Conf-9109316--2

UCRL-JC-108590
PREPRINT

Software Tools for Distributed Intelligent Control Systems

John R. James*
Charles J. Herget

This paper was prepared for submittal to the
Third IFAC International Workshop on
Artificial Intelligence in Real Time Control
September 23-25, 1991

* United States Army
Training and Doctrine Command
Fort Monroe, VA 23651

October 1991

Third IFAC International Workshop on
Artificial Intelligence in Real Time Control
September 23-25, 1991

UCRL-JC--108590

DE92 004656

Software Tools for Distributed Intelligent Control Systems*

John R. James
United States Army
Training and Doctrine Command
Fort Monroe, VA 23651 USA

Charles J. Herget*
University of California
Lawrence Livermore National Laboratory
Livermore, CA 94551 USA

Abstract - The future of intelligent control systems depends upon the extent to which Artificial Intelligence (AI) technology can help control engineers deliver practical solutions to difficult control engineering problems. Conventional control design approaches have achieved notable successes in the design and implementation of robust, adaptive controllers for systems with well-defined mathematical models. However, conventional approaches have had difficulty supporting engineers in the design and implementation of control systems when an accurate mathematical model is not available. Also, verification that computer-controlled systems perform to specifications, validation of the specifications, higher-level control, operator decision aids, system diagnosis, operator alerting, and reconfiguration of systems which experience large changes over time or potentially catastrophic failures are significant challenges to control science and engineering. It is in these difficult areas where the AI technologies of knowledge representation, learning, search, diagnosis, planning, and decision are being used to aid control engineers. Algorithms for computer-controlled systems and software tools to help implement these algorithms have been a subject of research and commercialization for decades. Computer-Aided Control Engineering (CACE) tools have achieved a degree of success in the past decade based on their ability to assist in the control system design and implementation process. Specialized tools have been made available for system identification, system simulation, controller design and controller implementation. Recently, efforts have been made to build integrated CACE environments. Also, some current research is aimed at increasing the utility of available systems by creating a mathematical basis and a software architecture for efficiently describing complex systems and using these as a means of achieving a higher level of integration of the diverse tools already available. A recent *Workshop on Software Tools for Distributed Intelligent Control Systems* was sponsored by the U.S. Army and The Defense Advanced Research Projects Agency (DARPA). This paper will describe the results of the workshop and subsequent efforts to use these results to shape a DARPA software development project. The first section of the paper provides a brief review of the current applications of AI in the design and implementation of control systems. The second section discusses areas where AI can be applied in the near term to help solve challenges in the implementation of computer-controlled systems. The third section gives an overview of the development of CACE tools. The fourth section provides a review of the Army/DARPA workshop and the last section discusses the use of the results of the workshop.

Key Words - intelligent control, expert systems, knowledge representation, distributed processing, learning, propagation of uncertainty, possibility theory, diagnosis, reconfiguration, decision aids.

---

* Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

1. Intelligent Control Today

Use of AI technology in the design and implementation of control systems grew rapidly in the 1980's. Today there are many examples of expert systems being used to assist in system identification (Nagy and Ljung, 1989), diagnose failure or degradation of performance of complex systems (Enand *et al.*, 1991), alert operators to possibly catastrophic failures (Barry, 1989), design control systems (Birdwell *et al.*, 1985; James 1985; Trankel, 1985), and implement control systems (Salle and Arzen, 1989). Fuzzy controllers have been used commercially since the late 1970's and are currently being applied to a wide range of consumer products. (Zadeh, 1990). Expert systems have been used from low-level tuning of Proportional-Integral-Derivative (PID) control loops (Astrom *et al.*, 1986) to advisory systems for control of dispersed military forces (SDIO, 1989). Early efforts in the mid-1980's to build real-time expert control systems have resulted in a broad range of technology demonstrations and industrial implementations (James and Suski, 1988). Several of the companies which specialize in the process control industry are now offering artificial intelligence options for their control products (Honeywell and AEG among them). Also, the theme of the Advanced Control Conference for 1991 sponsored by Control Engineering and the Purdue Laboratory for Applied Industrial Control was "Expert Systems Applications in Advanced Control (Successes, Techniques, Requirements, and Limitations," (Control Engineering, 1991). Given the range of applications and the sustained active interest in the area, it may be useful to consider what has not yet been achieved.

2. Near-Term Applications of AI to Control Engineering

While a number of projects have established the technical feasibility of building expert advisory systems for control analysis and design and some have also demonstrated on-line redesign of controllers, the difficulties in coupling the numeric and symbolic computations have slowed widespread use of these techniques. However, there is now available a tool which provides close coupling of symbolic and numeric computations. This tool, Stable Factorization Package (SFPAC), is available at no charge for academic and research work (Pang *et al.*, 1990). Commercially available control design tools currently support either numeric or symbolic simulation of system performance. It is reasonable to expect that commercial tools will follow this research tool in supporting both.

Neural networks have been remarkably successful in learning the control necessary to smoothly perform motion control for robotic arms for fixed paths and weights as well as in performing other difficult pattern recognition tasks. However, one criticism of neural networks has been that they must be treated as "black boxes" which can rapidly provide outputs for inputs but whose operations are not subject to analysis. However, a recent work has indicated some success in using neural networks to perform system identification (Bialasiewicz and Soloway, 1990). In addition, recent work has been done in the use of neural nets to generate fuzzy rules (Zadeh, 1990). These efforts indicate that it is possible to construct learning systems which can assist in the on-line recognition of both minor and major changes in system dynamics and use this knowledge in the application of conventional control analysis and design methods as well as in the application of expert systems technology. A near-term realization of this capability would be a significant advancement in the realization of adaptive control systems.

Architectures for complex computer-controlled systems have been proposed by several organizations. The National Institute of Science and Technology (NIST) has been actively attempting to gain support for its Architecture for Real-Time Intelligent Control Systems (ARTICS), (Albus *et al*, 1990), in an attempt to establish a national standard . Another major effort, called the Next Generation Controller (NGC), in establishing a specification for an open system architecture standard for machine controllers is being conducted by Martin Marietta Corporation for the Manufacturing Technology Directorate of the United States Air Force's Wright Laboratory. The US Department of Energy is also sponsoring an architecture for the control of robotics at Sandia National Laboratories, Albuquerque (Miller and Lennox, 1990). There is still no general agreement on the composition of such a reference architecture for systems such as flexible manufacturing systems, power generation and distribution systems, military command and control systems, and other man-in-the-loop, complex machines. Agreement on a reference architecture would ease the development of more capable software packages for building intelligent control systems.

The underlying mathematical representations of complex computer-controlled systems is still insufficient to create a set of models which accurately captures the dynamics of the system over the entire range of system operation. We remain in a situation where we must tradeoff the accuracy of our models with the manageability of the models. Closed-form solutions of mathematical models are almost exclusively limited to linear system models. Computer simulations of nonlinear and discrete-event models provide a means for off-line design of control systems through iterative search but such simulations cannot perform exhaustive search due to the complexity of the problem. Guarantees of system performance are limited to those regions where the robustness conditions apply. These conditions may not apply during startup and shutdown or during periods of anomalous operation. Excellent results are available for cases where adequate mathematical models are known and the system is operating "close enough" to a linear region. Also, effective tools are available to model high-level system changes as a finite state machine. Several attempts to improve our modeling capabilities are focused on mapping the continuous world into a discrete one (Ramadge and Wonham, 1987; Ho, 1987; Benveniste and Le Guernic, 1990; Inan and Varaiya, 1988). However, repeated results are available which indicate that large interactive systems evolve into states where minor events can lead to a catastrophe (Bak and Chen, 1991). We are left with the result that there is a pressing need for a more adequate theory and mathematical basis for representing and predicting the performance of hybrid (continuous and discrete) dynamical systems. In the near term we will probably be able to mathematically prove (verify) that the implementation of a subset of software for computer-controlled systems performs to specifications but will have to use conventional metrics for verification of the majority of the software being used.

3. Computer-Aided Control Engineering

The availability of increasingly more economical mainframe computer systems in the 1970's led to the proliferation of first-generation software tools to aid in the analysis and design of control systems (Grabow *et al.*, 1977; Smith *et al.*, 1976; Dongerra *et al.*, 1979; Edmonds, 1979) as well as to improvements of the algorithms and numerical analysis routines available. The MATLAB program became widely used in the early eighties (Moler, 1982) and has been since used as the basis for a number of commercial and research tools (Integrated Systems, 1985; Pang *et al*, 1990). The IEEE Control System Society Technical Committee on Computer-Aided Control System Design (CACSD) maintains the Extended List of Control Software which contains one-page abstracts of a large number of commercial, proprietary, and public domain software. By the early-1980's, efforts had begun to try to provide integrated environments (Spang, 1984; Taylor *et al.*, 1989) and knowledge-based tools to assist control engineers in the application of the powerful analysis, design and implementation software already available. The rapid increase in the number of engineering workstations and the growth of national computer communication networks in the last half of the 1980's helped to fuel a corresponding increase in the number and diversity of the tools available. Also, in recent years, research in the use of knowledge-based systems in Computer-Aided Control System Design (CACSD) has increased significantly (James, 1988; MacFarlane *et al.*, 1987; Lewin and Morari, 1988; Pang *et al.*, 1990). The triennial IFAC Conference on Computer-Aided Design of Control Systems, the series of IEEE workshops on CACSD , and a series of books on Computer-Aided Control Systems Engineering (Jamshidi and Herget, 1985 and 1992) have documented these progressive changes.

During the same period, examples of building knowledge-based control systems have increased dramatically, especially in the process control industry (Astrom *et al.*, 1986; Basila *et al.*, 1990; James and Suski, 1988; Karsai *et al.*,1987; Liu and Gertler, 1987; Moore *et al.*, 1987; Le Clair and Abrams, 1988). Aspreviously stated, the SFPAC software combines the ability to perform symbolic and mathematical computations in a single software tool. Also, there are now several commercially available programs which will help build and implement knowledge-based control systems. It should be noted that there is a growing appreciation that the fields of off-line CACSD and on-line adaptive control tend to merge in the implementation of supervisory intelligent controllers (Sanoff and Wellstead, 1984; James and Rapisarda, 1988; Astrom, 1991, Basila *et al.*, 1990). The knowledge-based controllers implemented thus far are all *ad hoc* systems which require a substantial investment in construction and maintenance. As noted in Section 2 above, advances in the mathematical foundations of intelligent control are needed to lower the cost of building hybrid systems.

4.Workshop on Software Tools for Distributed Intelligent Control

There remains a very large gap between those tools which a controls engineer can use today to assist in the design and implementation of control systems and those which will be needed to achieve the next level of system complexity and integration at an affordable price. Current digital control practice is largely focused on single processor implementation of fixed controllers. While significant results are available concerning multiprocessor control systems design and implementation for communicating sequential processes (*e.g.* the SIGNAL system of INRIA; Benveniste and Le Guernic, 1990), there remains a dearth of theory or tools to assist in the design and implementation of distributed intelligent control systems. In July of 1990 the U. S. Army and DARPA sponsored a *Workshop on Software Tools for Distributed Intelligent Control Systems* conducted by Lawrence Livermore National Laboratory at Pacifica, California (Herget, 1990). Forty-eight attendees met for three days to (1) identify the current state of the art in tools which support control systems engineering design and implementation, (2) identify research issues associated with writing software tools which would provide a design environment to assist engineers in multidisciplinary control design and implementation, (3) formulate a potential investment strategy to resolve the research issues and develop public domain code which can form the core of more powerful engineering design tools, and (4) recommend test cases to focus the software development process and test associated performance metrics. In September of 1990, DARPA released a Broad Agency Announcement requesting interested activities to submit proposals to conduct research in a number of areas. The proceedings of the Pacifica workshop were made available to all attendees of a DARPA bidders workshop conducted in Pittsburgh, Pennsylvania in October of 1990. Five-year projects in the Domain-Specific Software Architectures (DSSA) area started in the Summer of 1991 with the goal of reducing life cycle software costs for computer-controlled systems by providing a set of tools to support component-based programming..

5. Applying the Results of the Workshop

The workshop resulted in a number of lengthy discussions, the most notable being a general agreement about the breadth and quality of the tools currently available and an exchange of ideas about the fact that distributed intelligent control is ill-defined. The major recommendations were: (1) perform a review of the current state of the art, (2) Develop a taxonomy of currently available tools, (3) develop a high-level integration tool, (4) develop a technology transfer plan, and (5) Establish a repository for the software developed.

The United States Army has a number of technology insertion programs either currently underway or scheduled to start which may be able to apply the software tools being developed by the DARPA DSSA projects. These technology insertion programs include the Rotocraft Pilot's Associate Advanced Technology Transition Demonstration (ATTD), the AirLand Battle Management ATTD, the Lower Echelon Command, Control, Communications and Intelligence ATTD and the Advanced Field Artillery System ATTD. Each of these technology insertion efforts have specific goals associated with increasing the flexibility and functionality of the systems through the use of computers. Difficult problems include those associated with embedded training aids, smart sensors, crew station decision aids, automatic reporting of location and status, driver vision enhancement, navigational aids, interface with current and future command and control systems, electronic technical documentation, and self diagnostics and prognostics. It is possible that near-term advances in intelligent control can be applied to achieve these requirements more cheaply.

6. References

Albus, J. S. , *et al*, Concept for a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS), National Institute of Standards and Technology, NIST Technical Note 1277, (1990).

Astrom K. J., J. J. Anton and K. E. Arzen, Expert Control, *Automatica* **22**, 277-286 (1986).

Bak, P. and K. Chen, Self-Organized Critically, *Scientific American*, pp. 46-53, January, 1991.

Barry, M. R. PX1: A Space Shuttle Mission Operations Knowledge-Based Systems Project, *Proceedings of the AI Systems in Government Conference*, Washington, D.C. (1989).

Bialasiewicz, J. T. and D. Soloway, Neural Network Modeling of Dynamical Systems, *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, Philadelphia, PA (1990).

Basila M. R. Jr., G. Stefanek, and A. Cinar, A Model-Object Based Supervisory Expert System For Fault Tolerant Chemical Reactor Control, *Comput. chem. Engn,* 14, 551-560 (1990).

Benveniste, A. and P. Le Guernic, Hybrid Dynamical Systems Theory and the SIGNAL Language, *IEEE Transactions on Automatic Control*, **35**, 535-546 (1990).

Birdwell, J. B., J. R. B. Cockett, R. Heller, R. W. Rochelle, A. J. Laub, M. Athens, and L. Hatfield, Expert Systems Techniques in a Computer-Based Control System Analysis and Design Environment, *Proceedings of the 3rd IFAC Symposium on Computer Aided Design in Control and Engineering Systems*, Lyngby, Denmark (1985).

Birdwell, J. D. and S. Liang, Concurrent Processing Environments for Distributed Intelligent Control Systems, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA 21-29, (1990).

Boeing Commercial Airplane Company, Integrated Application of Active Controls (IAAC) to an Advanced Subsonic Transport Project, NASA Contractor Report 3880, (1986).

Control Engineering, Papers Sought for the 17th Purdue Conference, *News of Control*, p. 23 (January, 1991).

Dongerra, J. J, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPAK User's Guide*, SIAM Press, Philadelphia, PA (1979).

Edmonds, J. M., Cambridge Linear Analysis and Design Program, *IFAC Symposium on Computer-Aided Design of Control Systems*, Zurich, (1979).

Enand, R., K. Cowan, and T. Knutilla, Taking Expert Systems to the Field: Lessons Learned from the Deployment of PRIDE, *Proceedings of the Symposium on Artificial Intelligence Applications for Military Logistics.* Williamsburg, VA (1991).

Gavel, D. T., Control Systems Engineering in an Open-Architectured Object-Oriented Software Environment, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA 73-82, (1990).

Glass, B. J., Hierarchical Heterogeneous Symbolic Control: Lessons Learned from TEXSYS, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA, 83-87, (1990).

Garbow, B. S., S. Nash, and C. Van Loan, *Matrix Eigensystem Routines - EISPAK Guide Extension,* Lecture Notes in Computer Science, 51, Springer-Verlag, New York, (1977).

Herget, C. J. (ed.), *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems,* Lawrence Livermore National Laboratory Report CONF-9007134, Pacifica, CA (1990).

Ho, Y. C., Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems, *IEEE Transaction on Automatic Control*, 32, 563-572 (1988).

Inan, K and P. Varaiya, Finitely Recursive Process Models for Discrete Event Systems, *IEEE Transactions on Automatic Control,* 33, 626-639 (1988).

Integrated Systems Inc., *$MATRIX_X$ User's Guide*, Palo Alto, CA 1985.

James J. R., Expert System Shells for Combining Symbolic and Numeric Processing in CADCS, *Proceedings of the 4th IFAC Symposium on Computer-Aided Design in Control Systems,* Beijing, PRC (1988).

James, J. R. and L. Rapisarda, Digital Control Workstation, *Proceedings of the IEEE/ASEE Frontiers in Education Conference* (1988).

James J. R. and G. J. Suski, A Survey of Some Implementations of Knowledge-Based Systems for Real-Time Control, *Proceedings of the 27th IEEE Conference on Decision and Control,* Austin, TX. (1988).

James, J.R., J. H. Taylor and D. K. Frederick, An Expert System Architecture for Coping With Complexity in Computer-Aided Control Engineering, *Proceedings of the 3rd IFAC Symposium on Computer Aided Design in Control and Engineering Systems*, Lyngby, Denmark (1985)

Jamshidi, M. and C. J. Herget (eds.), *Computer-Aided Control Systems Engineering,* North-Holland, Amsterdam, (1985).

Jamshidi, M. and C. J. Herget (eds.), *Advances in Computer-Aided Control Systems Engineering,* Norht-Holland, (to appear, 1992).

Karsai G., E. Blokland, C. Biegl, J. Sztipanovits, K. Kawamura, N. Miyasaka and M. Inui, Intelligent Supervisory Controller for Gas Distribution System, *Proceedings of the 1987 American Control Conference.* pp. 1353-1358 (1987).

Klose, D.R., L.U. Dworkin and L. Gorr, Software Tools for Lower Echelon Systems Development, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA, 131-140, (1990).

LeClair , S., and F. L. Abrams, Qualitative Process Automation, *Proceedings of the 27th IEEE Conference on Decision and Control,* Austin, TX (1988).

Liu K. and J. Gertler, A Supervisory (Expert) Adaptive Control Scheme. *Proceedings of the 10th IFAC World Congress on Automatic Control,* Munich FRG (1987).

MacFarlane A. G., L. G. Gruebel and J. Ackermann, Future Design Environments for Control Engineering, *Proceedings of the 10th IFAC World Congress on Automatic Control,* Munich FRG (1987).

Mann, R. C., A Hardware/Software Environment to Support R&D in Intelligent Machines and Mobile Robotic Systems, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA 203-207, (1990).

Miller, D. J., and Lennox, R. C., An Object-Oriented Environment for Robot System Architectures, *Proceedings of the IEEE International Conference on Robotics and Automation,* (1990).

Moler, C., MATLAB User's Guide, *Department of Computer Science*, University of New Mexico, Albuquerque, NM, (1982).

Moore R. L., L. B. Hawkinson, M. Lewin, A. G. Hoffman, B. L. Matthews and M. H. David, Expert System Methodology for Real-Time Process Control. *Proceedings of the 10th IFAC World Congress on Automatic Control,* Munich FRG (1987).

Nagy, P. A. J., and L. Ljung, An Intelligent Tool for System Identification, *Proceedings of the IEEE Control Systems Society Workshop on Computer-Aided Control System Design (CACSD), Tampa,* FL (1989).

Nerode, A., Modeling Intelligent Control, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA 219-226, (1990).

Pang, G. K. H., M. Vidyasagar, A. Heunis, Development of a New Generation of Interactive CACSD Environments, IEEE Control Systems Magazine, Vol. 10, No. 5, pp 40-44 (1990).

Rimvall, M. Integrating Controls System Design with Systems and Software Engineering, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA 233-243, (1990).

Salle, S. E. and K-E Arzen, A Comparison Between Three Development Tools For Real-Time Expert Systems: Chronos, G2 and Muse, *Proceedings of the 1989 IEEE Control Systems Society Workshop on Computer-Aided Control System Design (CACSD)*, Tampa, FL (1989).

Sanoff, S. P. and P. E. Wellstead, Expert Identification and Control, *Control Systems Centre Report No. 618,* Manchester, UK (1984).

SDIO Technology Transfer Workshop, AI in Command and Control, *Proceedings of the AI Systems in Government Conference*, Washington, D.C. (1989).

Smith, B. T., J. M. Boyle, *et al.*, *Matrix Eigensystem Routines - EISPAK Guide*, Springer-Verlag, Berlin, 1976.

Spang, H. A. , The Federated Computer-Aided Control Design System, *IEEE Proceedings*, 72, 1724-1731, (1984).

Symon, J. R. and H. Watanabe, Single Board System for Fuzzy Inference, *Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems.* Pacifica, CA 253-261, (1990).

Taylor, J. H. , D. K. Frederick, C. M. Rimvall and H. Sutherland, A Computer-Aided Control Engineering Environment with Expert Aiding and Data-Base Management, *Proceedings of the IEEE Control Systems Society Workshop on Computer-Aided Control System Design (CACSD),* Tampa, FL (1989).

Trankle, T. L. and L. Z. Markosian, An Expert System for Control System Design, *Proceedings of Control '85*, Cambridge, UK (1985).

Zadeh, L., personal discussion, NSF/EPRI Workshop on Intelligent Control, Palo Alto, CA (1990)

Technical Information Department · Lawrence Livermore National Laboratory
University of California · Livermore, California 94551