Foreword

The importance of fault-tolerance and reliability issues in real-time computer control systems might easily be appreciated in the context of the ever increasing use of computers in application areas such as control of hazardous chemical plants and nuclear reactors in process industry, battle management and weapon delivery in defence, intensive care and diagnostic systems in health care and control systems for air and high-speed ground transportation. The use of computers in such systems, for fault-detection and diagnosis, and system reconfiguration, has the potential of dramatically improving the operational effectiveness of real-time systems. The computer system being the principal component of monitoring and control equipment, its failure could result in disastrous consequences, and hence, such a system should be installed only after adequate demonstration of its required level of reliability.

Real-time computer control systems have three major constituents: the physical plant, the computer system and the instrumentation system that interfaces the plant with the computer. The human operator also plays a crucial role especially in emergencies. Equipment failures, malfunctions of sensors, actuators, computer hardware and software, and operator lapses may cause major damage to the system, endanger human life and may turn the environment toxic. Systematic design of reliable computer control systems is therefore an important and a very challenging task. The system has to maintain optimal performance during normal operation, and must also cope with randomly occurring emergencies during which the plant conditions are hostile, by taking corrective actions with strict real-time deadlines.

A preliminary failure cause-consequence analysis should be performed for the computer controlled system to identify the potential hazards associated with failures in each of the subsystems and the human operator. Such an analysis would reveal the criticality of each of the faults. Hazard analysis techniques based on Failure modes and effects analysis, Event trees, Fault trees, Digraphs, and Cause-consequence diagrams are useful for this purpose. A fault-diagnostic system is then designed to detect, diagnose and compensate for these failures and is implemented via software along with other monitoring and control functions. Expert systems, Kalman filters, observers, parity space techniques, fault trees, detection filters etc. are used in the design of the fault diagnostic system. These algorithms are implemented using fault-tolerant hardware and software. The design of fault-tolerant systems is thus a complex problem requiring expertise from a variety of disciplines.

In this special issue, we concentrate on the reliability and fault-tolerance issues in real-time computer systems. We organise the papers in four sections:

(i) Fault-tolerant Software
(ii) Fault-tolerant Computer Architectures
(iii) Performance Modelling of Fault-tolerant Systems
(iv) Applications

1. Fault-tolerant software

In real-time systems, software is the key to the performance and error-free operation of the system. Real-time software is a special class of software with certain unique characteristics, such as operation in unpredictable and asynchronous environments and response generation according to strict deadline schedules.

Reliability, safety and fault-tolerance are desirable features of real-time software. Reliability is the probability that the system will perform its intended function for a specified period of time under a set of specified environmental conditions. Safety is the probability that conditions leading to an accident do not occur whether or not the intended function is performed. In general, reliability requirements are concerned with making the system failure-free whereas safety requirements are concerned with making it accident-free. Fault-tolerance is the survival attribute of the real-time software system. The software should provide correct results in the face of various failures. A major technological concern for the coming years is the ever widening gap between the demand for high quality, robust software and its supply. This is of utmost relevance in the Indian context since there are concerted efforts underway to produce control software for life critical systems including satellite launch vehicles, high-tech combat aircraft, C^3 systems, nuclear power plants and hazardous chemical processes.

There are four papers in this issue which focus on reliable and fault-tolerant software. The first paper by Shrivastava is a didactic exposition on the various issues in the design and implementation of fault-tolerant software. He presents a methodology for constructing software modules that can tolerate both expected and unexpected faults including design faults. Following this, the design of fault-tolerant algorithms – algorithms that incorporate software techniques for tolerating hardware faults – is discussed. The use of these techniques is illustrated in replicated distributed processing and for constructing robust distributed programs.

The survey paper on software dependability by Sarma presents a case for developing a unified framework for dependability. Dependability is a generic concept that has attracted wide attention recently and subsumes various quality factors such as reliability, availability, maintainability, complexity and safety. The paper also surveys the models and methods for software reliability which is the best-known dependability measure and discusses the important notion of software fault-tolerance.

Database consistency is a fundamental requirement of a database system. The paper by Bhargava & Lilien brings out comprehensively the role of fault-tolerant software in maintaining database consistency in the presence of faults and restoring consistency after site crashes and network partitionings. Besides, the paper also discusses the verification of integrity assertions which is fundamental for ensuring the semantic integrity of a database.

The fourth paper in this section, by Patnaik & Balaji, is a survey paper on an important class of fault-tolerant distributed computing systems. These systems are designed to tolerate Byzantine faults which could correspond to any arbitrary behaviour on the part of hardware or software components of the system. The authors discuss agreement problems, agreement protocols, and their applications, in the context of Byzantine-resilient systems.

2. Fault-tolerant computer architectures

Fault-tolerance is achieved in computer systems by introducing redundant or spare processors and memory elements, and capabilities for automatic fault-diagnosis recovery and reconfiguration. With the rapid progress made in VLSI technology and in semiconductor memories, high performance processor and memory units are available at low cost. Distributed computer systems which are interconnections of high performance processors, memory elements and I/O units by means of a communication network have natural fault-tolerance attributes and, by adding capabilities of fault-diagnosis and reconfiguration, they can be made ultra-reliable.

Distributed computing systems are broadly divided into multiprocessor and multicomputer systems. A multiprocessor system consists of a number of processing elements connected to a number of memory modules through an interconnection network. Three types of interconnection topologies have been proposed in the literature. They include crossbar, multistage interconnection networks and multiple bus organizations. In multicomputer systems, each processor has its own memory and inter-processor communication is achieved by a message/packet switching protocol. Several structures including loops, trees, full connections and hypercubes have been proposed. Studies relating to performance and reliability computations are needed to evaluate these distributed computer architectures.

In this volume, we have four papers dealing with fault-tolerant computer architectures. Biswas & Srinivas present a review of various approaches toward tolerating hardware faults in multiprocessor systems. A survey of various models, techniques and methods for fault diagnosis is provided in this paper. Reconfigurable architectures and fault-tolerant VLSI processor arrays are also considered. Raghavendra & Anujan Varma consider reliability and fault-tolerance issues in the design and analysis of multistage interconnection networks (MIN) for multiprocessors. They consider multistage networks which are typically built for N inputs and N outputs using 2×2 switching elements and $\log_2 N$ stages. Further, several approaches for achieving fault-tolerance in MIN are discussed and methods for reliability analysis of MIN are explained.

Reliability and fault-tolerance evaluation of multiprocessor and multicomputer architectures with emphasis on graceful degradation is considered by Das & Bhuyan. They define two measures, reliability and performance availability, to characterize and evaluate multiprocessor and multicomputer architectures. Bandwidth availability and computation communication availability are used to quantify performance availability of multiprocessors and multicomputers. To evaluate the reliability and performance availability, they describe two models: a bus-oriented model and a switch-oriented model. The former is useful for crossbar and multiple bus multiprocessors and the latter for all types of multiprocessors.

Reliability calculation in large computer networks is an issue that abounds in computational problems and, in general, the problem complexity is exponential. Aggarwal presents a high-speed approximate method for reliability analysis of computer communication networks using clustering methods. He also defines a reliability index, an approximate measure of the overall reliability of the system, that can be easily incorporated into reliable system design.

3. Performance modelling of fault-tolerant systems

One of the very important questions that arises when considering fault-tolerant systems is whether the system would perform the intended function at the specified levels of performance. Such a quantitative performance evaluation is required at each stage of the system life cycle: while evaluating alternative designs, while verifying a particular prototype, while guiding redesign and when the system is in operation.

Three methods are available for fault-tolerant system performance: measurement, simulation performance modelling and analytic performance modelling. Performance measurement is possible once a system is built, has been instrumented and is in operation. There are three major drawbacks of this method: first, performance figures relate to the specific system architecture under its current work load, second, measurement is not feasible during the design and development stages of the system, and third, measuring performance in a complex system environment is tedious and costly.

The most attractive approach to fault-tolerant system performance evaluation is through modelling. Fault-tolerant systems consist of a set of redundant resources: data, hardware and software elements and a set of randomly arriving tasks compete for these resources. The resources are prone to random errors and failures. Thus fault-tolerant systems are discrete-event dynamical systems where events occur at random time instants and the performance measures of interest include: resource utilization, contention for resources, response times, performance degradation, degree of fault-tolerance etc.

The tools of discrete-event simulation could be employed for simulation performance modelling of discrete-event dynamical systems. Here simulation models are driven by random input sequences and produce random output sequences. Statistical output analysis is required to interpret results of simulation models. Analytical models of discrete-event dynamical systems include Markov chains, queueing networks and stochastic Petri nets. All the analytical techniques lead to large-size models and their solution requires computer-aided analysis. User friendly computer-aided simulation and analytical performance modelling techniques would help the designer to concentrate on higher level decision-making rather than get bogged down with myriad computational issues.

In this special issue, we have three papers on performance modelling of fault-tolerant computer systems. Narayan Bhat & Kavi provide a critical overview of the approaches to reliability modelling. They show that Petri nets and dataflow graphs facilitate reliability analysis of complex systems. Narahari & Viswanadham present the performance evaluation of a fault-tolerant real-time multiprocessor (FTMP) using stochastic Petri nets. They develop four such models featuring various degrees of FTMP details and compute various performance measures including bus contention, processor utilization and waiting times. Trivedi & Dugan discuss the seven major issues in computer-aided reliability modelling and analysis of complex fault-tolerant systems and discuss the recent progress made in each of these seven areas.

4. Applications

We have three application-oriented papers in this issue, the first two dealing with spacecraft on-board fault-tolerant computers and spacecraft fault-tolerant control systems and the third one with nuclear reactor safety issues. The first paper in this section, by Basu *et al*, describes the on-board fault-tolerant computer system for ISRO's Augmented Satellite Launch Vehicle. They describe the architectural attributes of the on-board computer and the details of software testing carried out in order to ensure reliable operation.

The second paper in this section deals with another important application area–spacecraft control systems. Spacecraft have to function continuously without interruptions and without maintenance for periods of 7–15 years. The spacecraft control system has to detect, diagnose and estimate failures in various components and reconfigure the control system. This fault-tolerance feature has to be incorporated keeping in view the limitations on weight, power and computational facilities. Murugesan & Goel present a brief description of the attitude control system and highlight essential features of the fault-tolerant control system. They also present algorithms for fault detection, identification and reconfiguration for various elements of the spacecraft control system.

Safety in nuclear power plants is an important and widely discussed issue. The Three-mile Island and Chernobyl accidents and their consequences have brought to focus the deficiencies in the current safety control systems. Sri Ram & Iyer discuss safety issues in the CANDU type of nuclear reactors. They review the recent work on station blackout, operational transients, and small and large break loss of coolant accidents. They also stress on the nuclear safety culture to be practised by the operators in all operating nuclear power plants.

Taken together, the fourteen representative papers of the issue help the reader to obtain a global view of the design of real-time systems with specifications on reliability and fault-tolerance. I hope that this special issue will stimulate further interest in this area leading to more reliable and safer real-time systems.

I would like to express my sincere thanks to
— the authors for their enthusiastic response to my invitation,
— the reviewers for their help in providing me with prompt and critical reviews,
— Mr Y Narahari, for his invaluable and cheerful help in a variety of ways,
— Prof. R Narasimha, Chairman, Editorial Board, Sādhanā for his constant help and encouragement.
— Ms K Shashikala, for editorial help.

October 1987

N VISWANADHAM
Guest Editor