J. Indian Inst. Sci., May–June 1990, 70, 297–303.

BOOK REVIEWS

A robot ping-pong player experiment in real-time intelligent control by Russel L. Anderson. The MIT Press, 55, Hayward Street, Cambridge, Massachusetts, 02142, USA, 1989, pp. 275, $35. Indian orders to: Affiliated East-West Press Pvt. Ltd., 25, Dr. Muniappa Road, Kilpauk, Madras 600 010.

Robotics is a challenging discipline which is a frontier area at present. A large number of books have been written on robotics. But almost all of those deal with the conventional topics such as Kinematics, Dynamics, Control, Robot Vision to take care of conventional operations such as pick and place, continuous path control and object recognition for possible assembly operations.

The present book deals with a new topic 'Ping-Pong (Table-Tennis) Playing Robot' and is a documentation of the experiments in real-time intelligent control. In order to enable the robot play ping-pong, the author creates an 'expert controller' which is a combination of a specialized expert system and a robot controller. Also studied in the book are working sensors, low-level controller and actuator components.

Chapter 1 of the book deals with introduction of robots and their limitations, introduction to robot ping-pong, why ping-pong is a good problem for developing intelligent robot control methods; a preview of experimental work which is the subject matter of the book and talks about application areas.

Chapter 2 introduces the international standard robot ping-pong rules proposed by John Billingsley (details given in Appendix 1 of the book).

The game has been modified to be playable by moderate-sized immobile robots by scaling down and restricting the area that must be covered by robots. To compensate for this, the paddle is restricted to be only 12.5 cm in diameter whereas people use 15–17 cm paddles presenting 60% more area. A ping-pong ball is 4 cm in diameter.

The robot ping-pong table is shown in fig. 4. The wire frames provided at the ends and at the middle limit the maximum ball speed to approximately 10m/sec (for anything but an ideal player the limit is about 8m/sec). The table's geometry results in maximum available reaction times of 0.4 to 0.8 sec. Approximately 0.1 sec is taken by the camera for accumulating enough frames to get a rudimentary estimate of ball trajectory. Robot motion time can range under 0.3 sec or more, typically 0.5 sec. Robot ping-pong does require fast robots.

In this chapter the author gives a comparison to human ping-pong by considering return trajectory, ball speed, choice of contact point, spin estimation and paddle trajectory. In the same chapter the author discusses the ping-pong physics by considering aerodynamics of the ping-pong ball, bounce of the ball on the table, and the method of hitting the ball.

Chapter 3 is devoted to system design requirements and approaches. Cameras feed to vision processers. Data are sent to trajectory analyzer (TRANAL) which produces predictions of the ball's path for the expert controller processor (STRIKER). A kinematics and dynamics slave processor (K/D) and

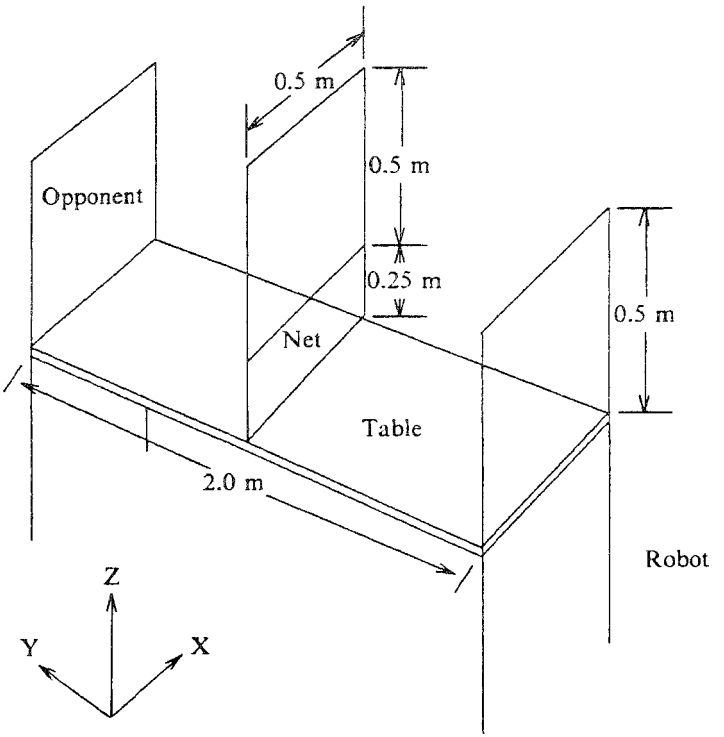

FIG. 4. Robot Ping-Pong Table. The players, background, and table are black. The ball must travel sequentially through the square frame at one end, through the center frame, then bounce once and travel through the frame at the table's other end.

two joint servo processors drive the robot. Also provided to the system are real-time debugger, a sequencer, a strategy analyzer and a video tape system driver.

Chapter 4 gives details about a real-time stereo vision system which can provide accurate position data. The author has developed detailed model of the sensor and processing system which are essential to high performance.

Chapter 5 gives an account of the robot controller and describes the robot's mechanical hardware, electronics hardware and software, architecture which are helpful in understanding the experimental

work reported in the book The computational building blocks used in the work include kinematics, gravity loads, Jacobian, inverse Jacobian, inertias, dynamics and motor model. A detailed account of trajectory generation and trajectory following is also given.

While Chapter 6 gives the preliminaries of the expert controller and discusses program architecture and (symbolic and numeric) data architecture, Chapter 7 describes the design of real-time expert controller. The controller processes a continuous stream of sensor data. During the initial planning process, based on limited amount of data available, 'good guesses' are made. During the next stage, the plan is modified in response to changing data.

Chapter 8 contains a description of the application of the expert controller to robot ping-pong. Several case studies are given. Based on the case studies about experiments in real-time intelligent control the author concludes: "The robot can play a reasonable game against human opponents. The machine keeps score, making mistakes only when particularly bizarre situations occur. Some mistakes could be eliminated by minor additions to its score-keeping analysis, but others would take much more substantial analysis using all the available data, especially the ball's path after an event on which a point must be assessed. Ultimately, we observe that human players do not always agree. Human players find the robot's small table and high net challenging at first, then gradually adapt.... The bounce prediction is probably the single element that most degrades the system's performance at the current time.... The longest volley we have observed was 21 strokes apiece by the human and the machine. The performance was as much a consequence of the human's lack of mistakes as anything else." The author also discusses implementation limitations in this chapter. Chapter 9 of the book is devoted to conclusions. One of the interesting conclusions made by the author is: The continuous self-perception of the robot's motion and the critical evaluation of that motion's quality seems much closer to the human experience than the blind plan/more cycles of today's robots. The author outlines further research topics and discusses what can happen if the robot plays with a human on a full size table.

This book is a documentation of an outstanding work involving challenging experiments in real-time intelligent control of a robot ping-pong player. It contains details to motivate and inspire the reader take up similar challenging tasks, and will be useful to practising engineering and research workers. This is simply an outstanding book about an outstanding experiment.

Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012.

M. R. CHIDAMBARA

Performance and evaluation of LISP systems by Richard P. Gabriel. The MIT Press, 55, Hayward Street, Cambridge, MA 02142, USA, 1985, pp. 285, $25.88. Indian orders to: Affiliated East-West Press Pvt. Ltd., 25, Dr. Muniappa Road, Kilpauk, Madras 600010.

A long-felt need of the community of LISP users has been an evaluation of various LISP systems. This book, which is the culmination of a study, conducted by the author over a period of three years, on the performance of various LISP systems, fulfils this need. A valuable handbook to LISP users, it describes in detail, LISP implementations currently in actual use, along with benchmarking techniques and performance results for all major LISP dialects in use.

The book provides a good set of benchmark programs to study the efficiency of LISP systems. These programs have been so chosen that different aspects of commonly used LISP constructs can be tested. Further, these benchmark programs (numbering 20) have been run on 12 different LISP

implementations. For each implementation, aspects that are important to performance, like data types, function calls, generic arithmetic, compiler, data representation, etc., have been described in detail. The role of benchmarking—"to measure various dimensions of LISP system performance and to order those systems along each of these dimensions"—has been very well brought out.

Performance and evaluation is made up of three parts, each corresponding to a chapter. The introductory chapter provides the necessary background by outlining the various issues involved in the performance evaluation of a LISP system. In the second chapter, various LISP implementations *viz.*, MacLISP, MitCADR, Symbolics, LMI Lambda, S-1 LISP, Fnanz LISP, NIL, Spice LISP, Vax common LISP, Portable standard LISP, Xerox D-Machine and Data General Common LISP. The final part, which runs to over 200 pages of the total 285 pages, presents details of the 20 benchmark programs. The LISP code for each benchmark, along with the description of what each benchmark does, is also provided. This enables one to run tests on one's own system.

The author should be complimented for the thorough and methodical way of presentation. This book can form a valuable reference for all users and students of LISP. A periodic update of the book is highly desirable.

Department of Electrical Engineering
Indian Institute of Science
Bangalore 560012.

K. R. RAMAKRISHNAN

Networks and distributed computation: concepts, tools and algorithms (Computer systems series) by Micheal Raynal. The MIT Press, 55, Hayward Street, Cambridge, Mass. 02142, USA, 1987, pp. 166, $29.50. Indian orders to: Affiliated East-West Press Pvt. Ltd., 25, Dr. Muniappa Road, Kilpauk, Madras 600010.

The book outlines some of the recent developments in the area of distributed computing systems. It is concerned with the concepts, tools, and algorithms that make up the foundations of the distributed applications and systems. The book consists of three parts, each handling a different aspect.

The first part gives an informal introduction to the problems posed by distributed computer systems and networks. Two simple examples of distributed problems are described and studied. The first one is the producer-consumer problem. The producer and consumer cannot communicate *via* a shared memory in a distributed system. Instead, they communicate using message transmission and reception primitives, which operate on communication channels connecting the sites of the producer and consumer. Various issues in solving the message-loss problem are discussed. The second problem discussed is the clock problem, which generalizes the producer-consumer problem. A number of algorithmic tools appropriate for solving these problems are then introduced.

The second part of the book tackles two basic problems which, in general, are necessary for design and realization of distributed applications and systems. These concern a certain number of simple algorithms allowing traversal through a network, learning of the structure, determination of the routes, etc. These algorithms concern the networks part of the applications and make it possible to relax the initial conditions, *e.g.* the need for knowledge of the number of sites. The second aspect is concerned with the system aspect of the applications. For a given site, it implies the detection of a global state allowing the site to take a decision or verify a property. This part illustrates the fact that depending on the system or application implemented, how the above two aspects must be resolved and solved appropriately.

The last part of the book deals with the knowledge required for distributed processing which can define the underlying distributed kernel, the suitable algorithmic tools, and their properties. These two aspects are considered keeping the global synchronization in mind; the protocols necessary for their implementation are examples of distributed control. Starting with the idea of toolbox, elements that make the solution of problems easier are introduced and analysed. Their use and the effects of their properties are illustrated with the algorithms for distributed control and computation.

On the whole, the book assimilates the general concepts, methods and mechanisms that are crucial to the design and implementation of distributed systems. The book does not emphasize any specific system or protocol, hence it is general from this point of view. Up-to-date literature in the subject is included. The book will be of immense use for researchers specialising in the area of distributed computing.

Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012.

L. M. PATNAIK

Performance models of multiprocessor systems by M. Ajmone Marsan, G. Balbo, and G. Conte. The MIT Press, 55, Hayward Street, Cambridge, Mass. 02142, USA, 1986, pp. 280, $35. Indian orders to: Affiliated East-West Press, 25, Dr. Muniappa Road, Kilpauk, Madras 600010, India.

In the design and evaluation of multiprocessor systems, performance modelling assumes great importance. This book is a valuable and timely contribution to the study of the performance issues in multiprocessors using analytical modelling tools such as Markov and semi-Markov processes, queues, queueing networks, and stochastic Petri nets. The authors are well known for their research publications in this area and this improves the credibility of this book. The contents of the book are current up to the date of publication and should prove immensely useful to researchers and practitioners in the area of multiprocessor design.

The book is in two parts: Part 1 (Chapters 2–4) provides an overview of the modelling tools employed in this book while Part 2 (Chapters 5–11) deals with the performance modelling of several different types of multiprocessor architectures. The inclusion of Part 1 has certainly made the book self-contained.

The first chapter (Introduction) is a quick, crisp statement of the purpose of the book. After motivating the need for performance evaluation of distributed systems in general and multiprocessors in particular, the authors present the differences between measurement and modelling and between simulation modelling and analytical modelling.

Chapters 2, 3 and 4 which constitute Part I of the book deal with overview material on stochastic processes, queueing models, and stochastic Petri nets, respectively. The important topics discussed in Chapter 2 are discrete-time Markov chains, continuous-time Markov chains, aggregation in Markov chains, and semi-Markov processes. From the viewpoint of basic concepts, a warm-up discussion of the memoryless random variables—geometric and exponential—would have made this chapter complete. Also as an illustrative example, a simple multiprocessor example would have been congenial instead of the weather model for the discrete-time Markov chain. Chapter 3 is very well organized though the material presented (on queueing models) is voluminous. Here again, an example or two from simple multiprocessor systems could have provided a beautiful setting for the rest of the book.

Chapter 4, which deals with stochastic Petri nets (SPNs), mainly discusses the generalized SPNs (GSPNs) proposed by the authors in their paper in *ACM Transactions on Computer Systems* in 1984. In fact, much of the material in this chapter, is culled out from the above classic paper.

Part II starts with Chapter 5 which comprises a brief survey of different types of multiprocessor architectures and the modelling assumptions that are relevant in multiprocessor modelling. Chapters 6 to 10 deal with the modelling of five different architecture types: crossbar multiprocessors (Chapter 6), single-bus multiprocessors with external common memory (Chapter 7), multiple-bus multiprocessors with external common memory, single-bus multiprocessors with distributed common memory (Chapter 9), and multiple-bus multiprocessors with distributed common memory (Chapter 10). The models presented in Chapters 6 and 7 are mostly queueing-based. The material in Chapter 6 does not include any numerical results unlike the following chapters. The organization of these chapters more or less follows the same pattern of several models followed by numerical results. The titles of the subsections in Chapters 9 and 10 are not indicative of the contents therein. The final chapter, Chapter 11, entitled 'Other aspects of multiprocessor performance evaluation', presents the modelling of synchronization of tasks, failure of system components, and prototype measurements and model validation.

While the book itself is a mature and authentic contribution, the inclusion of certain important topics would have enriched it a great deal and made it complete and comprehensive. The first of these topics is Reliability and performability of (fault-tolerant) multiprocessors which has received tremendous attention in recent times. Interestingly, the same modelling tools as discussed in this book have been used in the literature for evaluation of reliability and performability. The second important topic is the modelling of workload which has been discussed albeit briefly in a short subsection (5.4.2). A separate chapter, characterizing workload in multiprocessors, would have done justice to the importance of this topic. The third important topic concerns the use of other important SPN proposals such as Extended SPNs (Duke University), Generalized Timed Petri nets (University of Wisconsin, Madison), and Stochastic Activity Networks (University of Michigan, Ann Arbor) which have not even been mentioned in the book.

On the whole, this book is a timely offering in a very important area and should prove a useful source of considerable value to researchers in this area. It is highly recommended also for researchers in the general areas of performance evaluation and parallel computer architectures.

Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012.

Y. NARAHARI

ACTORS: A model of concurrent computation in distributed systems by Gul Agha. The MIT Press, 55, Hayward Street, Cambridge, Mass. 02142, USA, 1986, pp. 144, $25. Indian orders to: Affiliated East-West Press Pvt Ltd, 25, Dr. Muniyappa Road, Kilpauk, Madras 600010.

Artificial intelligence (AI) involves the study of humanlike intelligence through the ideas and methods of computation. The first-generation AI systems drew their inspiration from and are generally implemented on stand-alone sequential computers. Intelligent behaviour in such systems results from a global knowledge base acquired from the problem domain which is often assumed to be complete and consistent. With the evolution of the notion of distributed AI, the restriction to sequential computing is unrealistic and there is need to explore the suitability of concurrent/parallel computing in distributed systems for AI tasks.

This brings us to the research direction explored in this book, namely, the development of a suitable programming language for concurrent computing *i.e.* potentially parallel execution of desired actions in a distributed system environment. Here we have a large number of interconnected processors that can communicate and solve problems in a cooperative manner.

The computational elements considered in this book are called ACTORS which are more powerful than sequential processes and value-transforming functional systems, such as the ones used in dataflow computing. An actor receives communication and maps each incoming communication to a 3-tuple consisting of: 1. a finite set of communications sent to other actors, 2. a new behaviour, governing its response to the next communication processed, and 3. a finite set of new actors created.

Each actor has a local time and the concept of a unique global time is not meaningful. The use of actor model results in a system that is reconfigurable and extensible.

The book then proceeds to examine the structure of computation in the actor paradigm. A program in an actor language consists of behaviour definitions, new expressions which create actors, send commands which create tasks, a receptionist declaration which lists actors that may receive communications from outside and an external declaration listing actors that are not part of the population defined by the program but from whom communications may be sent from within the configuration.

Syntax for two minimal actor languages SAL (similar to ALGOL) and Act (similar to LISP) is defined. Subsequent chapters develop an operational semantics for actors and difficulties such as divergence and deadlocks arising while exploiting concurrency. Finally, a paradigm for addressing problems in distributed computing that is suitable for computing in open systems is presented. The actor event diagrams introduced by the author are a particularly good technique for illustrating the interactions in asynchronous concurrent systems. In summary, this is a brilliant book which substantially advances the understanding of concurrent/parallel computing and is very useful to students of architecture, programming languages and artificial intelligence.

Deptt of Computer Science and Automation,
Indian Institute of Science
Bangalore 560 012, India.

V. V. S. SARMA