's behaviors. She defines it as "a processing element or a procedure for a goal [*Maja1999*]", and the behavior's input maybe the sensory data or other behavior's outputs. The controller is built on a network [figure3] that consists of behaviors and the states among behaviors. We accept this idea, believing that the behaviors can have goals and some "plan" abilities, running concurrently. But how many states within and between behaviors are appropriate for behavior-based planning? The number of states in the behavior network will increase dramatically when making longer-term "planning"[*Jung etc.1999*].

We extend the concept to a more meaning one by introducing the "mechanical" thinking behavior, which used to be recognized as a deliberate process. The "mechanical" thinking behavior takes sensor's data and the symbolic knowledge as input and produces the planned output which are usually other behavior's inputs or goals. We believe that this purposive "thinking" is also basic behavior of human for some working, while it just focus more on the internal states compared with the conventional behaviors. The "mechanical" thinking behaviors will give the network more optimal response to environment and longer-term decision ability, when the robot has to perform life-long tasks in complex working environment. But this kind of behavior usually has more computational complexity, which make it difficult to complete the behavior network's running cycle within specific time. To overcome this problem, the network is realized as an asynchronous network in which each behavior node runs at its own frequency; the reaction of whole network is the reaction of the fastest node.

Learning can greatly improve the behavior's functionality, especially for the model based control behaviors [*Parker L 1993*]. But the learning results may not be valid if the task or environment changes. For a example, when AUV just on its course to the target, the "Collision Avoid" behavior should keep the robot far away from the obstacles at relatively high speed; while in target identification process, the behavior may keep the vehicle just a few meters or less close to the targets at high control precision. So we separate the knowledge that is task or environment specific from the behavior control module, and make it standalone.

2.2 Skill & Skill Simulation

The concept of "Skill" comes from the specialty of human. In our society, no individual is versatile; he is just an expert in a few fields and his main activities in society focus on his specialty. On the other hand, specialty means more detailed ability and knowledge in the limit field, which is a group of highly organized behaviors in the view of the behavior-based AI. The "Skill" is such a concept of specialty that resides in both task control layer and reactive control layer, taking deferent forms separately.

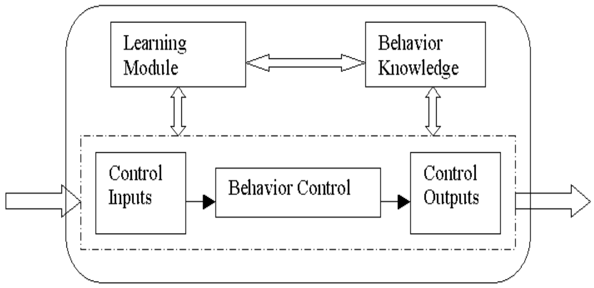

Figure 2 The Behavior Module

In fact the "Skill" is a task-specific behavioral controller in reactive control layer, including a special collection of basic behaviors and the specially tuned behavior-fusion method just for that group of behaviors, which enable the robot to function like a trained expert in special fields. The "Skill" is realized as an asynchronous behavior network based on "Extended Behavior" [figure 3], in which there are four deferent kind of modules: behavior control modules, the "Skill" or task related knowledge separated from behavior controller, the behavior fusion methods, and the states. Because the "Skill" only contains the carefully selected behaviors necessary for specific task, the network will not increase greatly when more behaviors are introduced for the complex missions requirement. And the designation of such a network and the behavior fusion method will also be simpler when the number of behaviors is kept small.

Though deferent "Skill" may contain the same behaviors such as "Avoid", we extract the task-optimized knowledge from these behaviors and save it as part of "Skill", in order to make behavior controller perform more efficiently for deferent task. Modularization and better maintenance can also be achieved by the knowledge extraction, for the behavior controller itself needn't be modified for deferent situation.

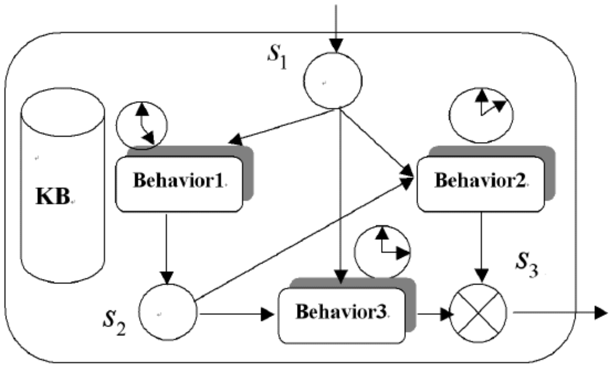

Figure 3 The Skill Controller Networks

The "Skill" controller is something like the coordination module responsible for the subtask's schedule in conventional hierarchical architectures. But here we prefer to the behavior-based approach, let the environments trigger the appropriate behaviors in order to achieve the fast reactiveness and the "emergent" intelligence.

While in task control layer, a skill is functional module, on which a real mission is built. By evaluating some of the states within the behavior network we can give the evaluation for a task carrying out based on this skill. This evaluation can take any form related to the task such as the time, the energy or the distance to the working area, but all this form should return a scalar "number" for evaluation. Deliberate controller for determining the mission state and the task level coordination between agents can use it for task analysis.

The "Skill" has two deferent running modes: the real mode and the simulation mode. The real mode is the normal running state of a "Skill", in which it operates on sensor's data and carries out the task desired. A "Skill" can also be designed to have the on-line simulation ability, which is the unique feature of S^2BHCA. When in the simulation mode, the "Skill" reads the sensory data as usual, but it just refresh the some states within the behavior network and doesn't send commands to low level function layer to execute. Not all of the behaviors within the network of the "Skill" in the simulation mode are activated; only part of behaviors can be activated to refresh the states for skill evaluation. So we can predict a task's cost beforehand by reading special states of the network within the "Skill" submitted in simulation mode. This prediction of a task's cost is the basis of many coordination mechanisms such as "Market Approach [*Dias2000*]", serving as the utility function for dynamic multiple agents task allocation problem（such as giving the dynamic price information in the market based cooperation）[*Gerkey etc.2002*].

2.3 Layered Control & Layered Coordination

We believe that the deliberate control and the behavior-base control are of the same importance, and neither can replace the opponent. The S^2BHCA stands for this idea, which uses the concept of "Skill" to incorporate the two deferent control approaches. The deliberate task control layer and the reactive "Skill" control layer plus the low level hardware function control layer make the whole AUV controller together. The same is the idea for coordination between AUVs, which involved the strategy level and the real-time behavior level coordination. Deferent level coordination will happen in corresponding control layer.

The task level controller manages the AUV's mission state at the abstract level above the concept of "Skill" and cares about the longer-term missions. Skill based task expression and task analysis is very convenience for controller designers because of its granularity. The task controller determines which skill should be selected for performing in order to achieve the goal that operator gives or suggested by other robots. Tasks that must be carried out at this AUV locally will be converted to the skills running in real mode right now, while the optional tasks that can dynamically allocated among the multiple AUV system will experience a little more complex process. The task is firstly submitted as a proposed task in the simulation mode for evaluation beforehand, and then it will be accepted and queued in the committed task queue [figure1] if the decision is "yes", otherwise rejected. One of the committed tasks will be carry out in real mode if the current running task completes. Anytime there is only one skill running in real mode, which controls the AUV. The queued committed tasks are examined continually for re-evaluation and if necessary can be re-delivered to other AUVs. The task controller is responsible for strategy level coordination such as task allocation, multiple AUV system self-organization etc. By the on-line simulation of skills, the controllers can predict the cost (or utility) of the tasks and analyze the cooperation requires from other robots based on the prediction.

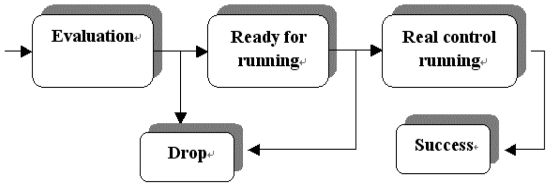

Figure 4 The Process of Task Submission

Once a skill is created in real running mode or simulation mode by the task controller, it will run in the reactive skill control layer. Thus in this layer, there are three running modules: the skill in real mode, the committed tasks queue in simulation mode, and a proposed task for evaluation in simulation mode.

The S^2BHCA is a truly distributed MAS based control architecture. All robot agents are physically separated from each other and there is no predefined central task controller or commander who monitoring the whole system, which makes the robot team more robust. Tasks are injected into the system by sending messages to one of the AUVs who will delivery the tasks to the others based on the chosen coordination mechanism. Through applying deferent cooperation mechanism in the deliberative controller or in reactive skill controller will provide more styles of AUV teams.

The S^2BHCA is simpler and easier to be realized than the traditional hierarchical architectures, atop of any modern real-time multitask OS such as Vxworks, windowsNT (for more interactive control in controller development), which means just one embedded control computer required and less hardware costs when more robots produced.

3 The Realized Controller

Our primary realization of such a controller, named as “MAUVS-CS”, is used to validate the idea. Most theoretic researches are conducted in the simulation environments, for which we develop a new simulator, MAUVS-SE, providing real-time simulation services for multiple AUVs simultaneously. The model of vehicle is the “ZhiShui3”, a test-bed for oceanographic research [figure5]. The simulation experiment involving four AUVs has successfully demonstrated the advantages of the S^2BHCA motioned above, which involves two deferent level of coordinations. A proposed lake test involved two AUVs equipped with these controllers are to carry out soon.

Figure 5 The Behavior Module

3.1 The Behaviors

The AUV’s behaviors are “Extended Behaviors”, including some “mechanical thinking”. All the behaviors described below are the necessary control schemes used for the simulation task. Each behavior is built as the same as in figure2.

a) Collision Avoid.

The “collision avoid” behavior is the basic scheme as a safeguard necessary for each AUV, which is based on a modified artificial potential field algorithm. “Collision avoid” isn’t a cooperative behavior, but in some situation the “emergent” behaviors such as “flocking” can be produced upon it [*Maja1999*].

b) Position Keep.

If there is no active control output, the AUV can’t keep its position in the sea, so we need a dynamic positioning control behavior when the AUV doesn’t want to move.

c) Formation.

This behavior is a typical cooperative action between AUVs, which is, in fact, the dynamically relative-positioning process relying on the other AUV's localization information updated real-time. It is a tightly coupled coordination between robots. Here we adopt the "master-slave" formation control principle, thus deferent AUV will have deferent roles. The localization information can come from either the Ultra Short Base Line (USBL) or the forward-looking sonar. Communication is needed if the AUV can't get other AUV's localization information by its own sensor such as the situation of USBL.

d) Plan Navigate

"Plan Navigate" is a kind of "mechanical thinking" behavior, which works on the map and produces a planned path directing the AUV to the target. The output may also be used for evaluating the skills that include this behavior.

e) Searching.

The algorithm of "Searching" can be either random action or planned action, and the output of "searching" can also be used for skill evaluation. Here we adopt the planned search rules.

3.2 The "Skills"

As a primary realization, we developed three basic skills for AUV operations: "Idle", "RallyAt" and "SearchArea" on the basis of the behaviors above.

a) **Idle**

"Idle" means the AUV is in state of no committed tasks and ready for new ones. But the task controller may be active and there perhaps are other skills running in simulation mode. The details of the skill "Idle" can be described by figure5.

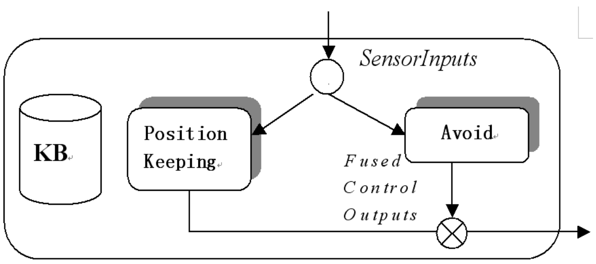

Figure 5 The Skill of "Idle"

b) RallyAt

The AUVs running at this cooperative skill will navigate to the working area in formation. The role of the leader in formation is assigned to the AUV that has more computational resource and environmental information, which is achieved by the coordination among the task controllers based on the simulation results. Figure6 describes this skill.

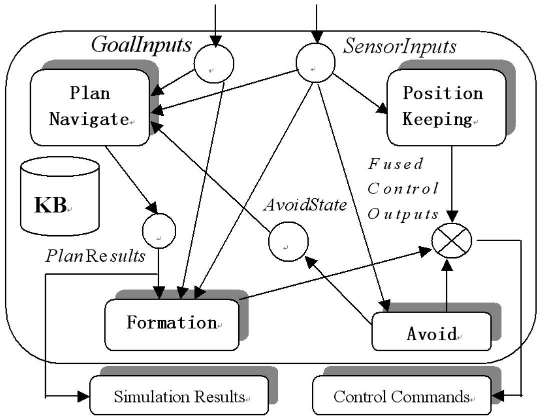

Figure 6 The Skill of "Rally At"

c) SearchArea

"SearchArea" is responsible for covering a specific area just by one AUV, which means it stands for the loosely coupled distributed tasks for multiple AUV system. Figure7 describes this skill.

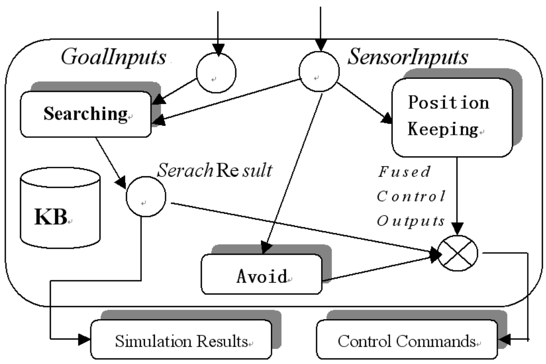

Figure 7 The Skill of "SearchArea"

3.3 Simulation Experiments

To substantiate the validity of S^2BHCA, we have evaluated MAUVS-CS in two very different domains: a long-term scenario consisting of some loosely coupled distributed minefield-covering tasks that a single-AUV can completes, and a formation navigation task requiring tight coordination among the AUVs. There are four AUVs in the simulation experiment, whose control software is identical except for the name of AUV. In the minefield-covering task, the market approach is used for dynamically allocating the searching areas among AUVs, while the formation are performed mostly in the skill control layer as a model based process. Figure9-figure10 are the descriptions of whole mission.

4 Conclusions

In this paper, we propose a new control architecture, S^2BHCA , for multiple AUV applications, which is inspired by the concept of specialty of human. The "Skill" introduced into the architecture can serve as the function module for task control and task level coordination between AUVs. At the same time fast reactiveness and situated intelligence can also be achieved by the reactive "Skill" controller. Simulation mode of a "Skill" which is specially designed for task level coordination gives the AUV the ability to analyze the proposed tasks before accepting it. The simulation experiments have successfully demonstrated that simple task expression, fast reactiveness, dynamic cooperation, robustness and modularization can be achieved by the controller based on S^2BHCA .

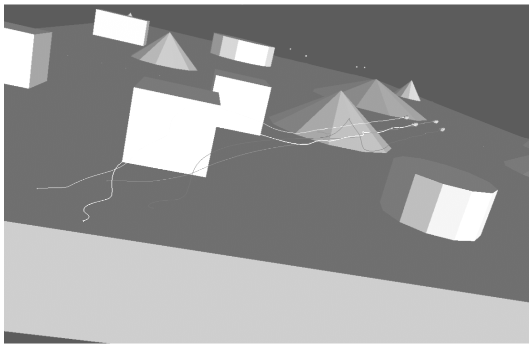

Figure 8 The Experiment of Formation

Figure 9 The Searching Task Allocation

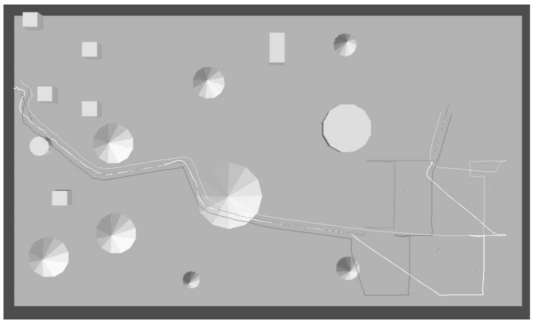

Figure 10 The Whole Process of Mission

5 Acknowledgements

The authors wish to thank Prof. Pang Yongjie, Prof Xu Yuru for their valuable advices, and others in the Lab without whos help I can't complete this paper.

Reference:

Bellingham James G. 1997. New Oceanographic Uses of Autonomous Underwater Vehicles. Marine Technology Society Journal, 31(3):34-47. Fall 1997.

J. Albus, H. McCain, and R. Lumia. 1987 NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM). NBS Technical Note 1235, National Bureau of Standards, Gaithersburg,MD, July.

R. Brooks.1986. A Robust Layered Control System for a Mobile Robot. IEEE Journal on Robotics and Automation, 2(1), March 1986.

Arkin,R.C.. 1989. Motor schema-based mobile robot navigation. International Journal of Robotics Research Research,8(4):92–112.

Ratnesh Kumar, James A.Stover, 2000. "A Behavior-Based Intelligent Control Architecture with Application to Coordiantion of Multiple Underwater Vehicles", IEEE TRANSACTIONS.MAN, AND CYBERNETICS-PART 1:SYSTEM AND HUMANS.VOL.30 NO.6 .NOVEMBER 2000.

David Jung,Alexander Xelinsky 1999. An architecture for distributed cooperative planning in a behaviour-based multi-robot system Rbotics and Autonomous System 26 (1999) 149-174

Maja J Mataric 1999. "Behavior-Based Robotics", in the MIT Encyclopedia of Cognitive Sciences, Robert A. Wilson and Frank C. Keil, eds., MIT Press, April 1999, 74-77.

Maja J Mataric1997. " Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior", Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents, 9(2-3), H. Hexmoor, I. Horswill, and D. Kortenkamp, eds., 1997, 323-336.

Parker L 1993. "Learning in Cooperative Robot Teams" Proceedings of International Joint Conference on Arti_cial Intelligence Workshop on Dynamically Interacting Robots. Chambery France 1993 12-23

Kimon P.Valavanis etc. 1997. "Control Architectures for Autonomous Underwater Vehicles". IEEE Control Systems 1997 vol 21 p48.

Brian P. Gerkey and Maja J. Mataric 2002 Sold!: Auction Methods for Multirobot Coordination. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 18, NO. 5, OCTOBER 2002

M. B. Dias and A. Stentz, 2000."A free market architecture for distributed control of a multirobot system," in Proc. 6th Int. Conf. Intelligent Autonomous Systems, Venice, Italy, July 2000, pp. 115–122.