plete. Additions are envisioned as the need arises.

Availability of code: The program (FORTRAN code) may be obtained from COSMIC (University of Georgia, Athens, GA) after report is published.

Person to contact for details: Bruce Lehtinen, NASA Lewis Research Center, 21000 Brookpark Road, Cleveland, OH 44135.

Additional comments: The program uses the IBM Graphics Package in generating graphic output. Plots can either be displayed on-line on a scope or typewriter terminal or produced off-line on an x-y plotter.

Package or program name: SLICE (Subroutine Library in Control Engineering)

Principal developer: Control System Research Group, School of Electronic Engineering and Computer Science, Kingston Polytechnic, U.K.

Software capabilities: A library of Fortran subroutines implementing a wide range of algorithms for control system analysis and design

Interactive capabilities: None

Programming language used: FORTRAN IV (PFORT subset)

Computers and terminals on which available: Any on which suitable FORTRAN compiler is available

Documentation: User documentation for each subroutine included in source code as initial comment section. User manual to be produced in near future

State of development: Approximately 40 subroutines completed out of anticipated 60-70 to be available in first version

Availability of code: Completed subroutines available on request, subject to restrictions on commercial use

Person to contact for details: Dr. M. J. Denham, School of Electronic Engineering and Computer Science, Kingston Polytechnic, Kingston Upon Thames, Surrey, England

---

Package or program name: Waterloo Control Design Suite

Principal developer: J. D. Aplevich

Software capabilities: Classical, optimal, multivariable, frequency-domain, and algebraic design of linear time-invariant, finite- dimensional systems containing a single linear operator. File management is performed by the IBM VM/CMS operating environment, under which the programs run. At present linear systems to order about 100 can be handled, with potential for simple extension to multidimensional large, sparse systems

Interactive capabilities: Commands are entered either from the terminal or from an EXEC file or both. Neither a light pen nor a graphical menu is used

Programming language used: File management and elementary command parsing in EXEC files, numerical operations in FORTRAN and RATFOR

Computers and terminals on which available: Available via computer communications networks (DATAPAC, TELENET, TYMNET). Compatible with minor changes with any IBM VM/CMS installation. Graphics output on any Tektronix compatible terminal

Documentation: On-line documentation and archive-retrievable user's manual, plus a 230-page report describing the theory and design applications of singular pencils (not necessary for routine work)

Memory and disk requirements: Default 1 magabyte virtual machine is used

State of development: Currently implemented for linear, single- dimensional non-sparse systems

Availability of code: May be used via communication network by arrangement with U of Waterloo. Otherwise by special arrangement

Person to contact for details: J. D. Aplevich, Electrical Engineering Department, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada.

Workshop* Summaries

Development of a Software Library and an Interactive Design Environment for Computer-Aided Control System Design

M. J. Denham

Kingston Polytechnic, U.K.

This paper will describe current work in the U.K. to develop a software library of numerically robust algorithms for use in CAD of control systems. The library consists of a set of FORTRAN subroutines conforming to a set of documentation, implementation and user interface standards. The algorithms incorporated employ exclusively well understood numerical linear algebra methods based on orthogonal transformations in order to give a higher degree of predictability of algorithm performance. Several well-known, but numerically disastrous, algorithms, widely related in the control design literature, have currently been replaced in the library by algorithms based on the above principles, and these will be described in the presentation. Many other algorithms are under current consideration, as are systematic methods for testing and quality assurance of the library.

The library development is part of, and intended to form the fundamental basis of, a larger U.K. national program for software development in this field. The paper will also describe the planned program of work to develop an interactive control system CAD environment based on the use of Pascal, FORTRAN, and the UNIX operating system on a Three Rivers/ICL PERQ high performance personal computer. The aim of this program is to transfer the sophisticated software tools which are becoming increasingly available in interactive software engineering environments (e.g., the interactive LISP environment) to an interactive control system engineering design environment. Such tools include a procedure oriented incrementally compiled interactive language for system and algorithm description, tools for administration of systems, algorithms and data, inter-description translation, a "history" facility for storage, editing and re-execution of previous operations, an "undo" facility to reverse the

*Workshop on Computer-Aided Control System Design, University of California, Berkeley, California; April 14-16, 1982.

0272-1708/82/1200-0044$00.75 © 1982 IEEE

effect of previous operations, specialized editing of systems, algorithms and data using natural language and graphical methods, automatic documentation, data structures for both internal representations (output of language parser) and machine language representations (output of language compiler) for systems and algorithms.

The proposed environment will be described in some detail in the paper and the ideas presented offered for discussion.

Note: The author is currently the Chairman of the Special Interest Group in Control Engineering of the U.K. Science and Engineering Research Council's Interactive Computing Facility, which is responsible for the software development program described above.

The Federated Computer-Aided Control Design System

H. Austin Spang, III
GE Research and Development Center, Schenectady, NY 12345

The advent of the microprocessor has changed the economics of applying modern control theory to a variety of industrial processes providing an opportunity for significant performance improvements. The challenge is the selection and reduction to practice of the most appropriate control algorithm for a particular application. A cost-effective approach to control system design requires computer-aided design tools. Without these tools, the cost of exploring alternatives, of answering the "what if" questions, and generating and testing the control software becomes prohibitive.

In this talk, the structure and operation of the Federated Computer-Aided Control Design System is discussed and demonstrated. The system is termed "Federated" to indicate that it consists of several independently developed subsystems tied together by a unified data base. In this manner, one takes advantage of existing software while providing the user with a unified system that spans the entire control design problem: modeling, design, simulation and implementation. While numerous Computer-Aided Control Design packages exist [1], most are focused on a particular aspect of the design problem. The Federated System is unique in the way it ties diverse packages together into a unified system.

In creating a control system design, the engineer must construct models for the process to be controlled, analyze their behavior, design and appropriate control strategy and evaluate its overall performance. Eventually, he will implement the design in appropriate hardware such as a microprocessor. The goal of the Federated System is to provide an engineer with a broad spectrum of alternative design approaches.

The control design software indicated in Table 1 are the initial major subsystems within the Federated structure. These subsystems provide most of the desired design capability: IDPAC for modeling, CLADP and SSDP for analysis and design, and SIMNON for nonlinear simulation. In addition, the Federated system includes programs to go from one subsystem to another. These include linearization of the SIMNON nonlinear model, and generation of parameter dependent nonlinear control algorithms. The Federated system allows additional software to be added on-line. Thus the user can customize the system to meet his specialized requirements with, for example, additional graphic display programs or specialized design algorithms.

The structure of the Federated System is designed to meet the following objectives:

1. Each subsystem can be operated as a stand-alone program.
2. Subsystems and programs can be added to the system easily.
3. Subsystems and programs can be modified without affecting other subsystems.
4. Federating adds a minimum amount of overhead to each subsystem.
5. User commands can be known locally or globally.

To meet these objectives, the Federated System is organized in a hierarchical structure of stand-alone subsystems connected by a supervisory program. In the talk, the organization and structure of this supervisory program will be discussed.

One of the major problems in tying several subsystems together is to provide a common unified data base which describes the user's process and his associated control system. The user can enter a description of his process in many ways, either from measurement data or linear or nonlinear models. Once entered, the user must be able to go from one subsystem to another without having to reenter any of the previously entered or generated information. However, each subsystem has its own way of handling the information it needs. Since each subsystem has been developed independently of the others, the way the data are handled reflects the developers insight into the problem and tradeoffs determined by his computer system. Generally the way data are handled is integral to the subsystem. Any attempt to force a common data base structure would result in essentially a complete rewrite of that subsystem. The Federated System solves this problem in two ways which will be discussed in detail in the talk.

Table I
Computer-Aided Control System Design Programs Used in the Federated System

Source	Name	Author	Function of Program
Cambridge University, England	Cambridge Linear Analysis and Design Program(CLADP)	Prof. A. G. J. MacFarlane	Multivariable Control System design by frequency domain methods
Lund University, Sweden	IDPAC	Prof. K. J. Astrom	System identification
Lund University, Sweden	SIMNON	Prof. K. J. Astrom	Nonlinear simulation
General Electric	StateSpace Design Package(SSDP)	H.A.Spang	Multivariable control system design by state space and time domain methods