A Multi-agent Architecture Based Cooperation and Intelligent Decision Making Method for Multirobot Systems

Tao Yang[1], Jia Ma[1], Zeng-Guang Hou[1], Gang Peng[2], and Min Tan[1]

[1] Key Laboratory of Complex Systems and Intelligence Science,
Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China
{yangtao, jia.ma, zengguang.hou, min.tan}@ia.ac.cn
[2] Department of Control Science and Control Engineering,
Huazhong University of Science and Technology, Wuhan 430074, China
penggg@263.net

Abstract. The design of a hybrid multi-agent architecture is proposed for multirobot systems. Analysis of the architecture shows that it is suitable for multirobot systems dealing with changing environments. Meanwhile, it is capable of controlling a group of robots to accomplish multiple tasks simultaneously. Two associated issues about the architecture are cooperation between robots and intelligent decision making. Ability vector, cost function and reward function are used as criteria to describe and solve the role assignment problem in multirobot cooperation. A solution of information fusion based on RBF neural networks is applied to solve the reality problem in decision making of multirobot systems. And an experiment about robot soccer shooting is designed. The experimental results verify that the method can improve the whole decision system in accuracy.

1 Introduction

In many practical applications, a multirobot system is usually faster and more efficiently than a single powerful robot to accomplish complex or heavy tasks. The advantages of multirobot systems are as follows:

An overall task can be separated into several parts which can be executed simultaneously by a robot team. Multiple robots can accomplish performance benefits and are not spatially constrained as a single robot. Compared with developing a versatile robot, multirobot system is actually the combination of lots of robots with various simple functions. So building and using several simple robots can be easier, cheaper, more flexible and more fault tolerant than having a single powerful robot for each separate task [1].

Multirobot systems can improve performance and reliability; however, in multirobot systems the most challenging task is the coordination and cooperation of these robots to satisfactorily perform the overall mission [2]. Many researches have focused on this issue [3-5]. Among them, the method based on multi-agent system can give us a good way to solve the problem.

M. Ishikawa et al. (Eds.): ICONIP 2007, Part II, LNCS 4985, pp. 376–385, 2008.

The multi-agent system (MAS) is an emerging subfield of artificial intelligence (AI) and is one of the two sub-disciplines of distributed artificial intelligence (DAI) [6]. It tries to provide principles for construction of complex system, involving multiple agents and mechanisms for coordination of independent agents' behaviors [2].

An efficient intelligent control structure of MAS is the foundation for multi-robot systems to handle uncertainty and complexity and achieve the goal in the dynamic environments. The major structures proposed by many researches [7-9] can be categorized into two general types: hierarchical structures and behavioral structures.

In a hierarchical structure, information flows from sensors to high-level decision units in a fixed way, then the decision units send commands to low-level actuator units. Agents in this structure are cognitive, but the structure has poor flexibility. So it is difficult to adapt to modern robotic systems. In a behavioral structure, control problem is broken into behaviors without any central intelligent agent present [1]. So high-level decisions are usually difficult to achieve. A hybrid structure which combines hierarchical structure and behavioral structure can be designed to get rid of drawbacks associated with the above two and can help to develop practical and powerful multirobot systems.

From the standpoint of MAS, an individual robot with the abilities of domain knowledge, action selection and communication with others is considered an agent in multirobot systems. A system made up of this kind of robots can be treated as a MAS. Robot soccer system is a good example of multirobot systems. Soccer robots must work together (cooperation). They play the game in unpredictable conditions. Also they decide which actions to be selected in order to put the ball in opponent's goal. As what is mentioned above, the robot soccer system is always discussed as a test benchmark for MAS. In this paper, robot shooting decision is considered as a test bed for the hybrid MAS architecture.

This paper is organized as follows. Architecture of MAS is described in detail in Section 2. Section 3 presents function of cooperation module in this architecture. Function of another module, decision making module, is presented in Section 4. And a shooting decision in robot soccer system is designed to verify the effectiveness of the module. Concluding remarks are given in Section 5.

2 The Proposed Architecture

Fig. 1 shows the basic diagram of MAS architecture for multirobot systems. As mentioned in section 1, it is a hybrid architecture, including high-level deliberative agents and low-level reactive agents. What's more, in terms of [1], the agents in a MAS may have homogeneous or heterogeneous structures. To most multirobot systems, agents must have different goals, knowledge and actions, they receive different sensory inputs, meanwhile, they have to cooperate each other in order to accomplish some complex tasks. So the architecture must be heterogeneous, which is composed of various agents.

Agents are classified as three types: host agent, logic agent and physical agent in [10]. The classification meets the specification of real multirobot systems. The MAS architecture presented in this section also consists of master agent and real robot which is the combination of reasoning agent and actuator agent.

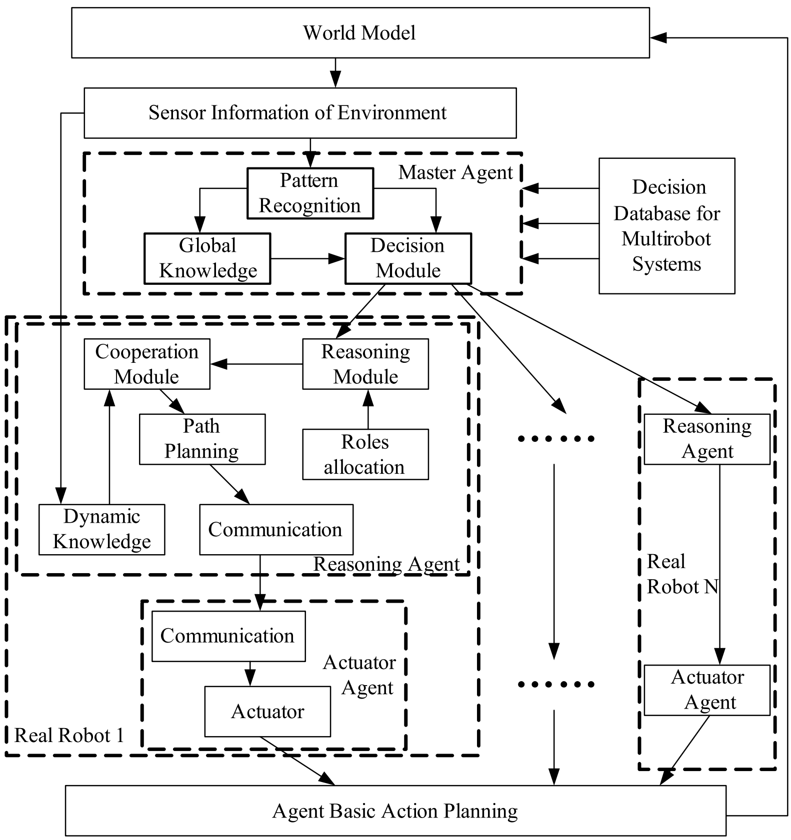


Fig. 1. A hybrid MAS architecture for multirobot systems

Master agent consists of strategies knowledge in global knowledge database, static knowledge and rational rules. Reasoning agent consists of dynamic knowledge database, reasoning and path planning. The components of dynamic knowledge include characteristics of robots and the objectives to be achieved. Each robot has its own reasoning agent, which can decide the path planning and share useful information with other robots. Actuator agent refers to mechanical and electrical devices of a robot. It can receive commands sent to the actuators of the robot and execute basic movement behaviors.

The architecture is a kind of reconstruction of multi-agent logic structure. It is not a straightforward compounding-form based on hierarchical and behavioral structures, but a joint intension framework of hierarchical reasoning and shared plan. The advantages are as follows:

(1) Robustness and fault tolerance. According to this architecture, breakdown of an individual robot will have little effect on the whole team, because of the existence of master agent; that is, the master agent has the ability to reallocate new roles to other reasoning agents and reconstruct team work.

(2) Real-time reactive ability. The architecture is a model based on knowledge and planning, which combines deliberative agents and reactive agents. On one hand, Agents behave more like they are thinking, by searching through a space of knowledge stored before, making a decision about action selection and predicting the effects of actions. On the other hand, Agents can simply retrieve present behaviors similar to reflexes without maintaining any rational reasoning. So multirobot systems based on this parallel distributed mechanism can fulfill the requirements of dynamic, complex and unpredictable environments.

(3) Flexibility. Agents with global or dynamic knowledge database can learn from experiences and learn from each other, so as to adapt themselves to dynamic environments. If a new agent can help to achieve the goal, it will be authorized to join the team by master agent. Accordingly the scale of teamwork can be enlarged.

(4) Simplification of reasoning process. Because it is not an efficient method to change strategies frequently for a certain task, design of master agents can be simplified. Reasoning agents become important parts of decision making system. The problem of collision between robots should be solved by means of negotiation. So the reasoning ability improves a lot.

As a result, the makeup of the hybrid architecture helps to coordinate planning activities with real-time reactive behaviors to deal with dynamic environments.

When the architecture is applied to real multirobot systems, there are several important functions that need to be performed. The associated issues are cooperation and intelligent decision making. Details of the two issues will be discussed in next two sections.

3 Role Assignment of Multirobot Systems

When multirobot systems accomplish a task by means of cooperation, how to assign roles of robots properly is a challenging problem. In order to implement team work, dynamic role assignment is acquired according to various robots' states.

Now, "ability vector" is introduced to describe whether a robot is able to accomplish its task. Generally speaking, a robot has various abilities including sensory and executive abilities. To a task, only when robots' abilities meet with it, can the objective be achieved.

"Ability set" C is defined, which is made up of unitary ability c_i , $1 \le i \le n$.

The ability T_j to accomplish a certain task is a linear combination of unitary ability c_i:

$$\hat{T}_j = \sum_{i=1}^{n} t_{ji} \cdot c_i, \quad j \in N, \quad t_{ji} \ge 0. \tag{1}$$

where t_{ji} is the weight value of c_i.

Correspondingly, R_j is used to describe the robot's ability.

$$R_j = \sum_{i=1}^{n} r_{ji} \cdot c_i, \quad j \in N, \quad r_{ji} \geq 0. \tag{2}$$

where r_{ji} is the weight value of c_i.

If the robot is competent for the task, $R_j \geq T_j$.

A task at different stages requires different abilities; that is, to fulfill the task, r_{ji} should change to correspond to the change of t_{ji}

Ability vector A_t is defined to describe various required abilities for the whole task.

$$A_t = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1j} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2j} & \cdots & t_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{m1} & t_{m2} & \cdots & t_{mj} & \cdots & t_{mn} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \tag{3}$$

where, $t_{ij} \geq 0$. When the task doesn't require the unitary ability c_i, $t_{ij} = 0$

And correspondingly, A_r describes various abilities of a robot to the task.

$$A_r = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1j} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2j} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mj} & \cdots & r_{mn} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \tag{4}$$

where, $r_{ij} \geq 0$. When the robot doesn't have the unitary ability c_i, $r_{ij} = 0$

So if a robot is fully qualified to the task, $A_r(i) \geq A_t(i), \quad i = 1, \cdots, m$

Cost function $f(\text{cost})$ is defined to represent the cost with which a robot is capable of accomplishing a task, for example, spending a period of time and consuming a quantity of energy. After task accomplished, a robot will be rewarded. Reward function $f(\text{rewd})$ is defined to represent the reward. So according to equations (1)-(4), we can get the benefit from these two functions:

$$b_i = \begin{cases} f_i(\text{rewd}) - f_i(\text{cost}), & \text{if } A_r(i) \geq A_t(i) \text{ and } f_i(\text{rewd}) \geq f_i(\text{cost}) \\ 0 & \text{else} \end{cases} \tag{5}$$

By adopting this form of description, a robot's ability and a task can be described in detail. And the role assignment can be implemented in terms of maximizing the benefit, which is calculated by the specific design of cost function and reward function. Elements, for example, the distance from robot to ball and the distance between two robots, etc. have been taken into account for soccer robots cooperation in [11].

4 Intelligent Decision Making

In multirobot systems based on MAS, each robot is autonomous; that is, it can make decision independently by global or local information. But as what is mentioned in most robotics domains, sensors are noisy, action must be selected under time pressure. An effective decision making method is in great demand. As the tasks and environments become increasingly complex, decision making system can help a group of robots to coordinate their limited physical and computational resources effectively, and ensure that the robots will achieve their complex tasks in dynamic environments.

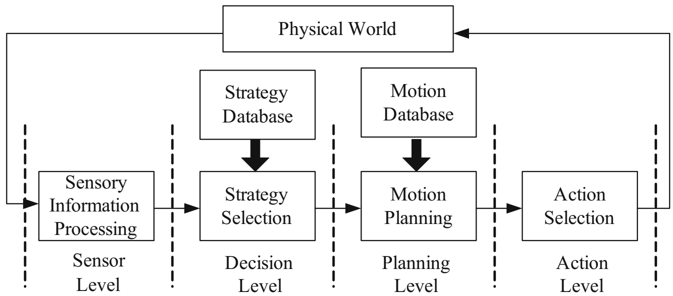


Fig. 2. A structure of decision making system

As what is mentioned in most researches, for example [12] [13], effective structures of decision making systems are almost all hierarchical. The structure of a decision making system is shown in Fig. 2, which consists of sensor level, decision level, planning level and action level. As a result, this hierarchical structure improves the efficiency and robustness of the robot decision.

Conventional decision algorithms always rely on the above decision structure presented in Fig. 2. While in most multirobot systems, reasoning methods often fail to handle large quantities of complex domains. Robots must have the abilities to learn from experiences as opposed to existing "If- Then" rules.

4.1 Information Fusion and Neural Networks

An information fusion decision method based on radial basis function neural networks (RBFNN) is proposed to solve the problem of learning from experiences. A typical example of multirobot systems, robot soccer, is used as test bed to verify the efficiency of the method.

Obviously, there are various sources of sensory information received by multirobot systems, for example ultra-sonic, laser-ranger, vision, etc. To robot soccer, raw information which can directly be obtained include coordinates of teammates, opponents (robots) and the ball; moving directions of robots; velocities of robots and the ball; predicted positions of robots; distances and angles between robots. Resources of information fusion made up of these data are the basis of next step decision.

Here a three-layered parallel information fusion structure is adopted for the decision system, which is proposed by Thomopoulos [14]. The parallel structure is constituted with sensor layer, reasoning layer and decision layer as is presented

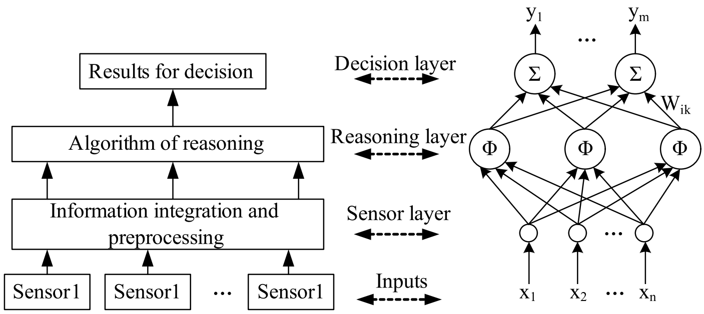


Fig. 3. A layered structure of information fusion and a feedforward neural networks

in the left side of Fig. 3. From the standpoint of information fusion and neural networks, the layered structure and function of each layer in information fusion totally correspond with those of neural networks. Fig. 3 shows the correspondence between information fusion structure and a feedforward neural networks with a single hidden layer.

An intelligent decision system in robot soccer usually involves huge state spaces, RBFNN poses as an attractive method for the task.

4.2 A Brief Introduction to RBFNN

The architecture of the RBFNN is presented in the right side of Fig. 3. The network contains a single hidden layer of neurons which are completely linked to input and output layers. The output of the RBFNN is calculated according to [15]:

$$y_i = f_i(x) = \sum_{k=1}^{N} w_{ik}\phi_k(x, c_k) = \sum_{k=1}^{N} w_{ik}\phi_k(\|x - c_k\|_2), \quad i = 1, 2, \cdots, m \qquad (6)$$

where $x \in R^n$ is an input vector, $\phi_k(\cdot)$ is the activation function of hidden layer, $\|\cdot\|_2$ denotes the Euclidean norm, w_{ik} is the weight from the hidden layer to output layer, N is the number of neurons in hidden layer, and $c_k \in R^n$ is the radial basis function (RBF) center of neuron in the input vector space.

The form of activation function $\phi_k(\cdot)$ in the hidden layer is nonlinear function with radial symmetry. In practical applications, the most widely used RBF is the Gaussian kernel function as the functional form: $\phi(x) = \exp\left[-\frac{(x - c_k)^T(x - c_k)}{2\sigma^2}\right]$, where parameter σ is the radius that controls the "width" of RBF [15]. Detailed training algorithm for a RBFNN is also described in [15].

4.3 An Experiment in Robot Soccer System

In a robot soccer system, in order to win the game robots must ceaselessly make decisions, for example interception, obstacle avoidance, cooperation each other,

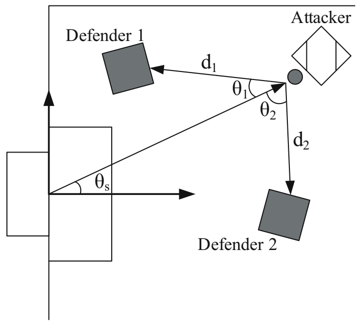


Fig. 4. Resources of information in robot soccer game

etc. Among them shooting is one of the important decisions. When an attacker is facing two defenders, an experiment about how to decide shooting angles is design to verify the effectiveness of the decision method mentioned above.

The following important information should be taken into account in the game. (1) Real-time data $\{x_i, y_i, \theta_i\}$, where (x_i, y_i) denote the current positions of robots and ball, θ_i denote the direction angle of robots. (2) Prediction data $\{x_i, y_i, \theta_i\}$, which represent the next positions and direction angles of robots and ball. (3) Command data $\{v_l, v_r, x, y, \theta\}$, v_l, v_r are the command of wheels' velocity, (x, y, θ) is the desired position and direction. Under the shooting condition, the data above should be preprocessed so as to be used to make decision.

The input vector to the network consists of four components $\langle d_1, \theta_{d1}, d_2, \theta_{d2} \rangle$ shown in Fig. 4. d_1, d_2 are distances between ball and two defenders respectively; θ_{d1}, θ_{d2} are angles between ball and two defenders respectively. The output to the network is θ_s, the desired angle of shooting.

Training set is necessary for the training of neural networks. A software "referee" manages the beginning and the end of training. Data for training are put into database respectively according to success and failure. Attacker is directed to shoot, while two defenders try to intercept the ball. Only if ball is put into the goal is the shooting successful and vice versa.

The position of attacker should be initialized stochastically, which is between 1 and 1.5 meters from goal. And two defenders are situated randomly between goal and attacker. The steps to be followed to obtain the training data are described below:

Step 1: Shooting angle is set to θ_s;

Step 2: Defenders rotate to face the ball if the distance between ball and goal is greater than 1 meter;

Step 3: If the distance between ball and goal is less than 1 meter, five components $\langle \theta_s, d_1, \theta_{d1}, d_2, \theta_{d2} \rangle$ are recorded and defenders rotate a random angle A between $-45°$ and $45°$;

Step 4: After Step 3, defenders try to intercept the ball;

Step 5: If shooting is successful (interception is failure), the experimental data will be sent to database, otherwise return to step 1.

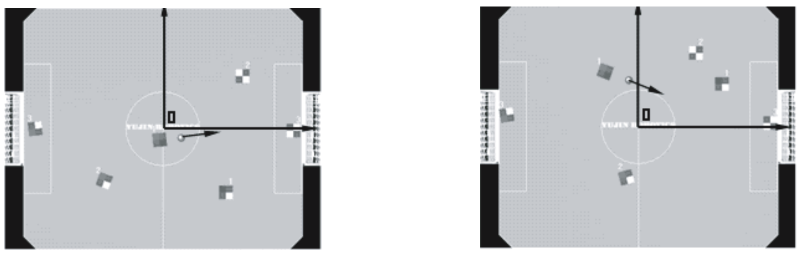
The above method is to obtain 597 successful training data from 1000 experiments (the rate of success is 59.7%) then the RBFNN trained by these data can be used to make decision of shooting in online robot soccer games.

To demonstrate the effectiveness of the decision method based on RBFNN, several combinations of $\langle \theta_s, d_1, \theta_{d1}, d_2, \theta_{d2} \rangle$ are input to the trained networks. As a result, there are 224 successful scoring in 300 experiments, which is much better than conventional methods.

Illustrations are shown in Fig. 5 when attacker selects a successful shooting angle. Where,

(a) Defenders are in different phases of coordinate.

(b) Defenders are in the same phase of coordinate.



(a) (b)

Fig. 5. Experimental results of soccer robot shooting decision

5 Conclusions and Future Work

A hybrid architecture of MAS, the role assignment method for cooperation and decision making based on RBFNN are proposed for multirobot systems. The architecture is composed of master agent and real robot that consists of reasoning agent and actuator agent. The favorable features of the architecture are as follows: **(1)** robustness and fault tolerance; **(2)** real-time reactive ability; **(3)** flexibility; **(4)** simplification of reasoning process. So the architecture, which is a combination of hierarchical and behavioral structures can meet the design specification of multirobot systems. Ability vector is used to describe the abilities of a robot and the abilities required for accomplishing a task. According to benefits calculated from reward function and cost function, role assignment can be implemented in an efficient way for cooperation between robots. Compared with conventional methods of decision making, a solution of decision based on RBFNN is more effective to improve the whole decision system. Results of shooting experiment in robot soccer game verify the efficiency and the effectiveness of the method.

The architecture and related issues are put forward to study the multirobot systems. In the opinion of the paper, the architecture should be further improved and be the basis for future research in the evaluation of multirobot systems.

References

1. Kim, J.H., Vadakkepat, P.: Multi-agent systems: a survey from the robot-soccer perspective. Intelligent Automation and Soft Computing 1, 3–17 (2000)
2. Zhu, A., Yang, S.X.: A som-based multi-agent architecture for multirobot systems. International Journal of Robotics and Automation 21, 92–99 (2006)
3. Parker, L.E., Emmons, B.A.: Cooperative multi-robot observation of multiple moving targets. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2082–2089 (1997)
4. Kube, C.R., Bonabeau, E.: Cooperative transport by ants and robots. Robotics and Autonomous Systems 30, 85–101 (2000)
5. Cao, Y., Fukunaga, A.S., Kahng, A.B.: Cooperative mobile robotics: Antecedents and Directions. Autonomous Robots 4, 1–23 (1997)
6. Stone, P., Veloso, M.: Multiagent systems: a survey from a machine learning perspective. Autonomous Robotics 8, 1–57 (2000)
7. Parker, L.E.: ALLIANCE: an architecture for fault tolerant multirobot cooperation. IEEE Transactions on Robotics and Automation 14, 220–240 (1998)
8. Yen, J., Yin, J.W., et al.: CAST: collaborative agents for simulating teamwork. In: The Seventeenth International Joint Conference on Artificial Intelligence, pp. 1135–1144 (2001)
9. Tambe, M.: Towards flexible teamwork. Journal of Artificial Intelligence Research 7, 83–124 (1997)
10. Tang, P., Yang, Y.M.: Study on multi-agent system and the structure of soccer game system. Journal of Guangdong University of Technology 18, 1–4 (2001)
11. Stone, P.: Layered learning in multi-agent systems. PhD. thesis. School of Computer Science Carnegie Mellon University Pittsburgh (1998)
12. Brumitt, B.L., Stentz, A.: Dynamic mission planning for multiple mobile robots. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, pp. 2396–2401 (1996)
13. Stone, P., Veloso, M.: Task decomposition, dynamic role assignment, and low bandwidth communication for real-time strategic teamwork. Artificial Intelligence 110, 241–273 (1999)
14. Thomopoulos, S.: Senor integration and data fusion. Journal of Robotic Systems 33, 337–372 (1990)
15. Ham, F.M., Kostanic, I.: Principles of neurocomputing for science and engineering. McGraw-Hill Science/Engineering/Math., pp. 140–162 (2000)