Meeting the Challenge of Computer Science in the Industrial Applications of Control: An Introductory Discussion to the Special Issue

Albert Benveniste, *Fellow, IEEE,* and Karl J. Åström, *Fellow, IEEE*

Abstract—**This introductory paper presents the project, discusses its motivations and objectives, summarizes the results and draws some conclusions. A brief presentation of the papers is also given. Since this is a joint issue with *Automatica*, short presentations of the papers that appear in that journal are also given.**

I. The Project and its Motivations

DEVELOPMENT of large real-time control and signal processing systems is one of the most challenging tasks facing the engineering community. This involves a combination of hardware, software, and intelligence (in particular algorithms). System integration requires advanced project management techniques. The motivation for the project were indications that:

- Industry sometimes does not effectively exploit recent advances in mathematical techniques for control and signal processing to enhance real-time systems with intelligence.
- The academic control community sometimes does not understand how the integrated development of hardware, software, and intelligence, is performed in the industry. Potentially fruitful interactions between the control and computer science communities in industry and academia are not properly handled today.

To face this dual challenge, the *IEEE Control Systems Society* and *IFAC Committee on Theory* launched a joint project entitled **Facing the Challenge of Computer Science in the Industrial Applications of Control**. The objectives of the project are the following:

- To understand how hardware, software, and intelligence, are combined in the integrated development of large real-time systems, and to understand how algorithms for such systems are designed;
- To evaluate the relative merits of obtaining intelligence by collecting human expertise, or by using mathematical models and relying upon mathematical techniques of control or signal processing; and
- To suggest a reorganization of the traditional academic landscape of control and computer sciences for the benefit of a more unified science of real-time and dynamical systems.

Results of the project are:

- Selected reports from meetings held with industrialists on the topic of the project ([2] [3]);
- A brief report [1] summarizing some recommendations to the academic control community, industry, and the funding agencies;
- This special issue and the companion issue in *Automatica*.

Before going further we would like to emphasize that our definition of the term "*control industries*" is a much wider notion than that generally referred to. In referring to "control industries" in the large, we have in mind industrial activities where concepts, notions, objects, and methods of control play an important role: such industrial activities handle dynamical systems, they may use models of different kinds, and perform signal, data, and information processing as well as control and decision making. Consequently, the application areas are diverse and numerous. Furthermore, the objects of these industrial activities range from very small ones (e.g., in consumer electronics) up to huge systems. Apart from variations of the system size there are also large variations in the system life-times. Some systems are expected to be in operation for tens of years. This means that systems undergo several revisions that span over several generations of hardware and software.

II. Facts from Control Industries

In this section we analyze various technical activities that control industries are involved in. Then we discuss their balance in terms of costs, efforts, and difficulty. This analysis is largely based upon the material of [2], [3] and the papers of this special issue.

Manuscript received February 23, 1993. This work was performed as part of the IEEE-IFAC joint project "*Facing the Challenge of Computer Science in the Industrial Applications of Control,*" supported by both the IFAC Technical Committee on Theory and the IEEE-CSS Society; participants of this project were K. J. Åström, A. Benveniste (Chairman), P. E. Caines, G. Cohen, and L. Ljung.
A. Benveniste is with INRIA-IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France.
K. J. Åström is with Department of Automatic Control, Lund Institute of Technology, P.O. Box 118, S-221 00 Lund, Sweden.
IEEE Log Number 9210268.

0018-9286/93$03.00

A. System Development in Control Industries

System development typically involves modeling, algorithm design, development of software and hardware, simulation, testing, installation and operation, see [8].

Modeling: is an important aspect of many phases of system development. The models can be normal dynamic models of systems, but modeling also more generally consists of collecting information on the system in consideration [3], [2]. Although CACSD tools have significantly reduced the cost of designing control and signal processing algorithms, there is much less support for the modeling activity. Hence, we have identified a significant need for tools that support all aspects of modeling. Simulation is an activity that is closely related to modeling. A solid background in control and in the considered application area is also required for the modeling. Collecting knowledge, or data, does not necessarily result in developing models, however [3] [17]; such a situation is especially encountered in large and complex systems [6], see however [15] for reports on a significant modeling effort in power industry.

Algorithm Design: We shall not elaborate too much on this point, since it is well known in our community, and several papers of this special issue report on it [8]–[13], [14], [16]. While algorithm design is often not considered as a major source of cost [2] [5], good design of algorithms is often critical [10] and helps in reducing the cost of application software development—mathematics result in a more compact code than heuristics do, see [2], meeting with GEC-Alsthom. An extreme case of heavy investment in algorithms to reduce the cost of software is [14] and is typical of what is considered as a feature of the control industry in the former Soviet Union.

Software Development: What software consists of is listed in the following few "equations:"

$$\begin{aligned}
\text{Total Software} &= \text{Application Software} \\
&\quad + \text{Basic Software} \\
\text{Application Software} &\equiv \text{modeling} && (1)\\
&\quad + \text{algorithms} && (2)\\
&\quad + \text{heuristics} && (3)\\
\text{Basic Software} &\equiv \text{scientific or other libraries} && (4)\\
&\quad + \text{operating systems and} \\
&\qquad \text{compilers} && (5)\\
&\quad + \text{communication software} && (6)\\
&\quad + \text{user interface software} && (7)
\end{aligned}$$

These equations reveal that algorithm design is only one aspect of software development. Furthermore, there is a tendency that (7) becomes increasingly important. In application software, (3) is the "nonmathematical" part: rules, case-by-case actions to perform, etc. Such features may also occur in (1), namely via knowledge bases. In contrast, basic (i.e., application independent) software mainly consists of items relying on a pure computer science background, with the noticeable exception of (4), which can still be viewed as basic software since it is generally application independent.

Hardware Development: We do not consider the hardware required for building the plant or the process, but only the computer hardware. Depending on security, speed, memory, volume, or other performance criteria when necessary, hardware developments fall into one of the following two broad categories:

- Configuration of an architecture from available components, with emphasis on safety, flexibility, and low cost. See [2], report on the visit to Siemens-AG, and [8].
- Custom or semicustom architecture developments with emphasis on speed, size, or power consumption, and on cost; typical instances are found in automotive, telecommunications, or military industries, and particularly in consumer electronics.

On the other hand, so-called modern *intelligent sensors* now call for hardware directly attached to them in order to support sophisticated data processing and algorithms.

B. Relative Costs for Each Item

The relative costs or efforts for the different phases in system development naturally depend on the particular application. Some approximate figures from ([2], [1]) are given below:

- software cost > hardware cost: relative costs typically range from 50%–50% (in some military or telecommunication subsystems) to 90%–10% (in low performance systems, or in very large systems);
- application software > basic software: typical figures range from 60%–40% to 80%–20%;
- heuristics $\geq$ or even $\gg$ mathematics: even if the mathematical part may also be critical, the heuristic part often dominates by one or two orders of magnitude in terms of cost and manpower [5].

Such figures may be drastically different in the case of consumer electronics, with much higher hardware and replication costs.

C. Software Development: A Primary Concern in Control Industries.

This fact is pointed out in both the reports on meetings with industrialists ([2], meetings with Thomson-CSF and GEC-Alsthom, and [3], meetings with EDF), and in several papers of this special issue [17], [5]. Furthermore, this is also expressed in the way financial efforts are balanced within the Computer Integrated Manufacturing chapter of European ESPRIT program: most of the budget is spent on basic software techniques (data bases, knowledge bases, expert systems), or even on management organization, and very few is spent on modeling and mathematical techniques of control. In general, we have noticed that control industries pay a significant attention to software factories and software development methods. It is also interesting to observe that the development effort is shifting from hardware to software over time. Different industries are in different stages in this development.

III. A Somewhat Deeper Analysis

The purpose of this section is to go beyond the above dry facts, and try to explore potential alternatives for reducing costs in control industries at those points where they are most significant.

A. The Role of Mathematics

Mathematical techniques are never for free: but where mathematics is brought in a proper way, performance increases and software costs decrease. This fact was pointed out at the meeting with GEC-Alsthom [2]: E. Daclin noticed that the part of the application software which is mathematical in nature does not bring much trouble when software development is considered. Indeed, mathematics do provide formal guarantee and support, to certify the domain of validity of the algorithms in consideration. Furthermore, mathematical equations build a compact specification and thus often result in small sized executable code.

So a natural question arises, namely: *why not try to bring mathematical techniques where they are not yet used, in order to improve formal guarantees and to reduce the size and combinatorial complexity of the resulting code*? We shall now discuss several available techniques which contribute to achieve this long range objective.

The CACSD Revolution: As already noticed, CACSD tools make the cost of designing algorithms essentially negligible, given that appropriate modeling has been achieved. CACSD packages are so valuable because they bring advanced mathematical techniques of signal processing and control to engineering, making them accessible through user friendly environments.

Alternatives to Expert Systems and Heuristics: The A.I. community itself questions expert systems: rules or knowledge bases are not the universal way of performing reasoning. This community itself has proposed neural networks and fuzzy systems as alternatives to rules and more classical logics. But these techniques are now better recognized as nonparametric modeling, identification, and control approaches, and have indeed deeper connections with mathematical statistics, and adaptive signal processing and control [20] [19]. Hence, one may easily guess that these alternative approaches will benefit from having now available mathematics associated with them.

Mathematical tools and models are often refused for large complex systems, see [3] and [6] for discussions on this point. Less formal concepts of "object oriented" approaches and "conceptual or functional architectures" are often preferred in the control industries, mainly because they do not require mathematical training. In this direction, *software models* [5] [17] may provide a significant help in mastering otherwise loosely structured processings. On the other hand, [6] and [15] clearly insist on the benefit one may expect from using, for large scale systems, modeling and mathematical techniques of control, when appropriate skills are available.

Similarly, diagnostics is considered as a typical area where expert systems techniques should be used [3], [17]. But early warning of deep internal faults or hidden malfunctionings in systems or plants may be better handled using modeling and appropriate statistical techniques of signal processing and system identification [14].

DEDS Synthesis, Evaluation and Control: Designing automata, developing supervision systems, is generally viewed as a programming activity, very different in nature from, say, control design. This is no longer true, however. In this special issue, Balemi *et al.* [4] show how to develop a discrete controller for a manufacturing system of real industrial size and complexity using a variation of the Wonham–Ramadge DEDS control paradigm. Here, simplification comes from the fact that explicit programming of the transition chart of the automaton has been replaced by some appropriate modeling, plus the expression of the control objective. And this reduction in complexity—replacing explicit programming by "executable specifications"—has been achieved thanks to the mathematical DEDS paradigm plus efficient computing techniques for boolean equations, which originated from computer science.

More generally, there is a new approach to real-time programming [18], where programs are viewed as dynamical systems so that control and system theoretic paradigms can apply in the very same way Balemi *et al.* handled their application. For large and complex systems it thus becomes feasible to generate executable code directly from high level specifications like block diagrams.

B. Limits of Classical Approaches of Control

These limits exist, however, and are bound to:

- The particular ways nonlinear systems are considered in control science: major emphasis is put on algebraic and geometric aspects, but little use of statistics is made today in these theories. Consequently model fitting from data is difficult, and nonlinear control has most success where models are known from physics.
- The lack of a proper paradigm for hybrid systems; consequently, "discrete" and "analog" worlds remain separated, in both research and in the way industrial application software is developed (different frameworks often result in different and noncompatible software tools).
- The lack of a proper paradigm for modularity in complex systems. Such a paradigm is really needed: complex systems must be designed by parts, but when developing one particular subsystem, an appropriate "summary" of other subsystems would be needed to help the final integration.

A.I. and computer science communities have proposed approaches, paradigms, and techniques to overcome some aspects of the above mentioned difficulties. Let us scan them briefly.

The Software Approach to Facilitate Handling Loosely Structured Heuristics: Expert systems shells, object oriented programming, and reuse of generic software, is viewed as a classical way to handle complexity in software—but not to remove it; see [3], [17] for related discussions.

Actor or Agent Based Software Models to Tackle Very Large Scale Systems: A different but interesting approach is that of the SPLICE system presented in [5]. SPLICE is a way to organize and structure software for large control and information processing applications according to some "software model." In this model, the application software is viewed as a "dynamical data base," where functions, also called *agents*, are dynamically created or killed, and communicate in a fully asynchronous mode via software-bus techniques. Here reduction in complexity generally results from loose coupling between agents. Typical relevant applications would be large distributed and dynamical systems such as transportation systems or air traffic control.

The "Fuzzy Success Story": Fuzzy information processing or control systems have been proposed as an alternative approach when mathematical modelling is either difficult or not needed. The technique seems appropriate for a direct automation of manual operating procedures, i.e., a controller that mimics a human operator. An interesting observation is also that several of the techniques like neural networks and fuzzy controllers are often packaged in software that is easy to use. The control community could learn from this way of packaging concepts and theory. The paper [9] contains interesting industrial views on how different techniques are combined.

IV. Control Paradigms Beyond Traditional Control Science

The aim of this section is to open new avenues for research by bringing control paradigms beyond traditional control science. This can be achieved in three different directions: algorithms, overall system design, and software development.

A. Models and Algorithms

Information Processing: Information processing is an area where tools are available to handle hybrid aspects in a mathematically established way. Mixing dynamical models of control with statistical decision making techniques helps balancing in a nice way numerical and discrete data sets. This is in contrast to the more unbalanced way expert systems are used for information processing: analog-to-discrete conversion is performed close to sensors and without performing any signal processing, so that a complex combinatorial object results which is difficult to handle. Efforts to further develop this area within the control community should be undertaken.

Nonparametric Modeling and Associated Algorithms: Neural networks have been one attempt of the A.I. community to enter the field of adaptive systems [20]. The general area of neural networks and wavelet approximations [21] is now recognized as a new *nonparametric approach to nonlinear systems modeling, identification, and* (to a lesser extent) *control*; mathematical results are available and further effort is also needed. Fuzzy control in its most widely used version just relies on a function structure closely related to that of the preceding classes, namely superposition of splines. It is thus not surprising that identification techniques (the so-called "fuzzy-neural" techniques) can be associated with them. But a good additional idea within fuzzy control is that of having some syntax associated with these functions (the "fuzzy rules") to describe them in a user-friendly fashion. Nonparametric systems theories and associated algorithms must be developed associated with those model classes.

B. Overall System Design

Here follow some of the major challenges:

Plant + Computer System ≡ Object of Control Paradigm: This is especially necessary for critical systems: control strategies include plant and hardware monitoring and diagnostics, computer system control reconfiguration, and issues of fault tolerance must be considered as part of algorithm studies [5]. This requires deep interplay between rapid prototyping of hardware and its flexible modeling and is undoubtedly a formidable challenge.

Asynchronous Algorithms for Asynchronous Software Models: For some large distributed applications, software maintainability issues call for object oriented asynchronous software models. But now a question arises with such software models, namely: how to design "asynchronous algorithms" for them? Let us mention as a key application the requirements for the US Air Traffic Control for year 2010, where routing is planned to be performed as a global optimization and control problem.

C. Software Development

Real-Time Programming ≡ Dynamical Systems Studies: In our opinion, real-time software must be considered as a dynamical system: it has inputs (sensor data), outputs (actuator commands), it obeys the principle of causality, so that a notion of a "state" can be generally associated with it. See [4], [18] for further details on this aspect. Thus, we feel it important to have the control community really interested and involved in the fundamentals of real-time programming.

Software Factories for Control Systems Development Call for "Pivot"[1] Formalisms: In developing large applications, it is often required to rely on different tools such as, for instance:

- matrix calculations and computer algebra,
- simulations and modeling,
- discrete systems and automata: proofs, verifications, synthesis,
- (distributed) code generators for automatic (or at least strongly assisted) code downloading,
- performance evaluation of application software on the given architecture.

[1] A "pivot" is a mechanical device allowing in a nice way relative moves of otherwise rigid bodies; in our framework, "pivot" formalisms allow to switch in a smooth way from one framework to a different one.

Currently available tools typically focus on one or two techniques, and, today, these different tasks require different formalisms, and translations are often performed by hand. It would be of significant interest to have computer support for more phases of system development, i.e., to have within integrated environments different tools communicating *without the need for manual transcoding.* Existence of a *pivot* formalism which

- has a sufficient expressive power to cover all previously mentioned ones in some appropriate way,
- amenable to syntactic transforms to be connected to different tools while preserving semantic equivalence,
- with proper emphasis on dynamics,

would greatly improve the present situation and contribute to strive towards CACE (Computer Aided Control Engineering). This is indeed a central question in the area of *hybrid systems.*

V. Teaching

Control engineering courses should reflect in a much deeper way this intertwining between the core of control science, and related software and hardware issues. Proper teaching should be provided, that involves real-time programming and software engineering at a proper level of theory and practice. Laboratory miniplants are often encountered and used in control courses: opportunity should be taken of using advanced computer science methods for real-time systems development in conjunction with advanced control design. And, vice-versa, computer engineering courses should involve real-time systems as an important topic with its dynamical aspects. This does not seem to be the case today. Laboratory miniplants available in control courses may be the opportunity to exercise computer science and control engineering students on programming in interaction with physical systems. In these aspects, the signal processing academic community seems more advanced than the control one: the attention paid to the interdisciplinary nature of signal processing in engineering is well reflected in [22].

VI. A Guided Tour Along this Special Issue

This special issue contains different types of papers. We shall present them by grouping those which are similar in nature.

We scan first control application papers in the classical setting. Paper [8] presents joint work involving both academic and industrial people in the area of controlling hydroelectric power. This paper is an example where an old plant is upgraded with a new control system. A systematic design methodology is used to deal with nonlinear issues. It is also of interest because of the cost constraint on hardware and software. The paper [9] deals with automation in the steelmaking industries. It is of interest because of the scale and complexity of the problem and because a large variety of techniques are used to solve the problem. The paper [16] describes two applications of model predictive control. This paper also has an emphasis on the industrial aspects. It also illustrates how one methodology can be applied to two very different problems. Paper [13] deals with flexible structure control and the problem of model-and-control reduction, with application to NASA's ACES flexible structure. Article [10] reports on an academic work as applied to a real-life and very challenging problem: the control of a VSTOL aircraft. In this application, control design is highly critical, and H_∞-control was successfully used; this was feasible thanks to the availability of advanced CACSD tools. The paper [11] reports on an industrial work with an academic consulting in the area of automotive industry. Automotive industry is going to be one of the major test areas of control in the near future. Automobiles are complicated systems which will have to be (partially) computer controlled and monitored, but at cost levels similar to those of consumer electronics industry. This feature induces particular constraints when control design is considered, and this is one interesting aspect of this paper. Finally, the article [14] is jointly presented by people from former soviet academic and government industries, in the area of navigation systems monitoring for aircraft industry. It is typical of information processing systems design where a deep effort in mathematics is paid to reduce the size and complexity of the implemented software while keeping performance at a high level.

A second class of papers is concerned with discrete event systems. Paper [12] reports on a joint academic and industrial effort towards improving scheduling techniques in manufacturing systems. This paper also deals with the large scale aspects of manufacturing systems. Then [4] is a nice academic paper reporting on a real-life application of (a variation of) Ramadge-Wonham supervisory control theory in the area of semiconductor manufacturing. The automaton obtained has a state space with about 10^6 states. Controller synthesis is effectively performed using the technique of Binary Decision Diagrams to compute with boolean formulae; this technique originates from computer science. This paper is a good illustration of how control theoretic paradigms may be usefully applied to computer science.

A third class of papers concentrates on large scale systems and how to take advantage of mathematical techniques of modeling and optimization or control for them. Paper [15] is a purely industrial paper from the area of power industry. It is concerned with day-to-day operation of electricity generation in France globally. This is a very challenging control problem. Modeling, algorithms design, and final software implementations are discussed. Then paper [6] is a fascinating one, with an academic origin. This paper reports on a ten year experience of cooperation with the water supply industry (daily optimal control, and state estimation and leak detection). Mathematical techniques and experimental results are presented, and the practical difficulties in the real-life implementation of the corresponding software and its acceptance by industrial engineers are discussed.

Finally, the last class of papers are less easily identified with any particular current branch of control engineering: these are papers from industry presenting purely software solutions to some of the problems we discussed above. These papers are present in this special issue due to their ability of stimulating control scientists to consider seriously the questions raised by their authors. Paper [17] deals with the problem of sensor data validation in large control systems. This work uses expert systems technology and relies on very little modelling as a way to summarize knowledge about a given plant. Emphasis is put on recording the operator's expertise, and it is argued that relying on such a technology allows this company to reuse easily the same shell for different applications with reduced marginal cost. Paper [5] concentrates on large control or information processing systems, where many functions—also called "agents"—have to cooperate (e.g., air traffic control). A software model is proposed for this purpose, which results in a fully asynchronous way of communicating data between these agents. This particular feature certainly would influence in turn the way control and information processing algorithms should be designed.

Acknowledgment

The authors are indebted to John Baillieul, Maarten Boasson, and Pravin Varaiya for helpful comments on and fruitful discussions about earlier versions of the manuscript. We would also like to thank Mrs. Eva Schildt who provided excellent administering assistance during the whole reviewing process.

References

[1] K. J. Åström, A. Benveniste (chairman), P. E. Caines, G. Cohen, and L. Ljung, "Facing the challenge of computer science in the industrial applications of control, a joint IEEE-IFAC project, Summary and Recommendations," to appear as an IFAC report, 1993.
[2] A. Benveniste, "Meetings held with Thomson-CSF, GEC-Alsthom, and Siemens AG," *IEEE Contr. Syst.*, vol. 11, no. 4, pp. 86–92, June 1991.
[3] G. Cohen, "Meeting with Electricité de France," *IEEE Contr. Syst.*, vol. 11, no. 4, pp. 92–94, June 1991.
[4] S. Balemi, G. J. Hoffman, P. Gyugyi, H. Wong-Toi, and G. F. Franklin, "Supervisory control of a rapid thermal multiprocessor," this special issue.
[5] M. Boasson, "Control systems software," this special issue.
[6] P. Carpentier and G. Cohen, "Applied mathematics in water supply network management," this special issue.
[7] G. A. J. M. van Ditzhuijzen, "The controlled cooling of hot rolled strip: A combination of physical modeling, control problems, and practical adaptation," this special issue.
[8] A. H. Glattfelder and L. Huser, "Hydropower forebay level control—A case study," this special issue.
[9] T. Harakawa and T. Kawaguchi, "Digital control in iron- and steelmaking process," this special issue.
[10] R. A. Hyde and K. Glover, "The application of scheduled H_∞ controllers to a VSTOL aircraft," this special issue.
[11] M. K. Liubakka, D. S. Rhode, J. R. Winkelman, and P. V. Kokotovic, "Adaptive automotive speed control," this special issue.
[12] P. B. Luh and D. J. Hoitomt, "Scheduling of manufacturing systems using the Lagrangian relaxation technique," this special issue.
[13] K. Liu and R. E. Skelton, "Integrated modeling and controller design with application to flexible structure control," this special issue.
[14] I. Nikiforov, V. Varavva, and V. Kireichikov, "Application of statistical fault detection algorithms for navigation systems monitoring," this special issue.
[15] A. Renaud, "Daily generation management at Electricité de France: from planning towards real-time," this special issue.
[16] J. Richalet, "Industrial applications of model based predictive control," this special issue.
[17] L. Thompson and G. Mertz, "Real-time expert system implementation at Monsanto-Krummrich," this special issue.
[18] A. Benveniste and G. Berry, Eds., "Another look at real-time programming," special section of *Proc. IEEE*, vol. 9, no. 9, Sept. 1991.
[19] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, Sept. 1990.
[20] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrope, "Neural networks for control—A survey," *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.
[21] I. Daubechies, S. Mallat, A. S. Willsky, Eds., *IEEE Trans. Inf. Theory*, special issue on *Wavelet Transforms and Multiresolution Signal Analysis*, vol. IT-38, no. 2, 1992.
[22] "DSP Education," special issue of *IEEE Signal Proc. Mag.*, vol. 9, no. 4, pp. 16–48, Oct. 1992.

Albert Benveniste (M'81–SM'89–F'91) was born on May 8, 1949 in Paris, France. In 1971, he graduated from Ecole des Mines de Paris, France.

He performed his Thèse d'Etat in Mathematics, probability theory, in 1975. For his thesis, he worked on *probability theory, stochastic processes, and ergodic theory*. In parallel, he pursued work on *automatic control and signal processing* in the area of *adaptive and change detection algorithms for time-varying systems*, with applications to data communications, vibration mechanics and fault detection in process control. From 1976 to 1979, he was an Associate Professor of mathematics at Université de Rennes I, France. Since 1987, he has been involved in developing a statistical theory of multiresolution signal and image processing. Since 1981, Albert Benveniste was interested in computer science, in the area of real-time languages and systems. Cooperating with P. Le Guernic, he participated to the definition of and theoretic studies on the Signal language. He has been Directeur de Recherche at INRIA since 1979.

In 1980, Albert Benveniste was co-winner of the IEEE TRANSACTIONS ON AUTOMATIC Control Best Transaction Paper Award for his paper on blind deconvolution in data communications. In 1990, he received the CNRS Silver Medal and in 1991 he has been elected IEEE Fellow. From 1986 to 1990, he was Vice-Chairman of the IFAC Committee on Theory and is now Chairman of this committee for 1991–1993. From 1987 to 1990, he was Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He is currently Associate Editor of the *International Journal of Adaptive Control and Signal Processing, International Journal of Discrete Event Dynamical Systems* and Associate Editor at Large for IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He has coauthored with M. Metivier and P. Priouret the book *Adaptive Algorithms and Stochastic Approximations*, and has been an editor, jointly with Michèle Basseville of the collective monograph *Detection of Abrupt Changes in Signals and Systems*.

Karl Johan Åström (M'71–SM'77, F'79) was born in Östersund, Sweden, in 1934. He was educated at the Royal Institute of Technology (KTH), Stockholm, Sweden, where he received the M.S. degree in engineering physics and the Ph.D. degree in control and mathematics, in 1957 and 1960, respectively. In 1987, he was awarded the degree Doctuer Honoris Causa from l'Institut National Polytechnique de Grenoble, France.

From 1955 to 1960, he held teaching appointments in different departments at KTH, and at the same time worked on inertial guidance for the Research Institute of National Defense in Stockholm, Sweden. During that period, he developed a new method for

Schuler tuning of an inertial platform jointly with Mr. F. Hector of the Philips Company. This system was built and successfully flight tested. He joined the IBM Nordic Laboratory in 1961 to work on theory and applications of computerized process control. He worked on optimal and stochastic control as a Visiting Scientist at IBM Research Laboratory in Yorktown Heights and San Jose, CA, in 1962 and 1963, respectively. Upon his return to Sweden, he was responsible for modeling, identification, and design of algorithms for computer control of paper machines. This work led to the development of the maximum likelihood method for system identification and the minimum variance control strategy. In 1965, he was appointed Professor to the Chair of Automatic Control at Lund Institute of Technology/Lund University, Sweden. From 1969 to 1970, he was a Visiting Professor of Applied Mathematics at Brown University in Providence, RI, and was a Visiting Professor at the University of Texas at Austin, from 1989 to 1990. He has also held shorter visiting appointments at many universities in the United States, Europe, and Asia. His interests cover broad areas of automatic control: stochastic control, system identification, adaptive control, computer control, and computer-aided control engineering.

Dr. Åström has been Vice Dean and Dean of the Department of Engineering Physics and Chairman of the Computing Board at Lund University. He has served as Research Advisor to many organizations, and is presently a member of the Scientific Advisory Board of L. M. Ericsson, the Technical Advisory Committee of Fisher Controls International, the Sydkraft Research Foundation, and the Research Advisory Board for the LIDS Laboratory at MIT. He is an Editor of Automatica and many other journals. He has published five books all of which have been translated into many languages: Reglerteori, (in Swedish); Introduction to Stochastic Control Theory; Computer Controlled Systems—Theory and Design, (coauthor B. Wittenmark); Automatic Tuning of PID Controllers, (coauthor T. Hägglund); and Adaptive Control, (coauthor B. Wittenmark). He has also contributed to several other books and is the author of a large number of technical papers, such as "System Identification," *Automatica*, vol. 7, pp. 123–162, 1971, (coauthor P. Eykhoff) which is a Citation Classics, "Theory and Application of Adaptive Control," *Automatica*, vol. 19, pp. 471–486, 1983, which was given the Automatica Prize Paper Award; and "Adaptive Feedback Control," Proc. IEEE, vol. 75, pp. 185–217, 1987, which was given the IEEE Donald G. Fink Prize Paper Award. He has supervised 40 Ph.D. students and numerous M.Sc. students. He holds one patent on Schuler tuning for inertial guidance (with F. Hector) and two patents on automatic tuning of PID controllers (with T. Hägglund). He is an IEEE Fellow, a member of the Royal Swedish Academy of Sciences (KVA), and a Vice President of the Royal Swedish Academy of Engineering Sciences (IVA). He has received many awards—among them the Rufus Oldenburger Medal from ASME in 1985, the Quazza Medal from IFAC, in 1987, and the IEEE Control Systems Science and Engineering Award, in 1990.