Guest Editor’s Introduction

Software for Industrial Process Control

Rudolph J. Lauber, University of Stuttgart

Twenty years ago, digital computer technology in the form of centralized digital control systems caused the first fundamental change in industrial process control. Unfortunately, these systems, which partly replaced analog controllers, had a serious disadvantage. All control functions were dependent on one device, and back-up controls—either provision of analog standby or duplication of the digital computer systems—were necessary. Consequently, complexity increased while reliability decreased.

Today, powerful 16-bit and 32-bit microcomputers are changing the structure of control systems. Together with a new way of developing control systems, this structure is forcing us to redefine the tasks of control engineers, who must now become real-time software engineers as well.

The structure of direct digital control systems is becoming more decentralized: Modern industrial control systems are using distributed microcomputer systems, connected via bus systems, allowing us to achieve the benefits of direct digital control without the disadvantages of centralized control. The first article in this special issue, by James Schoeffler, examines the alternative organization of modules into distributed system architectures, and
discusses communication networks based on the data highway concept, as well as software organization.

The conventional way of developing industrial control systems—"by hand," using mostly pencil and paper and perhaps a desk calculator—is giving way to computer-aided engineering. Software tools used with computer workstations are being applied extensively in all phases of industrial projects and for all kinds of engineering activities—ones that include the design of mathematical models and control algorithms and encompass software and hardware development.  

Because of the advanced information processing capabilities of modern microcomputers and because of the ease in using software tools, highly sophisticated control algorithms (such as adaptive and hierarchical control) are becoming feasible, replacing to some extent the conventional algorithms. The next two articles in this issue are devoted to the computer-aided development of control systems. Claudio Walter addresses how to translate the control requirements of the industrial process into control software specification and design, and Bruce Weide el et al. discuss the integration of software tools and the support of design methodologies for real-time process control systems.

Future industrial control systems will be completely digital, with control functions being realized by software. Consequently, software for industrial process control will have a number of specific characteristics that control engineers will have to cope with, including real-time requirements, reliability and safety requirements, and management requirements.

Real-time requirements. When a computer is controlling a technical process, its software must interact with the dynamic properties of the industrial system and must react to stochastically occurring events within the industrial system. It is this interaction between control computer software and the technical process that leads to two major real-time requirements:

- Requirements to perform control actions at a certain point in time. The following example illustrates this using PEARL (Process and Experiment Automation Real-Time Language):

  AT 11:45 EVERY 10 SEC DURING 2 MIN ACTIVATE MEASUREMENT PRIORITY 2;  
  [Activate the task MEASUREMENT at 11:45 a.m. and repeat this task activation every 10 seconds during a time interval of two minutes with the priority 2]

- Requirements to perform control actions dependent on stochastically occurring events within the process plant (such as to handle a pressure surge in a chemical reactor); for example, in PEARL:

  WHEN PRESSURE ACTIVATE VALVE;  
  [If the interrupt signal designated by PRESSURE occurs, activate the task VALVE].

With both types of requirements, we must have a certain control software reaction time as well as the synchronization of parallel tasks within the software system. The fourth article, by Hartwig Steusloff, discusses the requirements for advanced real-time computer programming languages for distributed industrial process control, explaining the realization aspects when the real-time language PEARL is used in industrial applications.

Reliability and safety requirements. Hidden errors in the software may cause wrong output signals of the control computer (wrong with respect to the value of the output signal or wrong with respect to the point in time, when the output signal is produced). Wrong control signals, then, may result in major damages to an industrial system or even in dangers to the lives of those who come in contact with safety-critical systems (for example, nuclear reactors system or an aircraft control systems). We can cope with these problems in two ways. One is to use fault-tolerant software, while the other is to produce safe soft-
ware without any errors. The next article, by Nancy Leveson, defines software safety and outlines some principles for building safe control systems.

Management requirements. Software management includes project planning, project control, and quality and configuration control of the software products. Because real-time software is so complex, conventional methods are proving inadequate in many projects. Therefore, computer-aided management support tools are being used to overcome the problems. In the last article on software for industrial process control, Yoshihiro Matsumoto overviews managerial issues experienced in a large software factory that manufactures embedded software for real-time process control systems.

Acknowledgments

I thank all persons who helped make this special issue possible, including the authors, the referees, and C. V. Ramamoorthy who stimulated this task. Many thanks also to R. Baumann who conceived the drawing on p. 7.

References


Rudolph J. Lauber is a professor of electrical engineering and director of the Institute of Control Engineering and Process Automation at the University of Stuttgart, Germany, where he has taught since 1970. Earlier, he worked in industry in technical and management positions. He has published over 100 papers and two books, dealing with control theory, industrial computer applications, real-time computer languages, reliability and safety of computer control, and specification and design techniques. He is currently involved with research on computer-aided engineering environments including knowledge-based expert system techniques. Educated at the Technical University of Stuttgart, he received a PhD in 1962. He is a member of the German engineering and computer societies VDE, GMR, and GI, as well as of the IEEE Computer Society and the ACM.

Lauber’s address is Institut für Regelungstechnik und Prozessautomatisierung, 7000 Stuttgart 80, Pfaffenwaldring 47, Postfach 801140, West Germany.