facilitated by means of a random access data base and a line editor developed exclusively for the Digikon software.

HONEY-X is an interactive package for frequency-domain oriented Linear-Quadratic-Gaussian feedback design of multivariable control systems. A review of this recently developed design procedure will be included in the paper. HONEY-X has been constructed on the MULTICS computer system and has been used to support multivariable control design research. It provides an effective illustration of the power of interactive packages for research in control design.

HONEY-X consists of a library of standard ANSI FORTRAN design programs which are controlled and accessed through the command and file management facilities offered by Multics. Multics also provides editing and macro command generation capabilities. The package is organized into three hierarchial levels as illustrated in Fig. 1. The levels are:

> *System Executive*—This level is an executive program written in the Multics Command Language. It is reponsible for establishing and maintaining the design system environment. Upon entry, it makes the design system library available to the user by designating subprograms locations within the Multics file system. Next the system executive solicits a command line from the user and performs first-level parsing to determine to which of the following classes the command belongs:
>
> 1. HONEY-X system command
> 2. User Macro command
> 3. Multics command
>
> The command is then passed to the appropriate processor for execution. The system executive continues to solicit commands until it is terminated with the quit command "*q*".
>
> *Program Executive*—This level consists of executive programs which interface between the system executive and the requested object programs. Its primary responsibility is to direct input to and output from the file system. It must perform second level parsing in order to determine which files should be attached. If the command line contains operational arguments, these are passed to the object program.
>
> *Objective Program*—This level performs the actual computations which are implied by the command line. It must check for compatibility of input data. If operational arguments have been passed by the program executive, it must parse them and determine if it has been given sufficient information to proceed. If input parameters are missing and do not have default values, they must be solicited from the user. Note that prompting for input is provided only if the required information is not present. This allows experienced users to avoid unnecessary input menus while giving unfamiliar users all the information needed. When applicable, the object program has the responsibility of directing printed output to the user's terminal I/O device.

Programs in HONEY-X's library were selected and developed in the course of several years of multivariable design research. Programs were added until they formed a set which would allow most manipulations to be performed by a straightforward combination of two or three commands. At that point, programs with significant overlap in function were combined. This produced two basic sets of routines—1) the "General Matrix Operations" which perform primitive operations on matrices, and 2) the "Special Purpose Programs" which perform more involved functions of linear systems analysis and synthesis. The special purpose programs required the existence of specific input matrices that have been named according to the established naming conventions. Similarily, their output matrix names are chosen so as to identify the contents. The use of standard naming conventions also allows the user to exploit the Multics file management system to performing operations (list, delete, copy, move, add name) on only a subset of the files (matrices).

Macros can be created by combining any of the supplied commands with each other or with other existing macros. If a specific capability is missing, the user can easily write his own FORTRAN program and program executive which are then usable within the HONEY-X framework.

Design Methods for Decentralized Control of Large-Scale Systems Based on Block Diagonal Dominance

W. H. Bennett
Naval Research Laboratory, Annapolis, MD

J. S. Baras
University of Maryland, College Park, MD

A methodology for design of decentralized feedback control for large-scale systems is presented based on a frequency dependent notion of low-interacting subsystems. The technique, which is motivated as an extension of Rosenbrock's Inverse Nyquist Array method, can employ one of several generalized Gershgorin-type theorems for partitioned matrices to define various measures of subsystem interaction.

A particular notion of block diagonal dominance for a rational transfer function matrix is chosen which permits the development of numerically stable algorithms based on standard software.

The technique incorporates a popular measure of a multivariable stability margin based on the maximum and minimum singular values of the subsystem transfer function matrices. Via the idea of block diagonal dominance, the method extends the local stability margins for the subsystems with interactions ignored to a global margin for the decentralized feedback structure with interactions included.

Several design examples are discussed which demonstrate the flexibility of the method in allowing different multivariable design techniques to be employed for each of the subsystems. Considerations for the development of interactive computer-aided design software for decentralized control systems are discussed.

On Control System Design by Multi-Objective Optimization Techniques

Anthony N. Payne
Lawrence Livermore National Laboratory, Livermore, CA

We consider posing control system design as a multi-objective optimization problem and survey some of the tools for solving this problem.

In the design of control systems, multiple design objectives and constraints typically specify an acceptable design. These objectives and constraints are defined with respect to system properties such as stability, robustness to noise and disturbances, speed of response, signal magnitudes, tracking accuracy, etc. The designer often cannot easily aggregate his objectives into a single objective to be optimized. Moreover, no design usually exists that is best with respect to all objectives. Hence design involves a multi-