

Retraction

Retracted: Multi-Agent Systems in Control Engineering: A Survey

Journal of Control Science and Engineering

Received 25 November 2009; Accepted 25 November 2009

Copyright © 2009 Journal of Control Science and Engineering. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted as it is found to contain a substantial amount of material, while without referencing, from the two papers which are:

“Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges,” McArthur, S. D. J.; Davidson, E. M.; Catterson, V. M.; Dimeas, A. L.; Hatziargyriou, N. D.; Ponci, F.; Funabashi, T., *IEEE Transactions on Power Systems*, Vol. 22, no. 4, pp. 1743-1752, Nov. 2007.

“Multi-Agent Systems for Power Engineering Applications—Part II: Technologies, Standards, and Tools for Building Multi-agent Systems,” McArthur, S. D. J.; Davidson, E. M.; Catterson, V. M.; Dimeas, A. L.; Hatziargyriou, N. D.; Ponci, F.; Funabashi, T., *IEEE Transactions on Power Systems*, Vol. 22, no. 4, pp. 1753-1759, Nov. 2007 [1].

References

- [1] F. Daneshfar and H. Bevrani, “Multi-agent systems in control engineering: a survey,” *Journal of Control Science and Engineering*, vol. 2009, Article ID 531080, 12 pages, 2009.

Review Article

Multi-Agent Systems in Control Engineering: A Survey

Fatemeh Daneshfar and Hassan Bevrani

Department of Electrical and Computer Engineering, University of Kurdistan, 66177 15175 P.O. Box 416, Sanandaj, Iran

Correspondence should be addressed to Fatemeh Daneshfar, f.daneshfar@yahoo.com

Received 10 December 2008; Revised 12 May 2009; Accepted 5 August 2009

Recommended by Tongwen Chen

This paper presents a survey on multi-agent system (MAS) capabilities in control engineering applications. It describes essential concepts of multi-agent systems that are related to the control systems and presents an overview on the most important control engineering issues which MAS can be explored. Most important technical aspects in MAS implementation and development in engineering environment are also explained. Design methodologies, standards, tools, and supporting technologies to provide an effective MAS-based control design are addressed and a discussion on important related standards and protocols is given. Finally, some comments and new perspectives for design and implementation of agent-based control systems are presented.

Copyright © 2009 F. Daneshfar and H. Bevrani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Nowadays MAS technology is being used for a wide range of control applications including scheduling and planning [1, 2], diagnostics [3], condition monitoring [4–6], distributed control [5, 7], hybrid control [8], congestion control [9, 10], system restoration [11], market simulation [12, 13], network control [13, 14], and automation [15]. Moreover, the technology is growing to the point where the first multi-agent systems are now being immigrated from the laboratory to the utility, allowing industry to gain experience in the use of MAS and also to evaluate their effectiveness [1]. Nevertheless, despite a growing awareness of the technology, some primary questions often arise from other researchers and, in particular, industrial partners when discussing multi-agent systems and their role in control engineering. These are what benefits are offered by multi-agent systems?, what differentiates them from the existing systems and approaches?, to which kind of problems can they be applied?, if and when MAS technology is supposed appropriate for a particular control engineering application, then other questions naturally follow: how should multi-agent systems be designed? how should multi-agent systems be implemented? are there any special considerations for the application of MAS in control engineering?

As MAS are a new technology, a number of technical challenges need to be overcome if they are to be used

effectively. Then identifying details of those challenges and providing technical leadership in terms of recommendation and guidance on the appropriate use of the standards, design methodologies, and implementation approaches which are currently available, is necessary too.

This paper begins by describing concepts and approaches related to multi-agent systems and consider the principal problems which can be dealt with MAS. Additionally, it presents a comprehensive review of the control engineering applications for which MAS technology is being investigated. However, it discusses necessary and appropriate standards and guidance to design and implementing MAS in the control engineering applications.

2. Definitions

In order to find benefits of MAS to control engineering, the basic concepts and definitions related to multi-agent systems need to be understood.

2.1. What Is an Agent? The computer science community has produced various definitions for an agent [16–20]. A comparison between these definitions and their relative merits and weaknesses, from a computer science view, can be found in [21]. However all the definitions referenced above are different, they all share a basic set of concepts: the notion

of an agent, its environment, and autonomy. According to Wooldridge definition [20], an agent is “a software (or hardware) entity that is situated in some environment and is able to autonomously react to changes in that environment.” The environment is everything external to the agent. The environment may be physical (e.g., the control system), or it may be the computing environment (e.g., data sources, computing resources, and other agents). An agent can alter the environment by taking some action. The separation of agents from environment means that agents are inherently distributable. Under Wooldridge’s definition [20], an entity situated in an environment is an agent if it can act autonomously in response to environmental changes. The definition of autonomy says that an agent “exercises control over its own actions” [21], meaning that it can initiate or schedule certain actions for execution.

Wooldridge and Jennings [22] identified three different classes of agents:

- (i) agents that execute straightforward tasks based on pre-specified rules and assumptions,
- (ii) agents that execute a well-defined task at a user’s request,
- (iii) agents that volunteer information or services to a user whenever it is deemed appropriate, without being explicitly asked to do so.

From an engineering view, this classification has some problems: it does not clearly separate agents from a number of existing systems. According to above definition, some existing systems could be agents. For example, a thermostat device could be considered as an agent. It is situated in its *environment*. It *reacts* to temperature changes of environment. It also exhibits a degree of *autonomy*. Therefore there is a need to know how agents and multi-agent systems differ from existing systems and system engineering approaches.

2.2. Intelligent Agents. Wooldridge [20] extends the above definitions of an agent to an intelligent agent by extending the definition of autonomy to flexible autonom. An agent which displays flexible autonomy, that is, an intelligent agent, has the following three characteristics.

- (i) **Reactivity:** an intelligent agent reacts to changes in its environment in a timely manner.
- (ii) **Proactiveness:** intelligent agents have goal-directed behavior. Goal-directed behavior supposes that an agent will dynamically change its behavior in order to achieve its goals. Wooldridge [20] describes this as an agent’s ability to “take the initiative.”
- (iii) **Social ability:** intelligent agents are able to interact with other intelligent agents. Social ability connotes more than the simple passing of data between different software and hardware entities. It connotes the ability to negotiate and interact in a *cooperative* manner. That ability is normally created by an agent communication language (ACL), which allows agents to converse rather than simply pass data.

Not only the characteristics of reactivity, proactiveness, and social ability help us to distinguish agents from traditional hardware and software systems, it is from these characteristics, that many of another benefits (discussed in this paper) are derived.

2.3. Multi-Agent Systems. A multi-agent system is a system comprising two or more agents or intelligent agents [23]. It is important to know that there is no overall system goal; however each separate agent has local goals [23]. Depending on the definition of agency mentioned above, agents in a multi-agent system may or may not have the ability to communicate with each other directly. However, under Wooldridge’s definitions [20], intelligent agents must have social ability and therefore must be capable of communication with each other. For the sake of this paper, the authors have focused on MAS where this communication is supported. This differentiates the type of MAS discussed in this paper from other types of systems.

3. MAS in Control Engineering Applications

To know how (and why) MAS is applied in control applications requires an understanding of how MAS can be used. Nowadays, MAS is exploited in two ways [24, 25]: as an approach for building flexible and extensible hardware/software systems, and as a modeling approach.

3.1. Using MAS to Construct Robust, Flexible, and Extensible Systems. There are many control engineering applications that flexible and extensible solutions are useful for them. Flexibility is the ability to respond to dynamic situations (environment), correctly. It is very similar to autonomy, and therefore intelligent agents should be flexible, automatically. But if autonomy is the ability of an agent to plan its own actions, flexibility is to select the most proper actions from a set of actions [20]. Some examples of flexible behavior would be like the ability to construct a new plan if a particular control action fails. Extensibility implies the ability to easily add new functionality to a system, or upgrading any existing functionality [25]. For example, in distribution networks, a distributed network control responsible for voltage control may be extended to responsible for frequency control. Across many applications in control engineering, there is also a requirement for fault tolerance and graceful degradation: if any parts of the system fail for whatever reason, the system should still be able to meet its design objective [26]. A MAS can provide a way for building such systems. However, the way in which a MAS provides flexibility, extensibility, and fault tolerance needs to be understood. The properties of agents that produce these qualities are examined below.

3.1.1. Autonomy. An agent encapsulates a set of functionality, (like modular or object-oriented programming [27]). It means that the benefits of standard interfaces and information-hiding are also available in agent programming through the use of messaging with a standard agent communication language, but there is also the additional capability

of autonomous action [28]. In an object programming, external objects can call and execute other object's functions [27]. However in the agent oriented programming, external agents can only send requesting messages to the action of a special agent: the autonomous agent can decide whether to fulfill the request and its priority [28]. This can be useful in situations when an agent is receiving many requests and cannot accomplish them within a reasonable time. The autonomy of each agent and the messaging interface are useful in most of flexible and extensible systems. Because agents are not directly linked to others, then it is easy to take one out of operation or add a new one while the others are running [29]. If one agent is stopped, any agents interacting with it can use the standard service location (Agent Management System and Directory Facilitator [30]) facilities to locate another agent that performs the same task, and by this mechanism, new agents can be added within the system too. The agent framework provides the functionality for messaging and service location, it means that new agent integration and communications are handled without effort from the system designer [30]. This create extensible systems: extra functionality can be added by deploying new agents in system, which use service location to find others to communicate with; and some parts of systems can be upgraded by deploying a replacement agent and removing the old one. Flexibility also considers the appropriate mixture of agents that can be deployed to qualify the individual situations or conditions, and flexible handling of messages between agents that allows the system to self-configure.

3.1.2. Open MAS Architectures. An open agent architecture places no restrictions on the programming language of the system, and allows flexible communication between all agents. This is achievable because of messaging standards [30]. The separation of an agent from its environment means that the messaging language that an agent understands is important for interagent communication, rather than the programming language it was implemented with. A set of standards for an open architecture is defined by the Foundation for Intelligent Physical Agents (FIPAs) [30, 31]. The FIPA Agent Management Reference Model includes the "framework in which FIPA agents exist," defining standards for creating, locating, removing, and communicating with agents. This is generally called the agent platform, and is one part of an agent's environment. One requirement of an open agent architecture is that the platform places no restrictions on the creation and messaging of agents, while a second is that some mechanism must be available for locating particular agents or agents offering particular services within the platform. Under the FIPA model, this is achieved through a separate agent called the Directory Facilitator. It is an agent that manages a list of services offered by other agents within the platform [30]. A closed architecture removes the possibility of an extensible or flexible system, severely limiting the benefits of using agents.

3.1.3. Platform for Distributed Systems. An agent is separate from its environment, it means that it can be placed in

different environments and still has the same goals and abilities (as the agent autonomously schedules action in the response to sensor inputs and messages). For this reason, an agent is distributable and does not have any fixed ties to its environment. In practice, distribution of agents through a network is supported by the agent platform. The platform can be deployed on every computer and the agents are deployed within the platform as usual [30]. On a platform, there is no difference between agents on the same computer and agents on a different computer. This means that the same set of agents can be deployed on one computer, and alternatively on multiple networked computers, without modifying or changing the agent code [30].

3.1.4. Fault Tolerance. Building redundancy into systems is one of the standard engineering approaches to gaining fault tolerance. Building redundancy in MAS involves providing more than one agent with a given set of abilities. If an agent needs the services of a second agent to accomplish its goals, and the second agent fails, the agent can seek an alternative agent (perhaps using the Directory Facilitator) to provide the services it requires [30]. This redundancy may be provided by simple duplication of each agent, and with distribution of duplicates across different computers. Also, the flexibility offered by an open architecture of agents with social ability will provide a tolerance to physical faults, such as the loss of a network connection, or damage to a computer.

3.2. Multi-Agent Systems as a Modeling Approach. Multi-agent systems are more than a systems integration method; they also provide a modeling approach. An agent system can represent a real-world with entities' interaction. Natural representation of the world has been given as an advantage of object-oriented (OO) systems design [27], where entities in a system are modeled as objects. The main benefit of the object programming is data-encapsulation. Agent-based design adds another level of abstraction to this: not only the internal data structures, but also the "methods" (actions) which an agent can do are hidden [29]. However, many control engineering applications can apply this way of viewing the world, such as power systems operation and control. Generators have a degree of autonomy and cannot be directly affected by external system actors [32]; therefore, they can be represented by agents. Such an application would be using agents for both their modeling properties and also as a way of building a flexible, extensible system.

4. Control Engineering Applications

In this section some control engineering applications which multi-agent systems can show their potential benefits more than another kind of controls are described.

4.1. Manufacturing Control. Recently globalization caused traditional manufacturing systems change to inherently multidisciplinary tasks. Then the manufacturing system of 21 century consists of a new complicated set of people, software systems, processes, and equipments (hardware)

[33]. The management and control of such systems are a multidisciplinary task based on knowledge of manufacturing strategies, planning, and operations, and is integrated with communication, information, and control functions, of the whole system [34–36].

The computer integrated manufacturing (CIM) concept has been introduced as a global solution that can deal with all of the mentioned challenges and create more flexibility in product spectrum and processes, agility of the production system, more responsiveness, and integration of hardware and software components [37, 38]. However such a centralized structure is not reliable and leads to system unavailability by single failures at one point in it [39]. For this reason, it changed from a centralized model to a decentralized one.

One structure is to have a group of distributed autonomous, intelligent, fault tolerant, and reusable manufacturing units, which operate as a set of cooperating entities. Each entity is capable to dynamically interact with each other to achieve both local and global manufacturing goals. This new generation of manufacturing systems is named intelligent manufacturing systems (IMs) [1, 2, 40]. Because of huge amount of interactions between the different components and the variety of performed functions, the control system of IMS is currently built in a distributed manner [41].

Definitely an IMS system is a kind of multi-agent system which entities are agents that are autonomously distributed and interact with each other to get system global goals. A software agent approach seems very suited with the control and supervision of each component of an IMS.

Agent-based software systems are becoming a control software technology for manufacturing control systems. A multi-agent based platform can offer distributed intelligent control functions with communication, cooperation, and synchronization capabilities that can cover the behavior specifications of components and also the production specifications to be fulfilled by the manufacturing system (see, [25]).

4.2. Congestion Control. Congestion control concerns controlling of traffic entry into a network (such as telecommunications network, urban and air traffic network [42], and all networks that consist of links and switches, and the controls that govern their operation, that allows for data transfer among links), to avoid congestive problems by controlling the rate of sending packets, and prevents the sender from crushing the receiver with flow control [43].

Since most kinds of networks are distributed geographically then a centralized congestion control cannot be scalable and reliable, in this case a distributed control is a good solution to flow control and prevent of congestion problems (like deadlock): each node (switch) can be controlled by one intelligent agent, that autonomously acts and has relation with its environment (network) also each agent can communicate with other agents (that are responsible for other nodes) to transmit its data sending and receiving rate and know about the traffic flow of each other, so the network

can be seen as a multi-agent system able to evaluate, at each period, the values of the different parameters depending on the state of the node, Note that in this scheme cooperation between the agents can be used in the sense of “cooperation for resolution” and can also permit the agents to mutually increase their knowledge by exchanging information periodically about their behavior and knowledge. Moreover, the agents can take local decisions immediately when they have enough knowledge. When the network is not too loaded, it is possible to exchange information on the life of the network to update and improve the agents’ knowledge. This implies that the agents are able to cooperate and to make better decisions using historical data.

Also multi-agent systems can make a logical network resource configuration management to comply with changes in user demand. Therefore this kind of controllers improve high bandwidth delay product networks, lossy links, fairness, advantage to short flows, and variable-rate links. Moreover by considering the fairness criterion they use max-min, proportional, and minimum potential delay.

4.3. Distributed Control. A distributed control system (DCS) refers to a control system of a manufacturing system, process or any kind of dynamic system, which the controller elements are not central but are distributed throughout the system that each component controlled by one or more controllers. The entire system is connected by networks for communication and monitoring [44].

DCS is a broad term used in different industries, to monitor and control distributed equipments [45]:

- (i) Electrical power grids and electrical generation plants,
- (ii) environmental control systems,
- (iii) Traffic signals,
- (iv) water management systems,
- (v) oil refining plants,
- (vi) Chemical plants,
- (vii) Pharmaceutical manufacturing,
- (viii) Sensor networks,
- (ix) Dry cargo and bulk oil carrier ships.

It is a system composed of components which are interconnected by communication network, the subsystems are able to cooperate between themselves as well as recognize system failure states. The behavior of this system has features that are appeared in multi-agent systems. Therefore it is a well-defined area to control with multi-agent structures.

In general a DCS consists of four levels which are as follows [46]:

- (i) technological level,
- (ii) supervisory level,
- (iii) information level,
- (iv) management level.

Technological level is represented by controllers, sensors, and specific component of system. All of them are connected with several networks. Supervisory level is a separated level of DCS. Its Main function is interaction between operators and technological process with real conditions. Information level contains information of technological process saved in huge database system. Management level includes tools for OLAP (online analyses processing) which provide final data for top management.

The DCS architecture mentioned above shows that it is necessary to extend intelligence into all levels of DCS. Some principles of DCS that can be related to MAS are described as follows [47]

- (i) First principle in DCS design is called the principle of direct communication of elements and says that technological level should have direct communication with elements on higher levels. This level should be implemented by some local agents in multi-agent systems. They can be information agents or collaborative agents. Each local agent is responsible for stability, robustness and reliability of its local area. They have correlation with each other in a good manner.
- (ii) Gateway principle in supervisory level of DCS. This principle is an extension of the first principle, where connection between technological and supervisory levels is direct, but if supervisory level is in large-scaled, it is efficient to use gateway communication server (gateway). This level can be implemented using an agent, as a supervisory agent and send supervisory signals to local agents for monitoring them.

In the considered DCS, an agent can be represented by intelligent components used within sensors, controllers, and actuators. Thus, in a DCS, multi-agent behavior can appear as follows [47]

- (i) Recognize the system states—change of logical structure, agent's (component's) failure.
- (ii) Ability to apply the reconfiguration mechanism when an agent's failure is occurred.

So this is an area that multi-agent systems can show their potential benefits more than another kind of control engineering applications because of its distributed nature.

4.4. Hybrid Control. Hybrid systems are generally reactive systems that consist of discrete and continuous components [48]. The discrete part of the system makes decision to switch to a set of control rules; however the continuous part works according to that rules. There are a lot of examples such as computers, manufacturing production and power stations which are designed to control and supervise the behavior of the continuous components. Also the applications of hybrid systems are vast, as the most of today's control systems use computers, and even use software to control physical processes. Therefore, it causes a lot of interest in the academic communities and industry [49].

Hybrid systems model the interactions between logical elements and continuous systems. This includes a variety of mathematical and engineering disciplines such as differential geometry, differential and difference equations, optimal control, automata theory, discrete event systems, data structures, and computation [48]. This is a new phenomenon for control engineers and computer scientists.

Hybrid systems are not only hard to model but also hard to analyze and simulate. In fact, a unified theory has not been created yet for them. However, nowadays intelligent solutions are in use to solve the modeling problem of this kind of systems, increasingly.

Because of complexity, problems that have similar requirements have been decomposed and modeled as systems of multiple interacting intelligent agents. Multi-agent systems can develop flexible intelligent behavior and coordination schemes, but their solutions are not easily represented or analyzed mathematically. Then to use both MAS capabilities and mathematical representation/analysis of system, a new agent framework that requires a modification to the basic agent paradigm to make its dynamics accessible for control design and has mentioned features is proposed in [8].

In the modified agent framework, a hybrid control system is embedded in the core of the agent (see, Figure 1). Also, the abstract "agent state" in Figure 1 is replaced with a hybrid automaton and a controller, while the "agent process" retains its intelligence and coordination function. This makes it possible to represent systems with multiple modes and to design suitable control laws that are valid for each of these modes. Moreover, existing mathematical tools (e.g., from the theory of discrete event systems) can be used to describe and analyze systems based on this kind of agent framework. The Hybrid Intelligent Control Agent (HICA) represents this kind of intelligent agent paradigm. HICA is an intelligent agent wrapped around a hybrid control system core. HICA agents are developed to have stable internal dynamics [8].

4.5. Remote Control. In a transmission network many substations and components are unmanned, and therefore must be controlled remotely. Currently this is normally done from a control center, using dedicated network links to the substation. There is the possibility of using mobile agents as a control mechanism to allow users to remotely control the substation plant over a standard IP network. A modern distributed industrial system, such as a power transmission system, consists of many sites distributed throughout a wide area. Each site contains a number of monitoring and control devices which perform various tasks such as condition monitoring, control, and protection. However, substations are often connected by networks with low bandwidth, which makes remote access to these devices for control or monitoring relatively difficult. Therefore, mobile agents are used for remote access to devices [5].

A mobile agent is a computer program, consisting of both code and data that is able to transmit between computers [20]. In addition, in many cases mobile agents provide more gains in performance compared to client-server methods. The agent is given the address of a server agent. When

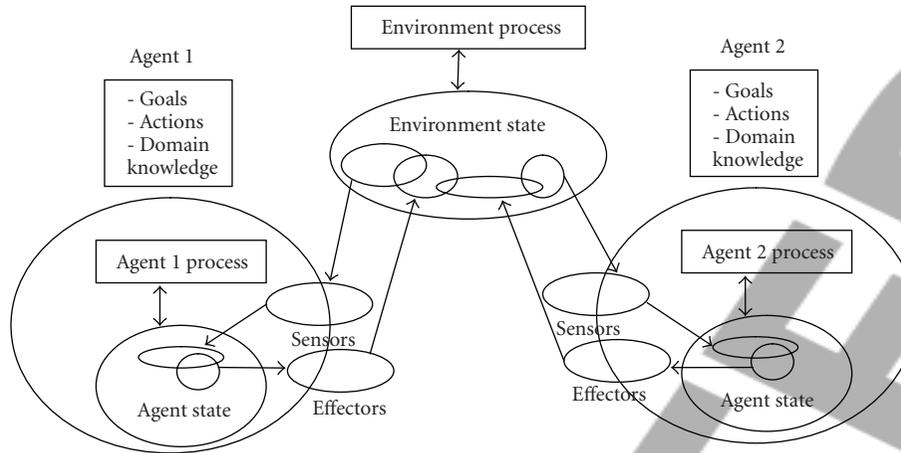


FIGURE 1: Agents in an environment.

the agent starts, it will move to the server agent and carry out the interactions with the device. A sequence of actions represented by FIPA SL [50] is given to the agent. For each of these actions, the agent will either perform it itself by moving to the substation and interacting with the appropriate intelligent electronic device (IED) or IEDs, or will request another agent capable of performing the action to do it [5].

In recent years, using mobile agents in wide area controls to remote monitoring, remote supervisory, remote control and distributed control [51] have been more common and many control applications improve their abilities using mobile agent capabilities in their systems.

4.6. Industrial Control. Industrial control system (ICS) is a general term that encompasses several types of control systems, including supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other smaller control system configurations such as skid-mounted Programmable Logic Controllers (PLC) often found in the industrial sectors and critical infrastructures. ICSs are typically used in industries such as electrical, water, oil and gas, data. Based on information received from remote stations, automated, or operator-driven supervisory commands can be pushed to remote station control devices, which are often referred to as field devices. Field devices control local operations such as opening and closing valves and breakers, collecting data from sensor systems, and monitoring the local environment for alarm conditions [52].

Power system stability is one of the main concerning topics in ICS and has been studied widely from many years ago. Many significant contributions have been made, not only in the direction of analyzing and explaining the dynamic concept, but also to improve the stability of transmission systems. Among these techniques, generator control is one of the most widely applied in the power industry. This typically includes load frequency control (LFC) or automatic generation control (AGC) and excitation controls [53]. LFC/AGC is

a very important issue in power system operation and control for supplying sufficient and reliable electric power [54].

In a power system, sometimes an imbalance between the actual and the scheduled generation power is occurred. This imbalance leads to a frequency error that is the difference between the actual and the synchronous frequency. The magnitude of the frequency error is an indication of how well the power system is capable to balance the actual and the scheduled generation [55]. To reduce the frequency error, different controllers are proposed for the AGC problem. In [56] a centralized controller is designed for the two area system, which requires the knowledge of the whole system. In [57, 58] decentralized controllers for a two area system are proposed. These controllers are designed based on modern control theory, and each area requires knowledge of the other area. If the dimensions of the power system increase, then these controllers may become more complex as the number of the state variables increase significantly.

Moreover, most of the centralized and decentralized controllers are designed for a specific disturbance, and if the nature of the disturbance varies, they may not act as expected. Also, due to the fact that these controllers are designed using a linear model, the inherent nonlinearities of the system are not mentioned and controlled there. Therefore, design of an adaptive controller is an interesting approach.

Multi-agent system with distributed and correlated features can be a good solution for AGC problem in multi areas. Furthermore, using intelligent agents in MAS it can manage inherently nonlinearities aspects that are appeared in reality of power systems. Such a control system used in [59] consists of some online local decentralized controllers supervised by a global controller. The controllers are designed independent of system parameters, and hence will learn to control the areas of the system in response to load variations.

4.7. Multi-Robot Systems Control. Multi-Robot Systems (MRSs) can often be used to fulfill the tasks that are difficult to be accomplished by an individual robot, especially

in the presence of uncertainties, incomplete information, distributed control, and asynchronous computation, and so forth.

Research activity in MRSs has increased substantially in the last few years. Topics include cooperative manipulation [60], multi-robot motion planning, collaborative mapping and exploration [61], software architectures for multi-robotic systems [62], and formation control [63]. Many practical and potential applications, such as unmanned aerial vehicles (UAVs), spacecraft, autonomous underwater vehicles (AUVs), ground mobile robots, and other robot-based applications in hazardous and/or unknown environments can benefit from the use of MRSs. Therefore, MRSs have received considerable attention during the last decade [64–67].

Another significant development of MRS stems from the studies on biological systems or complex models arising in cognitive science and economics (see, e.g., [68]), that most of them deal with a large number of robotic agents and the MRS is thus viewed as a swarm [68, 69], or a colony or, more generally, as a robot collective [70].

Therefore, there have been many challenging issues in MRSs. These challenges often involve the realization of basic behaviors, such as trajectory tracking, formation-keeping control, and collision avoidance, or allocating tasks, communication, coordinating actions, and team reasoning.

One way to resolve the above challenges is gained through cooperative control, since it allows the development of complex behavior based on several controllers combined to achieve the desired result. Moreover the multi-agent systems approach to distributed control systems is widely spread for important utilities it provides. They offer a decentralized control model based on agents. Then integrating these two lines of research (the multi-agent systems and the cooperative control approaches) results in a multi-agent architecture of cooperative controls.

In this approach several elements, namely, agents, cooperate with each other to reach their own goals, and these goals sum up to the final goal of the whole system. Since the multi-agent system is inherently multi-threaded, each agent has its own thread of control; each agent decides whether or not to perform an action on request from another agent (autonomy); and each agent exhibits a reactive, proactive and social behavior (flexibility). In addition, no centralized arbiter is defined, but the coordination of all the agents achieving a common goal (robot goals) is peer to-peer based.

Therefore, the tasks to be accomplished to achieve a global goal are distributed between agents and each agent is to perform its special task (each agent in the system has its own control block).

Also multi-agent systems can help to robot motion planning, robot navigation and cause to system reliability by controlling a group of mobile robots to complete multiple tasks simultaneously. By cooperative and competitive behavior of the agents the group of mobile robot can automatically arrange the total task and dynamically adjust their motion whenever the environment is changed. Also multi-agent systems can control a group of mobile robots to complete multiple tasks at different locations, such that the desired

number of robots will arrive at every target location from arbitrary initial locations. The robot motion planning, such that the robots can start to move before their destinations are finalized and the robot navigation can be dynamically adjusted to guarantee that each target location has the desired number of robots, even under unexpected uncertainties, such as when some robots break down, some robots and/or some tasks are added, or some tasks are changed can manage easily using multi-agent systems too.

4.8. Formation Control. Formation control has become one of the well-known problems in multi-robot systems. Compared with a single mobile robot, many advantages of a network of mobile robots working together have been shown in many tasks, such as object transportation, mobile sensor networks, cooperative classification and surveillance, robotic soccer, and so forth [71].

Formation control is an important technique to achieve cooperative behavior in multi-robot systems. The task is to control a group of mobile robots to follow a predefined path or trajectory, while maintaining a desired formation pattern. Its outcomes offer a wide range of applications, such as security patrols, search and rescue in hazardous environments, area coverage, and reconnaissance in military missions. Formation control does not restrict itself only to ground mobile robots. It can be applied to aircrafts, especially unmanned aerial vehicles (UAVs), spacecrafts, surface vessels, or underwater vehicles [72].

Since using groups of artificial agents could similarly benefit to formation tactics, robotics researchers and those in the artificial life community develop multi-agent systems to formation behaviors for simulated robots [73].

Using multi-agent systems to formation control require individual agents to satisfy their kinematics while constantly satisfying interagent constraints. In typical leader-follower formations, the leader has the responsibility of guiding the group, while the followers have the responsibility of maintaining the interagent formation. Distributing the group control tasks to individual agents must be compatible with the control and sensing capabilities of the individual agents. As the interagent dependencies get more complicated, a systematic framework for controlling formations is vital [74].

One of the applications of formation control using MAS is a framework for formation control of multi-robot systems. In this framework formations can be modeled using *formation graphs* which are graphs whose nodes capture the individual agent kinematics, and whose edges represent interagent constraints that must be satisfied. Two distinct types of formations are considered: *undirected formations* and *directed formations*. In undirected formations each agent is equally responsible for maintaining the formation. For each edge constraining two agents of the formation graph, both agents cooperate in order to satisfy the constraint. Undirected formations, therefore, present a more centralized approach to the formation control problem as communication between agents is, in general, necessary. In directed formations, for each edge constraining two agents, only one of the agents (the follower) is responsible for maintaining the

constraint. Directed formations, therefore, represent a more decentralized solution to the formation control problem.

Another formation strategy for coordinated control of groups of mobile robot is as follow: each mobile robot captures the individual agent that relies only on locally available information, namely, the relative locations of a sensed subgroup of agents. Global information and communication are not required. Instead, in this approach local sensors (perhaps vision) can be used to generate effective global group behavior.

5. Technical Challenges and MAS in Control Engineering

While the benefits of agent technology and some control engineering applications that MAS can be used within have been described, it is important to identify the key technical challenges that should be overcome to allow most effective implementation of multi-agent systems within the control engineering. These include what follows:

- (i) Platforms: a number of multi-agent system platforms exist (e.g., JADE [30, 75], JASON [76]). The necessity to develop agents that can interact with each other is fundamental to the development of flexible, extensible, open architectures. For this reason, platform choice is extremely important.
- (ii) Toolkits: according to the increasing amount of agent research within the control engineering systems, there is the opportunity to reuse agent designs and functionality. Therefore, there are some toolkits [77, 78], which allow the reuse of existing agent behaviors and capabilities.
- (iii) Intelligent agent design: a number of different architectures for intelligent agents can be found in the literature including belief desire and intention agents (BDI) [20], reactive agents [20], agents with layered architectures [20], and agents implemented using model-based programming [79]. Each of these implementation strategies will produce agents with different degrees of reactivity, proactiveness and social abilities. What is not easily understood is how flexible autonomy varies across these implementation strategies and their suitability for different kind of control engineering applications.
- (iv) Agent communication languages and ontologies: foundation of agents' social ability is agent communication languages. These define how agents exchange information, communicate, and negotiate. There are some protocols and content languages within them which allow meaningful messages sending. International standards are set by the Foundation for Intelligent Physical Agents (FIPAs) [31]. An important point of using agent-based technology is that all agents within control engineering applications should be able to cooperate and interoperate. Therefore, the community must agree on appropriate agent communication language standards. This includes

the area of ontologies [80] which define the terms and concepts which agents are able to exchange, interpret, and understand.

- (v) Security: due to the peer-to-peer communication between agents, security can be a key problem. There must be measures to determine the level of trust between agents and the security of messaging [81]. Competitive agents may need more services. Similarly, communication between two agents is open to attacks such as sender spoofing (the message claims that be from a more trusted agent) and message modification (a message is changed while traveling between agents, particularly in negotiation situations).
- (vi) Mobility: a number of researchers are interested in mobile agents, which move completely (source code and data) from machine to machine [30]. While this has been suggested within a few control engineering applications, no credible reason for using this approach is clear. In [82], Pěchouček and Thompson say "People often claim that agent mobility is inevitable and more essential than is actually the case. Often, migration of data or simple communication is sufficient, rather than migration of an agent's code."

Moreover, technical and implementation issues described above, the lack of experiences of using multi-agent system technology in industry is an obvious problem of manufacturers considering MAS solutions. According to Wooldridge and Jennings [83], the migration of an agent system from prototype to a solution that is robust and reliable enough to be used in practice is a nontrivial step. This requires the exhibition of MAS technology in the industrial environment for some applications. Furthermore, there is also a requirement for clear reporting of industrial experiments results of MAS technology, failures and problems as well as successes.

6. Implementation Issues and Further Investigation

6.1. Agents Implementation. None of the design methodologies mentioned above help for selection of a specific pattern of agent implementation as they display the correct levels of reactivity, proactiveness, and social ability. An agent can be conceptualized as a black box which sends and receives messages and interacts with its environment autonomously. However, the functionality of engineering multi-agent systems means that developer need the different agent design options or agent anatomies, and their characteristics so that developers can select the best appropriate agent anatomy related to their applications.

6.1.1. Agent Structures. Various approaches to build autonomous intelligent agents can be found in literature: Belief-Desire-Intention (BDI) agents, reactive agents, agents with layered architectures [20], and agents implemented

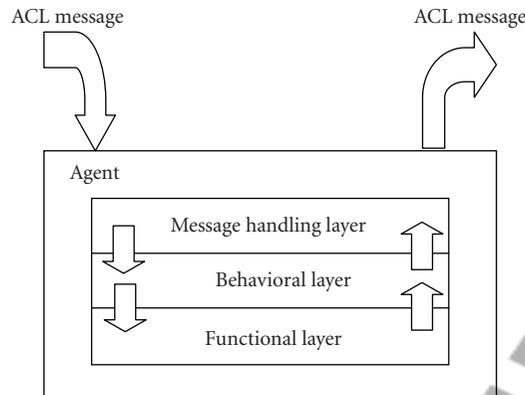


FIGURE 2: Layered architecture employed by JADE agents.

using model-based programming [79]. The BDI approach to building agents is based on mental models of an agent's beliefs, desires, and intentions. It considers agents to have beliefs (about itself, other agents and its environments), desires (about future states), and intentions (about its own future actions) and is particularly useful for

- (i) developing formal models of agents,
- (ii) developing a deep model of agent communication,
- (iii) inferring an agent's internal state from its behavior
Wooldridge lists four characteristics of intelligent agents which naturally fit the purpose and design of the BDI model [20]:
- (iv) situated—they are embedded in their environment,
- (v) goal directed—they have goals that they try to achieve,
- (vi) reactive—they react to changes in their environment,
- (vii) social—they can communicate with other agents (including humans).

Reactive agents are normally associated with the model of intelligence. The fundamental property of reactive agents is that they do not perform reasoning through interaction with environment. Instead they react to inputs from their environment and messages from other agents [20]. They are easy to implement, but the proactiveness of the agents it produces is arguable. Several layered agent anatomies are discussed in [20]. As an example, agents developed for the JADE platform consist of three basic layers [30]: a message handling layer, a behavioral layer, and functional layer (see, Figure 2). The functional layer has the core functional attributes of the agent, that is, the actions the agent can perform. The behavioral layer provides control to when an agent will carry out specific tasks. The behavioral layer will instruct the message handling layer to inform other agents of the new data. Similarly, the action taken by an agent in response to the receipt of a new message is decided in the behavioral layer. The message handling layer is responsible for sending and receiving of messages from other agents, (here the related ACL and ontology parsers are implemented)

as well as the functionality for the control of conversations with other agents [20].

6.1.2. Tools for Agents and MAS Implementation. In recent years, both commercial and open source agent development tools have become available [84]. When implementing a multi-agent system, truly selection of MAS development tools is required. Firstly, the toolset has to agree with the standards that developers want. Secondly, agents implemented using the chosen toolset must display a level of robustness required for the application. JADE [75] has become a favorite by researchers in control engineering in recent years. While JADE supports FIPA standards and agent's robustness make it attractive, it also has a certain style of agent implementation which may not be optimal for exploiting autonomy. Regardless of the agent anatomies, there is an opportunity to reuse agent designs and functionality for other applications in the whole community. Therefore, there is a role for toolkits that allow the reuse of existing agent functions, behaviors, and capabilities tuned for applications to control engineering problems. The extension of ontologies may also help reduce the development costs of multi-agent systems and promote interaction between them.

6.2. Discussions. MAS is now investigated as a new approach for control systems modeling and implementation. This causes a necessary need to various methods of problem analysis and agent-based programming to construct efficient, distributed problem-solving and well-coordinated interactions agents. We may say that there is a need for using of several simultaneous complementary (and sometimes overlapping) models of agency in control engineering applications [85]. Theories behind these models take into account the agents' desired behavior and "subjective" motivations internal to the agents based on their explicit intentions and commitments [86].

Current and future research is (should be) directed towards the using of such models of agency which are responsible for agents (and multi-agent) rationality (logical and economic), sociability, interactivity, and adaptability in real applications. These kinds of models and agencies

features have not been considered in many of MAS control engineering applications, yet. Using the capabilities of MAS (as mentioned earlier) can help to provide more effective problem solutions in these areas.

Also some MAS applications in control engineering need some improvement. In another word, delays will affect on transmission and reception of supervisory and coordination signals in all areas. Therefore, an important extension to some MAS usages in control applications would be to undertake some analytical studies of multi-agent stability in the presence of significant communication delays [8].

7. Conclusion

This paper has defined the fundamental terms and concepts that are related to multi-agent systems, discussed why it is being used for a number of control engineering applications and a survey of recent works has been used to highlight the application areas for which the MAS technology is currently being investigated. It also has discussed how the MAS should be designed and implemented for control engineering applications.

References

- [1] A. W. Colombo, R. Schoop, and R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 1, pp. 322–337, 2006.
- [2] D. Z. Zhang, A. Anosike, and M. K. Lim, "Dynamically integrated manufacturing systems (DIMS)—a multi-agent approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 37, no. 5, pp. 824–850, 2007.
- [3] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, T. Cumming, and I. Watt, "Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 559–567, 2006.
- [4] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The design of a multi-agent transformer condition monitoring system," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1845–1852, 2004.
- [5] D. P. Buse and Q. H. Wu, "Mobile agents for remote control of distributed systems," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1142–1149, 2004.
- [6] J. Zhou, G. Chen, H. Zhang, W. Yan, and Q. Chen, "Multi-agent based distributed monitoring and control of hazard installations," in *Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA '07)*, pp. 2892–2895, 2007.
- [7] J. Galdun, L. Takáč, J. Liguš, J. M. Thirie, and J. Sarnovský, "Distributed control systems reliability: consideration of multi-agent behavior," in *Proceedings of the 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI '08)*, pp. 157–162, 2008.
- [8] K. Fregene, D. C. Kennedy, and D. W. L. Wang, "Toward a systems- and control-oriented agent framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 5, pp. 999–1012, 2005.
- [9] K.-S. Hwang, S.-W. Tan, M.-C. Hsiao, and C.-S. Wu, "Cooperative multi-agent congestion control for high-speed networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 2, pp. 255–268, 2005.
- [10] D. Srinivasan and M. C. Choy, "Cooperative multi-agent system for coordinated traffic signal control," *IEEE Proceedings: Intelligent Transport Systems*, vol. 153, no. 1, pp. 41–50, 2006.
- [11] T. Nagata and H. Sasaki, "A multi-agent approach to power system restoration," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 457–462, 2002.
- [12] S. E. Widergren, J. M. Roop, R. T. Guttromson, and Z. Huang, "Simulating the dynamic coupling of market and physical system operations," in *Proceedings of IEEE Power Engineering Society General Meeting*, vol. 1, pp. 748–753, 2004.
- [13] D. Koesrindartoto, J. Sun, and L. Tesfatsion, "An agent-based computational laboratory for testing the economic reliability of wholesale power market designs," in *Proceedings of IEEE Power Engineering Society General Meeting*, vol. 3, pp. 2818–2823, 2005.
- [14] A. L. Dimeas and N. D. Hatziaargyriou, "Operation of a multi-agent system for microgrid control," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1447–1455, 2005.
- [15] D. P. Buse, P. Sun, Q. H. Wu, and J. Fitch, "Agent-based substation automation," *IEEE Power and Energy Magazine*, vol. 1, no. 2, pp. 50–55, 2003.
- [16] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.
- [17] P. Maes, "Artificial life meets entertainment: lifelike autonomous agents," *Communications of the ACM*, vol. 38, no. 11, pp. 108–114, 1995.
- [18] L. N. Foner, "Entertaining agents: a sociological case study," in *Proceedings of the 1st International Conference on Autonomous Agents*, pp. 122–129, 1997.
- [19] B. Hayes-Röth, "An architecture for adaptive intelligent systems," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 329–365, 1995.
- [20] M. Wooldridge, "Intelligent agents," in *Multi-Agent Systems*, M. Wooldridge and G. Weiss, Eds., pp. 3–51, MIT Press, Cambridge, Mass, USA, 1999.
- [21] S. Franklin and A. Graesser, "Is it an agent or just a program?" in *Proceedings of the 3rd International Workshop Agent Theories, Architectures, and Languages*, Springer, New York, NY, USA, 1996.
- [22] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
- [23] M. Wooldridge, *An Introduction to Multi-agent Systems*, John Wiley & Sons, New York, NY, USA, 2002.
- [24] A. H. Bond and L. Gasser, "An analysis and research in DAI," in *Reading in Distributed Artificial Intelligence*, Morgarn Kuffman, San Mateo, Calif, USA, 1988.
- [25] N. R. Jennings and M. J. Wooldridge, "Application of intelligent agents," in *Agent Technology: Foundations, Applications, and Markets*, N. R. Jennings and M. Wooldridge, Eds., pp. 3–28, Springer, New York, NY, USA, 1998.
- [26] D.-L. Yu, T. K. Chang, and D.-W. Yu, "Fault tolerant control of multivariable processes using auto-tuning PID controller," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 1, pp. 32–43, 2005.
- [27] D. Binkley, M. Ceccato, M. Harman, F. Ricca, and P. Tonella, "Tool-supported refactoring of existing object-oriented code into aspects," *IEEE Transactions on Software Engineering*, vol. 32, no. 9, pp. 698–717, 2006.
- [28] Z. Zhang, J. D. McCalley, V. Vishwanathan, and V. Honavar, "Multi-agent system solutions for distributed computing,

- communications, and data integration needs in the power industry,” in *Proceedings of IEEE Power Engineering Society General Meeting*, vol. 1, pp. 45–49, June 2004.
- [29] R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, *Multi-Agent Programming Languages, Platforms and Applications*, Springer, New York, NY, USA, 2005.
- [30] F. Bellifemine, A. Poggi, and G. Rimassa, “Developing multi-agent systems with JADE,” in *Intelligent Agents VII*, C. Castelfranchi and Y. Lesperance, Eds., vol. 1571 of *Lecture Notes in Artificial Intelligence*, pp. 29–42, Springer, New York, NY, USA, 2001.
- [31] Foundation for Intelligent Physical Agents (FIPA), 2007, <http://www.fipa.org/>.
- [32] A. Ibraheem, P. Kumar, and D. P. Kothari, “Recent philosophies of automatic generation control strategies in power systems,” *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 346–357, 2005.
- [33] Committee on Visionary Manufacturing Challenges, Commission on Engineering and Technical Systems, and National Research Council, *Visionary Manufacturing Challenges for 2020*, National Academy Press, Washington, DC, USA, 1998.
- [34] G. Rathwell, “Design of plant control and information systems within an enterprise architecture,” PERA—Standard documentation ISA S95, 2001.
- [35] T. Williams and H. Li, “PERA and GERAM—enterprise reference architectures in enterprise integration,” in *Proceedings of the 3rd IFIP International Working Conference on the Design of Information Infrastructure Systems for Manufacturing II*, pp. 3–30, Forth Worth, Tex, USA, 1998.
- [36] <http://www.pera.net/>.
- [37] K. Asai and S. Takashima, *Manufacturing Automation Systems and CIM Factories*, Chapman & Hall, London, UK, 1994.
- [38] U. Rembold and B. O. Nnaji, *Computer Integrated Manufacturing and Engineering*, Addison-Wesley, Reading, Mass, USA, 1993.
- [39] A. W. Scheer, *CIM Computer-Integrated-Manufacturing toward the Factory of the Future*, Springer, Berlin, Germany, 1993.
- [40] A. Tharumarajah, A. J. Wells, and L. Nemes, “Comparison of the bionic, fractal and holonic manufacturing system concepts,” *International Journal of Computer Integrated Manufacturing*, vol. 9, no. 3, pp. 217–226, 1996.
- [41] S. C. Lauzon, A. K. L. Ma, J. K. Mills, and B. Benhabib, “Application of discrete-event-system theory to flexible manufacturing,” *IEEE Control Systems Magazine*, vol. 16, no. 1, pp. 41–48, 1996.
- [42] C. Tomlin, G. J. Pappas, and S. Sastry, “Conflict resolution for air traffic management: a study in multi-agent hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.
- [43] S. Kunniyur and R. Srikant, “End-to-end congestion control schemes: utility functions, random losses and ECN marks,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 689–702, 2003.
- [44] B. Bamieh, F. Paganini, and M. A. Dahleh, “Distributed control of spatially invariant systems,” *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1091–1107, 2002.
- [45] http://en.wikipedia.org/wiki/Distributed_control.
- [46] J. Galdun, L. Takáč, J. Liguš, J. M. Thirie, and J. Sarnovský, “Distributed control systems reliability: consideration of multi-agent behavior,” in *Proceedings of the 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI '08)*, pp. 157–162, 2008.
- [47] J. Liguš and J. Horanska, “The principle of direct communication in distributed control system design,” in *Proceedings of the 12th EAEEIE Annual Conference on Innovations in Education for Electrical and Information Engineering (EIE '01)*, Nancy, France, May 2001.
- [48] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, “Verifying hybrid systems,” in *Hybrid Systems*, Z. Manna and A. Pnueli, Eds., Lecture Notes in Computer Science, Springer, Bonn, Germany, 1993.
- [49] Hybrid Control Systems and Optimal Control, 2008, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/ahak/report.html.
- [50] FIPA SL Content Language Specification, Foundation for Intelligent Physical Agents, 2000, <http://www.fipa.org/specs/fipa00008/>.
- [51] L. Zhiyun, M. Broucke, and B. Francis, “Local control strategies for groups of mobile autonomous agents,” *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.
- [52] http://en.wikipedia.org/wiki/Industrial_Control_Systems.
- [53] T. Yong, R. H. Lasseter, and W. Cui, “Coordination of excitation and governing control based on fuzzy logic,” in *Proceedings of IEEE Power Engineering Society General Meeting*, vol. 1, pp. 737–742, New York, NY, USA, 1999.
- [54] M. S. Anower, M. G. Rabbani, M. F. Hossain, M. R. I. Sheikh, and M. Rakibul Islam, “Fuzzy frequency controller for an AGC for the improvement of power system dynamics,” in *Proceedings of the International Conference on Electrical and Computer Engineering (ICECE '06)*, 2006.
- [55] G. Gross and J. W. Lee, “Analysis of load frequency control performance assessment criteria,” *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 520–525, 2001.
- [56] Y. H. Moon and H. S. Ryu, “Optimal tracking approach to load frequency control in power systems,” in *Proceedings of IEEE Power Engineering Society General Meeting*, 2000.
- [57] M. Alrifai and M. Zribi, “A robust decentralized controller for power system load frequency control,” in *Proceedings of the 39th International Universities Power Engineering Conference*, vol. 1, pp. 794–799, 2004.
- [58] A. S. Emarah, “Design of decentralized load frequency controller with limited state feedback,” in *Proceedings of the International Conference on Control*, vol. 2, pp. 899–903, 1991.
- [59] S. Eftekharnejad and A. Feliachi, “Stability enhancement through reinforcement learning: load frequency control case study,” in *Proceedings of iREP Symposium on Bulk Power System Dynamics and Control—VII. Revitalizing Operational Reliability*, Charleston, SC, USA, 2007.
- [60] W. Kang, N. Xi, and A. Sparks, “Formation control of autonomous agents in 3D workspace,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1755–1760, San Francisco, Calif, USA, 2000.
- [61] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 476–481, San Francisco, Calif, USA, 2000.
- [62] L. E. Parker, “Current state of the art in distributed robot systems,” in *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen, Eds., pp. 3–12, Springer, New York, NY, USA, 2000.
- [63] J. P. Desai, J. P. Ostrowski, and V. Kumar, “Controlling formations of multiple mobile robots,” in *Proceedings of IEEE*

- International Conference on Robotics and Automation*, vol. 4, pp. 2864–2869, Leuven, Belgium, May 1998.
- [64] 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, 1998.
- [65] L. Iocchi, D. Nardi, M. Piaggio, and A. Sgorbissa, "Distributed coordination in heterogeneous multi-robot systems," *Autonomous Robots*, vol. 15, no. 2, pp. 155–168, 2003.
- [66] M. J. Matarić, G. S. Sukhatme, and E. H. Østergaard, "Multi-robot task allocation in uncertain environments," *Autonomous Robots*, vol. 14, no. 2-3, pp. 255–263, 2003.
- [67] C. F. Touzet, "Distributed lazy Q-learning for cooperative mobile robots," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 5–13, 2004.
- [68] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–27, 1997.
- [69] L. E. Parker, "ALLIANCE: an architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.
- [70] G. Dudek, M. R. M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.
- [71] A. V. Savkin, "Coordinated collective motion of groups of autonomous mobile robots: analysis of Vicsek's model," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 981–983, 2004.
- [72] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, 2001.
- [73] T. Balch and R. C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1999.
- [74] P. Tabuada, G. J. Pappas, and P. Lima, "Feasible formations of multi-agent systems," in *Proceedings of the American Control Conference*, vol. 1, pp. 56–61, Arlington, Va, USA, June 2001.
- [75] Java Agent Development Framework (JADE), <http://jade.cse.it/>.
- [76] R. H. Bordini, J. F. Hubner, and M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason*, John Wiley & Sons, Chichester, UK, 2007.
- [77] <http://www.webnms.com/javaagent/index.html>.
- [78] <http://sourceforge.net/projects/fipa-os/>.
- [79] B. C. Williams, M. D. Ingham, S. H. Chung, and P. H. Elliott, "Modelbased programming of intelligent embedded systems and robotic space explorers," *Proceedings of IEEE*, vol. 91, no. 1, pp. 212–237, 2003.
- [80] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [81] M. Wooldridge, J. P. Müller, and M. Tambe, "Agent theories, architectures, and languages: a bibliography," in *Proceedings of the Intelligent Agents II: Agent Theories, Architectures, and Languages*, pp. 408–431, 1996.
- [82] M. Pěchouček and S. G. Thompson, "Agents in industry: the best from the AAMAS 2005 industry track," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 86–87, 2006.
- [83] M. Wooldridge and N. R. Jennings, "Pitfalls of agent oriented development," in *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 385–391, 1998.
- [84] E. E. Mangina, "Review of software products for multi-agent systems," <http://www.agentlink.org/index.php>.
- [85] N. R. Jennings, K. Sycara, and M. Wooldridge, "A road map of agent research and development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7–38, 1998.
- [86] E. Oliveira, K. Fischer, and O. Stepankova, "Multi-agent systems: which research for which applications," *Robotics and Autonomous Systems*, vol. 27, no. 1-2, pp. 91–106, 1999.