INFORMATION AND COMMUNICATION TECHNOLOGY EMBRACES CONTROL

Status Report prepared by the IFAC Coordinating Committee on Computers, Cognition and Communication

Wolfgang A. Halang [*] **Ricardo Sanz** [**]
Robert Babuska [***] **Hubert Roth** [****]

[*] *Fachbereich Elektrotechnik und Informationstechnik*
Fernuniversität in Hagen
58084 Hagen, Germany
wolfgang.halang@fernuni-hagen.de
[**] *E.T.S.I. Industriales*
Universidad Politecnica de Madrid
28040 Madrid, Spain
ricardo.sanz@etsii.upm.es
[***] *Faculteit van Werktuigbouwkunde, Maritieme Techniek*
en Technische Materiaalwetenschappen
Technische Universiteit Delft
2600 AA Delft, The Netherlands
r.babuska@dcsc.tudelft.nl
[****] *Fachbereich Elektrotechnik und Informatik*
Universität Siegen
57068 Siegen, Germany
roth@rst.e-technik.uni-siegen.de

Abstract: A new approach in control engineering ("Information Processing for Action") is presented, in which control, computers, communication and cognition play equal rôles in addressing real-life problems from very small-scale devices to very large-scale industrial processes and non-technical applications. Thus, the C2 paradigm of "Computers for Control" is shifting towards the C4 paradigm of "Computers, Communication and Cognition for Control" providing an integrated perspective on the rôle computers play in control systems and control plays in computer systems. This change is mainly due to new developments in computers and knowledge management, and the rapidly emerging field of telecommunications providing a number of possible applications in control. Control engineers will have to master computer and software technologies to be able to build the systems of the future, and software engineers need to use control concepts to master the ever-increasing complexity of computing systems.

Keywords: Real-time, soft, networked and ubiquitous computing, cognition, communication, control, embedded systems.

1. INTRODUCTION

Today, computerised and networked control systems are being used in almost all application sectors to perform non-safety and, increasingly, also safety-related functions for reasons of their inherent efficiency and flexibility. The significance of these embedded systems in everyone's life is rapidly growing. The competitiveness and prosperity of entire nations now depend on the early application and efficient utilisation of computer-controlled automation systems. Therefore, control engineering must embrace these applications and their real-time implementations. From the point of view of computer operation and control, one could likewise claim that control and system theory should be considered as supporting activities.

Computer and control researchers are today concerned with all aspects of computer-based control including real-time computing systems, real-time communications and distributed control systems, hardware and software architectures and platforms, development methodologies, software engineering and software tools, hardware and software in safety-critical applications, as well as control of the operational processes in computing systems themselves. Meanwhile, research in cognitive or so-called intelligent systems addresses all aspects of knowledge-based, fuzzy and neural systems relevant to control including modeling, identification, stability analysis, design, learning, adaptation, evaluation, implementation, optimisation of structure and parameters by means of genetic algorithms, definition of performance objectives and operation constraints, as well as awareness for computational issues and computer-aided design tools. Communication researchers are today interested in all aspects of computerised and telecommunication-based automation systems, providing services to remote equipment, particularly methods of remote and distributed control, remote sensor data acquisition, the Internet, telepresence, teleoperation, telemaintenance, telediagnosis, telemedicine, teleeducation, traffic control, robots for hazardous environments, remote industrial production, maritime and aerospace systems, and smart homes.

Although the significance of computer control is clear and quite important, researchers working in control engineering and systems science sometimes regard computers as serving the control society and as merely providing tools to implement their control laws and control structures. They develop highly sophisticated control algorithms, but are less interested in the transfer of these algorithms to well defined real-time control systems. Moreover, the application of control algorithms is, in practice, often confined to the use of existing hardware and software solutions which, in many cases, were designed for unique applications or suppliers and were sometimes developed for other purposes. This is the reason why in many cases ad hoc solutions are applied with often unsatisfactory or unreliable results. There is relatively little attention to study and to apply safe and well proven real-time software and computer architectures offered by the experts in those fields. Although there is a tendency to give stronger emphasis to applications of control engineering, there is still little attention for support technologies, such as computer control systems, which are, however, of major importance in many application areas, in which only certified systems are accepted. Besides, the complexity of systems to be controlled increases rapidly. Systems are subjected to many constraints concerning energy consumption, safety and reliability conditions, environmental protection, next to the ever-increasing demands for costs and economical production. The amount of software necessary to build a state of the art controller is continuously rising for reasons such as new required functionality, integration with alien and/or legacy systems, standards conformance, support for heterogeneous platforms, or new deployment and management capabilities. These requirements necessitate even more the concern of our community for reliable, safe and secure computer control and embedded systems.

Much attention was paid during the last decade to the real-time implementation of intelligent control systems. Recently, there is an emphasis on such issues as non-linear systems, fault detection, fault diagnosis and control system reconfiguration, which was also beneficial for the acceptance of intelligent modeling and control within the control community. New and rapid developments are taking place in computer engineering and communications, which will certainly influence the field of computer control. Telecommunications and cognitive systems are determining very much the way control systems are implemented, as well as the field of application of control engineering. A key issue in control engineering is the application to highly complex systems: the coupling of complicated and large heterogeneous systems in which different disciplines are involved and different types of information are available or has to be uncovered and to be discovered. This new approach in control engineering could be called "Information Processing for Action", in which control, computers, communication and cognition play equal rôles in addressing real-life problems from very small-scale devices to very large-scale industrial processes and non-technical applications. Thus, the C2 paradigm of "Computers for Control" is shifting towards the C4 paradigm of "Computers, Communication and Cognition for Control" providing an integrated perspective on the rôle computers play in control systems and control plays in computer systems. This change is mainly due to new developments in computers and knowledge management, and the rapidly emerging field of telecommunications providing a number of possible applications in control. Control engineers must master computer and software technologies to be able to build the systems of the future, and software engineers need to use control concepts to master the ever-increasing complexity of computing systems.

2. CURRENT KEY PROBLEMS

The current key problems in relation to computers, cognition and communication for control emanate from the significant trends in this area, which were identified in the IFAC Emerging Areas Workshop that took place in Rotterdam in August 2003. These trends and the problems they cause, in particular with respect to deteriorations of control quality, losses in safety and security, unpredictable delays etc., are discussed in the sequel of this report.

2.1 Complex System Requirements

Complexity of software-intensive controllers is increasing at an extremely fast pace. The old, well-established methods of real-time systems engineering are unable to adequately deal with the complex controls required for many important applications. Control systems now use software applications that are too complex to be constructed this way. Many requirements are hard to meet in isolation and extremely difficult to attain in conjunction. Today's software-based controllers must be:

Time-critical: The systems require high performance and/or are subject to stringent real-time constraints.

Embedded: The systems must execute on platforms with limited computing resources and interact with the external environment.

Fault Tolerant: The systems must show good behaviour under fault conditions.

Distributed: The systems are often distributed on several interacting computers.

Intelligent: The systems may require the solution of ill-posed problems, requiring a substantial level of intelligent and autonomous behaviour.

Large: The systems are large consisting potentially of millions of lines of code.

Integrated: An application requires the integration of a number of subsystems into a single, cohesive unit.

Open: The systems should be open to be interoperated by others.

Heterogeneous: The systems should execute on heterogeneous platforms.

2.2 Use of Open-market Technologies

In view of these complex requirements, the computers used in today's control systems are ceasing to be static information processing systems. The evolution towards dynamic information systems makes flexibility and hot replaceability key design issues for present and future systems. Local component intelligence is increasing and large distributed controllers are being developed as communities of interacting intelligent agents. The increased information availability implied, e.g., by the Internet and mobile device technology, is having a major impact on control system architectures, particularly in distributed systems. Modern products are often based on component architectures using commercial-of-the-shelf (COTS) elements as units. Standardisation and use of open-market technologies are current requirements in many control systems. New languages and platforms like Java, C# and CORBA are promising enhancements of ease and portability, and are expected to enable heterogeneous distributed control system platforms.

The rapid development of COTS computing and communication platforms lacking stringent timing guarantees makes static system designs based on worst-case assumptions increasingly conservative. Research is being performed on design and implementation techniques that allow dynamic run-time flexibility with respect to, e.g., changes in workload and resource utilisation patterns. In addition, it is necessary to improve the understanding of how this dynamic flexibility may be combined with more traditional real-time system approaches based on static design approaches. For example, how should event-driven execution be combined with pre-scheduled time-driven execution in embedded control systems?

Ostensibly to save development costs up front, and to shorten the time to market, more and more web technologies, COTS components, and so-called software of unknown pedigree (SOUP) are used in industrial control systems. Often, equipment, operating systems and application programs employed originate in the office world. As these components lack both the reliability and real-time capabilities required by industrial control systems, we are witnessing a technological setback raising severe security and safety problems. Owing to the complexity and unreliability of these artifacts, it is doubtful whether there will be any cost savings in the end at all, because in the overall life-cycle of a software system about 80% of the development costs are related to the test phase and further to the maintenance of the system. Moreover, in safety-related systems, the test phase assumes a critical rôle, since one cannot postpone verification and elimination of software errors to a later maintenance phase.

2.3 Support by Design Tools

One of the main drivers in control theory is the development of generic tools, independent of the application domain. Consequently, control algorithms and design techniques are applicable to classes of systems, characterised by the dynamic properties of their mathematical models, thus abstracting from the actual physical processes. This allows control engineers to re-use similar control concepts in completely different domains, and without having detailed knowledge of the application. This approach obviously has many advantages, but it also imposes some limitations on the obtained solutions. First of all, no process exactly fits in one particular class and, thus, approximations and simplifying assumptions have to be made. Secondly, one tends to select the control structure at an early stage, often on the basis of limited a priori information. The choice is then based on one's background and experience from previous projects, without seriously considering other solutions. Therefore, design-support tools need to be developed that automatically generate alternative solutions (models, controllers) to the problems at hand. By means of an autonomous commissioning system, several of these solutions could then be deployed simultaneously in some kind of self-organising controller, by using evolutionary programming, for instance. A step further would be the integration of this concept with the design of the processes themselves.

2.4 Dependability

As control systems are becoming mission-critical in many applications, i.e., a failure of them often means a failure of the systems as a whole, severe requirements such as ultimate stability and robustness, high accuracy and sampling rates, or guaranteed real-time performance, are being imposed on their performance. On top of this, the development and production costs are kept as low as possible. As a consequence, we often build systems with the minimum possible number of sensors, actuators, etc., leaving little space for additional "intelligent" features. In contrast to this, intelligent biological systems do not seem to be driven by this greedy strategy of low-cost, optimum, fail-safe performance and high accuracy at each and every step towards the final goal. Instead, they adaptively focus and defocus attention, conduct deliberate explorative actions that temporarily worsen the performance, but on the long term lead to improvement, unwillingly make mistakes that eventually facilitate more effective learning. Biological organisms are equipped with highly efficient, redundant systems for sensing the environment, processing and storing the information acquired. This is something we still can only dream of in the case of man-made systems. Clearly, a major progress can only be achieved with the use of new, much more affordable sensor technologies. However, along with better sensors, we need to develop tools to interpret the data acquired and to represent them in the form of knowledge, thus adding truly cognitive functions to control systems.

A very important new research direction within the framework of dependability is the reconfiguration of control systems. This issue is highly related to the research carried out in the area of fault detection and fault diagnosis. The results of a diagnosis should directly influence remedial measures to be taken, resulting in adaptation of control laws or complete reconfiguration of control strategies. In hazardous situations the real-time aspects related to these actions are of major importance.

2.5 Paradigm Shift

Finally, a key problem is also the need to change the thinking patterns and attitudes of control engineers to better adapt to the new realities. As a consequence of the still rapidly decreasing hardware costs, the general design philosophy must shift from resource limitation to resource adequacy rendering more understandable solutions possible. A challenge related to this is how to make better use of the ever increasing computing power without putting too much effort in programming. The parties involved in real-time computing and control must become aware of each others' problems. One needs to work closer with various groups of application engineers, and to take into account the real needs of applications in developing appropriate solutions. Consequently, many studies and designs must be documented and examined in this regard, to come up with libraries of design patterns to build on in the course of new projects.

3. RECENT MAJOR ACCOMPLISHMENTS

The nature of the requirements posed for a large variety of applications, and the availability of appropriate run-time resources, were the major driving forces that have scattered the field of computer-based controllers into technological niches with their own ideas, techniques, terminology and even technical communities. Depending on the particular applications, issues such as cost, simplicity, size, robustness, weight, availability and ability to upgrade, to distribute or to make the systems fault-tolerant were emphasised. This sketches the background in front of which the developments discussed in this section are to be seen.

3.1 Real-Time Computing

Although already in widespread use, many real-time computer control systems are still built with ad hoc techniques. Therefore, research in this area has undergone a revival, resulting in the beginnings of a more scientific basis for the construction of such systems. Many new and challenging applications were developed in the areas of command and control systems, process control systems, automated manufacturing, flight control systems, avionics and aerospace, automated defence systems, ship-board systems, submarine systems, visions and robotics. Hence, there is a wealth of results being produced in industry and academia, and the community of real-time experts is growing at a fast pace.

The research and development efforts in the area of real-time computer control yielded the following achievements so far: comprehensive tools to support the entire engineering and design process of real-time control systems from requirements elicitation via hardware configuration and software requirements specification to code generation and documentation; tools for reliability engineering and software quality assurance in hard real-time environments; high-level real-time programming languages implementing concepts of tasking, concurrency, synchronisation, and timing; efficient real-time operating systems and task scheduling strategies; adequate architectures for distributed and fault-tolerant real-time systems; hardware and software support for process interfacing; provision of artificial intelligence techniques with special emphasis on real-time expert systems; active and real-time data-base systems; fieldbus communication; predictable execution behaviour; precise synchronisation of the clocks in the nodes of distributed systems based on Universal Time Co-ordinated; real-time extensions of the Unified Modeling Language; methods for the schedulability analysis of distributed systems; and systematic re-use of system components in order to improve system reliability and to increase the productivity of designers and programmers.

The paradigm of object-orientation and especially the concept of distributed active objects as autonomous entities with concurrent behaviour was applied very successfully to the development of large and complex distributed real-time applications. Among the advantages achieved are understandability and maintainability, as problem domain semantics can directly be mapped onto models being built, modularity, because an object class represents a self-contained piece of a problem domain which can be examined separately from or within the context of a system, concurrency, since objects are natural units for concurrent execution and allow the real-world concurrency of applications to be modeled in a natural way, and better management of complexity by providing powerful abstraction concepts. Despite all this, it is counterproductive that both in academia and in industry the second step is taken before the first, viz., that object-oriented techniques are usually applied in the domain of real-time systems on the basis of inadequate languages with weak or even no capabilities for the programming of real-time and distributed systems, as this approach increases complexity again by difficult to survey work-arounds and, thus, ruins the above advantages attainable by object-orientation.

Software-intensive distributed embedded controllers are structured as dynamic collections of autonomous real-time agents interacting which each other. They have the required levels of autonomy to be effective in their tasks, while keeping interaction to achieve global objectives and modularisation to make them easy to use. Component-based development has become a critical topic in this field, because it simplifies system development. Object-agent technology provides the general development framework, while real-time distributed software architectures provide the necessary patterns for rapid and safe construction of applications. Good architectures and control component frameworks simplify the tasks of design, construction, commissioning, monitoring and maintenance of complex controllers.

Control-based approaches to model, analyse, and design embedded computer and communication systems have received increased attention from the real-time systems community as a promising foundation to control the uncertainty in large and complex real-time systems. Areas of growing interest include feedback architectures for adaptive real-time computing, theory for performance guarantees under uncertainty, integrated resource scheduling and feedback control, control-theoretical models of dynamic real-time systems, application of control theory to control timing behaviour, and optimal, robust, or adaptive feedback control in real-time systems. The use of control has the potential to increase flexibility, while preserving dependability and efficiency. Control techniques can be used to compensate for shortcomings and imperfections of implementation platforms. Control approaches to resource allocation are especially interesting for distributed control systems. A feedback scheduler can distribute the computing and communication resources in such a way that the global control performance is maximised. This is also an alternative approach to increase dependability, e.g., through dynamic reconfiguration of resources in critical situations or for graceful degradation.

3.2 Soft Computing

The term "intelligent control" has been introduced some three decades ago to denote a control paradigm with considerably more ambitious goals than typically used in conventional automatic control. While conventional control methods require more or less detailed knowledge about the process to be controlled in the form of a mathematical model, an intelligent system should be able to autonomously control complex, poorly understood processes such that some well-defined goal can be achieved. It should also cope with changes not anticipated in the process or its environment, learn from past experience, actively acquire and organise knowledge about the surrounding world and plan its future behaviour. Given these highly ambitious goals, clearly motivated by the wish to replicate the most prominent capabilities of our human brain, it is perhaps not so surprising that no truly intelligent control system has been implemented to date. Thus, at this moment, the term "intelligent control" serves as a common label for various modeling, control and optimisation techniques replicating some aspects of biological intelligence, rather than a compact research direction aiming to develop systems with a high degree of machine intelligence. The progress in the development of truly intelligent control systems has been slower than the prominent researchers in the field predicted.

In the current literature on control, the word "intelligent" is often used as a label for techniques originating from the field of Artificial Intelligence, which are intended to replicate some of the key components of intelligence, such as reasoning or learning. Among these techniques are artificial neural networks, expert systems, fuzzy logic systems, genetic algorithms and various combinations of these tools. While in some cases these techniques really add truly intelligent features to a system, in other situations they are merely used as an alternative way to represent a fixed nonlinear control law, process model or uncertainty. In this latter case, we cannot speak of any direct contribution to a higher degree of machine intelligence, but this does not mean that these methods are not useful. The opposite is true — intelligent techniques have enriched the area of control by employing alternative representation schemes and formal methods to incorporate additional relevant information that cannot be used in the standard control-theoretical framework of differential and difference equations.

For instance, fuzzy logic systems are suitable to represent qualitative knowledge, either provided by human experts (knowledge-based fuzzy control) or automatically acquired from data (rule induction, learning). Artificial neural networks, on the other hand, can realise complex learning and adaptation tasks by imitating the function of biological neural systems. Their main strength is the ability to learn functional relations by generalising from a limited amount of training data. Neural nets can be used, for instance, as (black-box) models for non-linear, multivariable static and dynamic systems, and can be trained by using input-output data observed on the system. Genetic algorithms are randomised optimisation techniques inspired by the principles of natural evolution and survival of the fittest. They proved to be effective in searching high-dimensional spaces and have found applications in a large number of domains, including the optimisation of model or controller structures and the tuning of parameters in nonlinear systems.

3.3 Networked Computing

The technology of distributed and networked computer control includes fundamental concepts and theoretical issues in system architectures, intercomputer communication, algorithms, scheduling, programming and man-machine interfaces for real-time distributed computer control systems or, more specifically: theories and techniques to ensure predictable timing, predictable behaviour under failure conditions, reliability and maintainability; methodologies and tools for specification, logical design, physical design, implementation, validation, verification and testing/evaluation; computer architectures, local-area networks, programmable logic controllers, fieldbus and standards-based platforms and environments; computer networks for real-time distributed computer control systems. The challenging requirements of telepresence systems offer new research areas for decentralised and adaptive control concepts, autonomous control strategies, real-time communication features, especially in protocols of mobile communication, but also in distributed and mobile sensor networking.

Recent developments in computer networks and communications, combined with new ways of information processing provide new possibilities also for control purposes, especially in the area of remote services. Current trends of interest take advantage of new navigation capabilities and improved telecommunication infrastructure to efficiently perform telematics applications in traditional industrial automation and the non-industrial field. Remote control techniques range from simple supervisory control, where human operators and/or computers receive simple sensor information about the situation at remote locations, and react accordingly to fulfill mission goals, up to telepresence systems, where the information includes audio-visual immersion and kinesthetic feedback in order to give human operators the feeling of being present at the remote locations. The stability and performance of remote control systems depend on the quality of communication networks. Important network parameters are varying time delay, jitter, package loss, and bandwidth. These requirements offer new fields for decentralised and distributed control concepts, autonomous control strategies, and real-time communication networks.

Optical bandwidth grows faster than processing power. Thus, there is a paradigm shift underway from processor orientation to a network centred view, resulting in different architectures of computer control systems, and in enabling various forms of telepresence. Supported by the Internet techniques remotely controlled systems become, especially in the area of robotics, more and more important. In order to have a "real" feeling of "being at" the remote place, haptic interfaces requiring force feedback control are necessary. Depending on the available bandwidth of the communication medium considerable and varying time delays may occur, which prevent at least a "well feeling" of the human operator, and may lead to command sequences resulting in unstable systems at the remote sites. Therefore, distributed control concepts including control with variable time delays are a pressing research area. It should be noted that control of communication networks is an application area of growing importance for control theory and engineering in its own right and, thus, an emerging research area.

In the realm of communication systems, wireless technologies are increasingly used in control and telematics. An important issue for control engineering is design and application of instruments and control schemes for telemonitoring, telepresence and telecontrol. Telepresence is a key technology for many robotics applications in hazardous environments, search-and-rescue missions, harvesting, telemedicine and telesurgery, telediagnosis, and telemanufacturing, but also in fields such as teleshopping, virtual environments, or entertainment. Wireless machine-to-machine communication is applied for purposes of traffic guidance and toll collection, in machine control, logistics, surveillance and alarm systems, environmental data acquisition, facility automation, remote control, consumer devices, and tools for the handicapped.

3.4 Ubiquitous Computing

Transparent-to-the-user computer software offers a platform for the development of smart systems embedded in the operation of industrial plants as well as penetrating the everyday life of modern society. By combining this type of infrastructure with new developments in virtual environments, real-time systems, complex signal processing and artificial intelligence techniques, user friendly interfaces, new intelligent control systems based on a multi-sensory perception of the state of the controlled processes and their environments are currently emerging. These so-called ubiquitous or pervasive systems encompass both communication and information technology, and provide tools to manage information easily. This enables the introduction of intelligent adaptive controllers in these systems to achieve specified goals in complex and continuously changing environmental situations. World models, built and maintained from information gathered by a multitude of sensors, provide a common abstract representation of the state of the environment. At the perception level, world models are analysed to infer relationships between different objects and to assess the consequences of controller actions.

4. FORECAST

Computer control belongs to the support technologies, which are of eminent importance for implementing control algorithms. It has been felt for a long time by the researchers working in the field of computer control that their activities did not receive the same attention as the activities of control theoreticians.

Nevertheless, for the control systems community as a whole, software will become ever more and more critical as computers and communications assume an even greater rôle in the practical implementation of advanced controllers. Therefore, software will be more highly appreciated and better understood. Future systems in this progressively hard requirements framework will be characterised by smaller size, lower cost, shorter development time, more functionality, more evolvability and, hopefully, higher dependability. In order to cope with this, the control systems community will actively follow developments within a number of areas such as agent technology, architecture-based design, artificial intelligence, concurrent engineering, composability, design patterns, distributed embedded systems, domain engineering, embedded systems, frameworks, integration, life cycles, model-based software engineering, modular systems, object-oriented programming, ontologies, product line engineering, real-time distributed systems, re-usability, software components, and software processes.

For future industrial competitiveness, new types of competence and system solutions are needed. The use of control-based approaches in analysis and design of embedded (computing) systems is one promising approach. Furthermore, low-level real-time technology will be combined with high-level aspects, concerning programming, networking, safety, security, simulation and control. The use of these technologies in the implementation of complex controllers will necessarily be based on development tools and methodologies that provide support for design, implementation, verification and deployment in a holistic manner. Extensions of the Unified Modeling Language will potentially provide us with methods to model embedded systems in such a way that we shall be able to determine properties such as responsiveness, schedulability or resource requirements of real-time systems already in early design phases.

When complexity increases, engineers rely on practice-proven, effective designs. Capturing design knowledge is a critical issue familiar to control engineers, who repeatedly re-use control designs that are well known and well documented. However, when software is involved in the final implementation many other factors, not well documented in the control design textbooks, must be taken into account. It is not easy to concisely document the control and the software parts of a controller in a coherent, integrated way. An interesting methodology that can help us with this knowledge capture task is the use of design patterns for the systems in our field. These patterns will contribute to the effective sharing of the best design knowledge, and will serve as a basis for effective systems development based on automated tool support.

Among artificial intelligence methods, computing with words is becoming increasingly relevant, especially in web mining and text recognition. So far, links and relevance to control are not clearly visible, yet, but one can expect that it will become an important item in the future. Possible applications may be in areas where co-operation of (semi-) autonomous controlled systems (robots) with humans is essential, such as collaborative decision making.