points of the classical control theory which is based on the frequency domain description of the plant, so that the $H$-infinity control theory has been actively studied in the recent years. It is the design method in the frequency domain based on the modern control theory. However, the $H$-infinity theory at the present time lacks the identification of the frequency domain. The topics treated here are, therefore, very interesting for the users intending to apply the $H$-infinity control theory, too. The topics also include an application to test civil engineering structures.

All chapters contain the sections of 'Outline and Learning Objectives', 'Summary', 'Problems', and 'Notes and References', so that one can read this book very easily even without being acquainted with self-tuning systems. A variety of examples in the book may help the reader's understanding of the self-tuning systems. Now, open the jewelry box with your own hand.

Temporal Logic for Real-time Systems*

Jonathan S. Ostroff

Reviewers: L. MOTUS
estonian Academy of Sciences, Institute of Cybernetics, Akademia tee 21, Tallinn 200108, Estonia.

The first book in this series elaborates the fair transition system idea (proposed by computing scientists), extends its application domain to the case of metric time, and suggests a new formalism—real-time temporal logic—for analysing the state sequences generated by the instantiations of the proposed extended fair transition system.

The application domain of the presented methods is real-time control, more specifically, control of discrete event dynamic systems (e.g., flexible manufacturing, sequencing control and, of course, computer software). The author of this book follows the increasingly popular trend of blending different formalisms. This book mixes successfully logical models of discrete event dynamic systems (e.g., temporal logic and finite state machines) with algebraic models (i.e., communicating sequential processes)—the taxonomy of models is given by Ho in his plenary paper at the 11th IFAC World Congress in 1990.

An excellent feature of this book is a nice linking, explained in a few words, of many different techniques used in computing science so that a control engineer can see the connections between different techniques and methods. As an example, just note how elegantly the interconnections are explained between fair transition systems (proposed by Pnueli and Manna), the UNITY approach to concurrent programs (proposed by Chandy and Misra) state machines, communicating sequential processes and temporal logic. It is true, though, that a fair amount of background knowledge is needed to be able fully to enjoy the author's style.

In the following the book is reviewed, chapter by chapter, subjective merits and demerits are listed for each chapter.

Chapter 1, the Introduction, defines the basic notions used in the book. The author does not use the term discrete event dynamic systems, instead he uses real-time discrete event processes. Another definition that I am not very happy with is real-time—I cannot agree that real time is just a metric time. This certainly is one of its possible interpretations.

About the reviewer
Seiichi Shin received degrees of Bachelor, Master, and Doctor of Engineering all from University of Tokyo, in 1978, 1980, and 1987, respectively. From 1980 to 1988, he worked at the University of Tokyo, as an associate researcher in the former, and as an assistant professor in the latter. He became an associate professor of Institute of Information Sciences and Electronics, University of Tsukuba in 1988. In April 1992, he moved to the Department of Mathematical Engineering and Information Physics, University of Tokyo.

His main interest is research on control theory and its application. It includes adaptive control, time delay systems, nonlinear systems, neural network, and distributed control systems. He is a member of IEEE, SICE, ISCIE. IEE of Japan, Japan SIAM, and JSME.

are in this book described by extended state machines (ESMs) which are a mixture of modified finite state machines, channel concepts of CSP (communicating sequential processes) language and the method of partial weakest preconditions/strongest postconditions.

Chapter 2 is ESM Syntax which gives more details about extended state machines which are obtained from finite automata by:

- Extending the type list of state variables (activity variable (next state variable), data variable and time variable are added).

- Adding typical CSP (Communicating Sequential Processes) language properties for describing communication and synchronized actions of separate extended state machines (ESMs).

The ESM syntax is illustrated in two examples—the two trains sharing a track, and transforming a piece of CONIC (a language for distributed computer system description) code to ESM notation.

The ESM is based on the interleaving model of concurrency and assumes "...a precise behavioural description of all crucial features including plant dynamics, concurrency, nondeterminism, synchronization and real-time". This chapter leaves me with a feeling that the suggested methodology is much better suited for code verification than for system specification.

Chapter 3, the ESM semantics introduces the notion of state sequences and trajectories of the extended state machine. This is important since the set of legal trajectories is the control systems description used for further study by temporal logic. When reading this chapter I could not avoid uncomfortable feeling that weak fairness and time bounds on the state-to-state transitions may cause a new symbiosis that cannot be called weak fairness any more—a transition cannot be continuously enabled under the time constraints. May be one should consider "a bounded-delay" fairness in such case. This problem is not studied in the book.

Chapter 4, Real-time Temporal Logic gives a brief overview of Manna–Pnueli temporal logic. The Ostroff's real-time temporal logic (RTTL) is then obtained by extending a definition of formula in Manna–Pnueli temporal logic by adding specific ESM variables. Three typical problems inexorable in ESM/RTTL framework are:

- Verification problem of checking whether the generated trajectories satisfy the given requirements specification.

- Synthesis problem of finding a controller's algorithm provided the plant has been described as an ESM and requirements specification is given.