

**AECL-8958**

**ATOMIC ENERGY  
OF CANADA LIMITED**



**L'ÉNERGIE ATOMIQUE  
DU CANADA LIMITÉE**

## **COMPUTER-AIDED CONTROL SYSTEM DESIGN**

### **Conception assistée par ordinateur de systèmes de contrôle**

**J.R. LEBENHAFT**

Chalk River Nuclear Laboratories

Laboratoires nucléaires de Chalk River

Chalk River, Ontario

January 1986 janvier

ATOMIC ENERGY OF CANADA LIMITED

COMPUTER-AIDED CONTROL SYSTEM DESIGN

by

J.R. Lebenhaft

Reactor Control Branch  
Chalk River Nuclear Laboratories  
Chalk River, Ontario KOJ 1J0

1986 January

AECL-8958

L'ENERGIE ATOMIQUE DU CANADA, LIMITEE

Conception assistée par ordinateur de systèmes de contrôle

par

J. R. Lebenhaft

Résumé

Typiquement, les systèmes de contrôle sont mis en oeuvre en utilisant des contrôleurs PID conventionnels qui sont ensuite ajustés manuellement lors de la mise en service de la centrale afin de compenser pour les interactions entre les boucles de rétroaction. A mesure que les centrales augmentent en grosseur et en complexité, de tels contrôleurs peuvent manquer de fournir un processus de régulation adéquat. Des méthodes à multivariables peuvent être utilisées afin de surmonter ces limitations.

Aux Laboratoires Nucléaires de Chalk River, des systèmes de contrôle modernes sont conçus et analysés à l'aide du MVPAK, un système de programmes d'ordinateur, qui ressemble au dire de l'utilisateur, à une calculatrice évoluée. L'ensemble de programme résout des problèmes compliqués de contrôle et donne un aperçu utile de la réponse dynamique et de la stabilité des systèmes multivariables.

Département du Contrôle des Réacteurs  
Laboratoires Nucléaires de Chalk River  
Chalk River, Ontario K0J 1J0

1986 Janvier

ATOMIC ENERGY OF CANADA LIMITED

COMPUTER-AIDED CONTROL SYSTEM DESIGN

by

J.R. Lebenhaft

ABSTRACT

Control systems are typically implemented using conventional PID controllers, which are then tuned manually during plant commissioning to compensate for interactions between feedback loops. As plants increase in size and complexity, such controllers can fail to provide adequate process regulation. Multivariable methods can be utilized to overcome these limitations.

At the Chalk River Nuclear Laboratories, modern control systems are designed and analyzed with the aid of MVPACK, a system of computer programs that appears to the user like a high-level calculator. The software package solves complicated control problems, and provides useful insight into the dynamic response and stability of multivariable systems.

Reactor Control Branch  
Chalk River Nuclear Laboratories  
Chalk River, Ontario K0J 1J0

1986 January

AECL-8958

## TABLE OF CONTENTS

	<u>Page</u>
LIST OF FIGURES	(ii)
LIST OF TABLES	(iii)
1. INTRODUCTION	1
1.1 Control System Design and Analysis	1
1.2 Description of Document	2
2. WHAT MVPACK DOES	2
2.1 Purpose of the Program	2
2.2 Design and Analysis Options Available	3
2.2.1 Frequency-Response Design and Analysis Options	3
2.2.2 State-Space Design and Analysis Options	4
2.3 Input and Data Manipulation Options Available	7
2.4 Output Options Available	9
3. A TYPICAL DESIGN AND ANALYSIS PROBLEM	9
3.1 The Nuclear Steam Generator	9
3.2 Stability Analysis	11
3.3 The Design of a Multivariable Controller	14
4. HOW MVPACK DOES IT	25
4.1 Interaction between MVPACK and User	25
4.2 Mathematical Support	25
4.2.1 Mathematical Libraries	25
4.2.2 Error Handling	25
4.2.3 Advanced Mathematical Facility	26
4.3 Data Structure	26
4.3.1 In-Core Structure	27
4.3.2 Disk Data Structure	27
4.3.3 Task Control Structure	27
4.4 Hierarchic Modular Structure	27
4.5 Coding of MVPACK Modules	28
4.6 Graphics	28
5. SUMMARY	30
6. ACKNOWLEDGEMENTS	30
7. REFERENCES	30

## LIST OF FIGURES

		<u>Page</u>
FIGURE 1	Pressurized Water Reactor Coolant Loop [6]	10
FIGURE 2	Nuclear Steam Generator - 1 Unstable Open-Loop Pole	11
FIGURE 3a	Steam Generator Open-Loop Response to a Step Change of 10% in Steam Valve Lift	13
FIGURE 3b	Steam Generator Open-Loop Response to a Step Change of 10% in Feedwater Flow	13
FIGURE 4	Open-Loop Frequency Response of the Nuclear Steam Generator	15
FIGURE 5	Open-Loop Response of the Nuclear Steam Generator	16
FIGURE 6	Unity-Feedback Control	17
FIGURE 7	Conventional Control of Compensated Plant	17
FIGURE 8	Initial INA Plot	19
FIGURE 9	INA Plot After Initial Scaling	20
FIGURE 10	INA Plot After Achieving Dominance	22
FIGURE 11a	Closed-Loop Response to Initial Offset of Steam Pressure	23
FIGURE 11b	Closed-Loop Response to Initial Offset of Downcomer Level	24
FIGURE 12	MVPACK Hierarchic Structure	29

LIST OF TABLES

		<u>Page</u>
TABLE 1	Frequency-Response Design and Analysis Modules	3
TABLE 2	State Space Design and Analysis Modules	5
TABLE 3	Input and Data Manipulation Modules	8
TABLE 4	Output Modules	9
TABLE 5	INA Commands	18
TABLE 6	Advanced Mathematics Modules	26

## 1. INTRODUCTION

### 1.1 Control System Design and Analysis

A control system consists of the plant or process that must be controlled, and the controller, or device that maintains the plant at the chosen operating level. Control engineers traditionally view the plant as a collection of separate processes each with a single input and single output. This input-output combination, together with feedback, is known as a control loop. Since the various components that make up the plant are invariably interconnected, interactions do exist between the various control loops. In the initial design stage, this interaction (or cross-talk) is usually neglected but is then often reintroduced during the detail design stage which involves computer simulation of the plant/controller system or later, by trial and error, through actual adjustments to controller settings during the plant commissioning tests.

In the past two decades, however, plants in the process industry have become more complex as a result of technological advances and government regulations regarding safety and the environment. Ever larger plants are being built to obtain economies of scale. The complex processes used in these plants tend to show stronger interactions than smaller, simpler units. Economic considerations further dictate that these plants be operated closer to maximum capacity, and thus call for significantly improved regulation.

Dynamic interactions manifest themselves in the effect of each input on several outputs. For example, an increase in the flow of feedwater into a steam generator will affect both the level of water in the vessel and the pressure of steam produced. Single-variable design methods do not adequately take these interactions into consideration. Therefore, there has developed a need for more rigorous control system design techniques to be used by engineers.

To meet this demand, multivariable design methodologies have been developed that simultaneously analyze several plant variables, allowing system performance requirements to be met during the course of designing the controller. These multivariable techniques require an extensive framework of mathematics to develop the underlying theory, and this has resulted in a wide choice of methods [1].

Although the task of control system engineers appears to have increased in complexity, two factors assist them in their work. First, computer hardware has increased in performance and greatly decreased in price in recent years. As a result, the installation of a mini or micro-computer system to assist control engineers can be readily justified. Second, computer-aided design (CAD) software, such as MVPACK, has been developed to supply the engineer with ready-made computer programs that simplify the design procedure [2]. The methods used by MVPACK lead to effective, workable solutions that meet the practical requirements of real systems. MVPACK is a set of modules that appears to users as a high-level calculator, and therefore control engineers are spared the complexities of the mathematics involved and can concentrate on the design problem at hand [3].

## 1.2 Description of Document

Section 2 consists of a description of what MVPACK does; the purpose of the program, the design and analysis options available when using it, and the resulting output. Section 3 gives an example of a design problem and describes how to use MVPACK to solve the problem. Section 4 describes how MVPACK works; the input required from the engineer, the structure of the program, its mathematical support, and its data structure.

Throughout the document, emphasis will be placed on how MVPACK can be of assistance to control system engineers who may want an understanding of how the program works but do not require detailed information to run it effectively.

## 2. WHAT MVPACK DOES

### 2.1 Purpose of the Program

The purpose of control system design is to alter the dynamic behavior of the process in order to meet the practical requirements of real systems. Thus, the control engineer strives to ensure the stability of the system, prevent overshoot, enhance the process's speed of response, and minimize its sensitivity to small perturbations or system parameter changes. To do this, the engineer uses MVPACK as a tool to work through the various steps involved in defining and solving the specific application problem.

Once the customer has specified the requirements, a mathematical model is constructed that represents the process. This model is entered into the MVPACK database, and the open-loop system is simulated on the digital computer to observe its response to various external perturbations. MVPACK solves the differential equations that describe the process, and the engineer can then interpret the results. The engineer can also examine the controllability and observability of the process, and study its frequency response by looking at Bode and Nyquist plots. Further information is obtained from an examination of the model's eigenstructure, i.e., its degrees of freedom.

The next step is to commence modifying the system as it exists by designing a multivariable feedback controller, using one or more of the design modules available in MVPACK. Finally, the entire system, namely both the process and controller, is simulated and its performance analyzed. The controller can be modified by repeating these design and analysis steps until it meets the specifications of the customer.

The controller is represented mathematically in MVPACK's database as a collection (or matrix) of feedback gains, which are then used in the actual implementation of conventional analog or digital controllers on the plant's floor. With MVPACK, the control engineer can tune the controller at the process design stage, and thus effectively eliminate the need for costly post-commissioning plant modifications.

## 2.2 Design and Analysis Options Available

### 2.2.1 Frequency-Response Design and Analysis Options

The frequency-response model describes the plant by a matrix of transfer functions representing the dynamic coupling between each input and output. It can be obtained through analytical derivations in the Laplace domain or via identification techniques.

Modules relevant to frequency-response techniques are used through a supervisor module called FREQ. The modules available under FREQ for design and analysis are listed in Table 1.

TABLE 1  
FREQUENCY-RESPONSE DESIGN AND ANALYSIS MODULES

Module	Function
COMB	Create an overall frequency-response matrix from a block diagram
GSGW	Evaluate the frequency response of a transfer function matrix
INA	Implement the Inverse Nyquist Array design method
SSGS,SSGW	Create a rational transfer-function matrix from a state-space model

(i) COMB:

COMB creates an overall frequency-response matrix from a set of subsystem frequency response matrices. It is used to analyze a block diagram in which each block can have several inputs and outputs.

(ii) GSGW:

GSGW evaluates the frequency response of a transfer-function matrix.

(iii) INA:

An important multivariable frequency response method is the inverse Nyquist array (INA) method. This method can be used with measured plant responses. It gives good

guidance for simple compensators and also generalizes to nonlinear and time-dependent systems and multivariable systems.

The INA module implements the inverse Nyquist array design method. The goal of this method is to design a matrix compensator so that the inverse frequency-response matrix is diagonally-dominant, i.e., so that for all frequencies, the magnitude of each diagonal element exceeds the sum of the magnitudes of the off-diagonal elements in the same row or column. When this has been achieved, conventional compensators can be designed for each of the resulting loops.

(iv) SSGS and SSGW:

A typical development of a control system may involve state-space design techniques followed by performance analysis in the frequency domain, and vice versa. The capability to perform model conversion from one representation to the other is provided by modules SSGS and SSGW which use reliable numerical methods to convert a state-space model into a transfer-function/frequency-response representation.

### 2.2.2 State-Space Design and Analysis Options

The state-space model describes the system by a set of ordinary differential equations. Such equations can be obtained from a conventional modelling procedure or from a system-identification technique.

Modules relevant to state-space techniques are used through a "supervisor" module called SYNT. (For more details about the structure of modules, see Section 4 of this document.) Names and functions of modules available under SYNT for design and analysis are listed in Table 2.

(a) REDUCTION OF LINEAR SYSTEMS

For most control problems, the linear model derived from an available plant model is too complex to meet the needs of the controller designer. Order reduction is an analytical procedure that extracts the matrices of the desired low-order model from the given high-order model.

(i) RED:

In MVPACK, two reduction techniques are implemented in module RED to derive a reduced-order, linear model from a given high-order model. The philosophy of the techniques is to drop the fast, stable modes, while retaining all the unstable modes and the slow modes that lie in the bandwidth of interest for the controller.

TABLE 2

STATE SPACE DESIGN AND ANALYSIS MODULES

---

Module	Function
DCOM	Design and analysis of a dynamic compensator
GRL	Design of an integral controller
MODA,MODE	Design and analysis of a modal controller
OPT,OPTA, OSIM	Design and analysis of a Kalman filter-optimal controller
PPCO	Perform pole placement with constrained output feedback
RED	Perform order reduction of linear model
REG1,REG2	Perform pole shifting
RSIM	Closed loop simulation
SIM	Simulation of linear model in state-space

---

(b) REGULATOR DESIGN MODULES

The objectives of any process or plant regulators are stability augmentation, sensitivity reduction and good steady-state tracking. MVPACK provides tools to meet these objectives as modules MODE and MODA, or REG1, REG2 and GRL, or DCOM can be used to design a conventional proportional/integral controller.

(i) MODE and MODA:

MODE synthesizes a modal controller. It performs a modal analysis on the open-loop system and prompts for the modes to be shifted and for the inputs to be used. It generates the controller matrices and activates MODA to perform the controller analysis. MODA analyzes the controller designed by MODE and is called automatically by MODE.

(ii) REG1 and REG2:

These modules are used to design proportional feedback controllers using static output feedback.

The open-loop eigenvalues are computed and displayed. Then the user is prompted to enter the eigenvalues which are actually desired and those which should be protected (i.e., not shifted).

The modules then compute the feedback matrix, form the plant closed-loop matrix, perform eigenanalysis and display the closed-loop eigenvalues. If results are not satisfactory, the user can repeat the process or use one of the other pole-shifting modules. When the results are satisfactory, the user can automatically call the integral controller module, GRL.

(iii) GRL:

The purpose of GRL is to add integral action to the proportional controller computed by REG1 and REG2. It may also be used in connection with any other proportional controller designed by other techniques. It prompts for a gain coefficient to shape the overall transient response, and it computes the gain matrix of the integral controller.

(iv) DCOM:

DCOM designs a dynamic compensator that assigns a large number of closed-loop poles. It can be called after REG1 to protect some poles while assigning some others, or it can be used independently.

DCOM prompts for the information required to compute the controller matrices. DCOM is highly interactive, so the user can influence the design step-by-step. The final result displayed by DCOM is the closed-loop eigenvalues. If they are not satisfactory, the user can resume DCOM at any of several levels. When results are satisfactory, the user can call RSIM to evaluate the dynamic response of the closed-loop system.

(v) RSIM:

RSIM simulates controllers designed by REG1 and REG2 or DCOM. It calls SIM to compute the transient. After SIM has run, RSIM can compute the mode coefficients for a given initial condition, and calls PLOT to plot the results.

(vi) SIM:

SIM simulates a multivariable controller with or without a controller.

(c) OPTIMAL CONTROL MODULES

The general problem of optimal control is to find a control law such that the response of the system minimizes a specified performance index. The performance index is to take into account engineering and economic constraints to ensure the safe and efficient operation of the system.

(i) OPT and OPTA:

In MVPACK, modules OPT and OPTA provide the facility to design an optimal deterministic regulator and a steady-state Kalman filter.

OPT designs a linear-quadratic-optimal controller or a Kalman filter by finding the solution of the appropriate matrix Riccati equation. The duality of the two problems allows both to be solved using the same technique. The method involves the modal solution of the combined state/co-state differential equations.

OPTA is a simplified version of OPT to aid in the interactive selection of cost factors. The results of an optimal control design can be simulated using the module OSIM.

(ii) OSIM:

OSIM uses data describing a Kalman filter and an optimal controller to generate a composite system, suitable for simulation with SIM which it calls directly.

(d) OPTIMIZATION-BASED METHODS

If a control system is designed to meet closed-loop performance specifications, the designed controller must go through time-consuming manual tuning with a possible degradation in the overall performance. Optimization-based design approaches have the potential to circumvent this type of problem.

MVPACK includes such a flexible and powerful technique. The module PPCO implements a design method for pole shifting with constrained output feedback. It designs a structurally constrained controller to assign a specified spectrum of the closed loop eigenvalues. Constraints may also be put on the controller gains. PPCO is highly interactive, so the user can influence the design step-by-step.

2.3 Input and Data Manipulation Options Available

Modules exist under a supervisor called DATA which are used for input and data manipulations. The names and functions of these modules are listed in Table 3.

TABLE 3

INPUT AND DATA MANIPULATION MODULES

---

Module	Function
CALC	Interactive real and complex matrix calculator
EDI	Data editor
EIG	Perform eigenanalysis on a matrix
LOA	Load data from disk file into memory
LUEN	Perform Luenberger canonical transformation
PCAL	Interactive polynomial and rational matrix calculator

---

(i) CALC:

CALC is a matrix calculator providing direct access to the MVPACK mathematical routines. It allows the user to perform most calculations on real arrays in the MVPACK data structure. Available operations range from matrix multiplication and addition, through matrix inversion, up to complex algorithms such as eigenanalysis and generation of canonical forms.

(ii) EDI:

EDI is the MVPACK data editor. It is a polynomial or real matrix editor capable of creating a new matrix, or editing an existing one.

(iii) EIG:

EIG finds the eigenvalue and eigenvectors of a square matrix using the Wilkinson QR method.

(iv) LOA:

LOA reads real and integer data into the data structure from an ordinary file created by the operating system's text editor.

(v) LUEN:

LUEN performs Luenberger canonical transformation.

(vi) PCAL:

PCAL is a calculator for polynomial and rational matrices. It performs basic matrix arithmetic operations on matrices stored in the MVPACK data structure.

2.4 Output Options Available

Table 4 lists the modules that output data, as well as the supervisors through which they are accessible.

TABLE 4  
OUTPUT MODULES

---

Module	Supervisor	Function
DIR	DATA	List matrices in directory
OUT	DATA	Display selected matrices
PLOT	SYNT	Plots results of simulation
TPL	SYNT	Produce quality time-response plots
BODE	FREQ	Produce Bode plots
NYQ	FREQ	Produce Nyquist plots

---

All plots are generated on Tektronix 4010/4014 series graphical display terminals or a Tektronix 4663 interactive digital plotter. Drivers for other devices are also available.

3. A TYPICAL DESIGN AND ANALYSIS PROBLEM

In this section, an example is given to demonstrate how MVPACK can be used to solve a particular application problem. The reader will be guided through the sequence of steps that are followed in the design of a multivariable controller for a steam generator [4].

3.1 The Nuclear Steam Generator

The function of the nuclear steam generator is to transfer heat from the reactor's primary cooling system to the secondary steam cycle which, in turn, drives a turbine generator, as shown in Figure 1. The behavior of the steam generator must be closely controlled if a steady supply of electricity, as well as the safe operation of the pressurized water reactor, is to be guaranteed.

The dynamics of the steam generator is described mathematically by a set of 15 coupled differential equations, which can be linearized about an operating point and represented using matrix notation as follows [5,6]:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t)$$

and

$$y(t) = C x(t).$$

The 15 elements of vector  $x$  describe the state of the system at any given time,  $t$ . A typical variable is the primary water temperature. The steam generator has two inputs (steam throttle valve lift and feedwater flow) and two outputs (steam pressure and downcomer level), which are given by the vectors  $u$  and  $y$ , respectively. A schematic representation of the steam generator is shown in Figure 2 (note the internal coupling of the inputs and outputs).

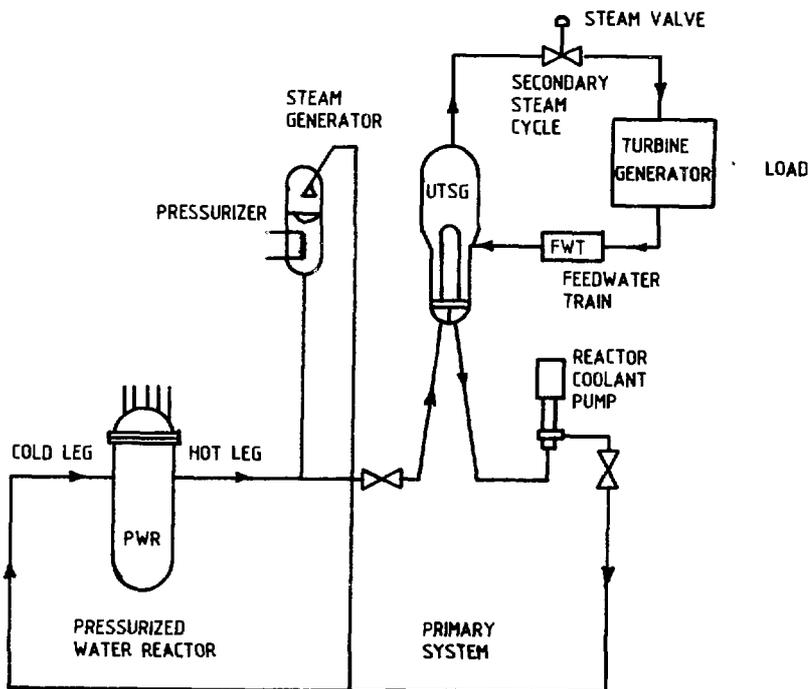


FIGURE 1 Pressurized Water Reactor Coolant Loop [6]

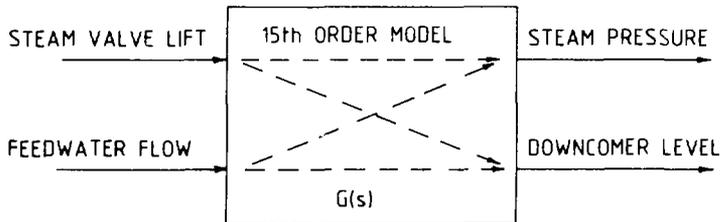


FIGURE 2 Nuclear Steam Generator - 1 Unstable Open-Loop Pole

The dynamics of the open-loop system are embodied in matrix A. Matrix B represents the way in which the inputs influence the states, and matrix C denotes how the states affect the outputs.

### 3.2 Stability Analysis

MVPACK is activated by typing "CAD" in response to the system prompt (" \$" on the VAX). The user is then prompted for information, and help messages can be obtained by entering "?".

```
$ cad
COMPUTER-AIDED CONTROL SYSTEM DESIGN PACKAGE
GIVE SYSTEM NAME > nsg
CAD COMMAND > ?
THE FOLLOWING FACILITIES ARE AVAILABLE IN MVPACK:
  DATA — DATA LOAD, PRINT, AND MANIPULATE
  SYNT — SYNTHESIS DESIGN METHODS
  FREQ — FREQUENCY RESPONSE DESIGN METHODS
  GEN — ACCEPTS ANY COMMAND OR TASK NAME
  BYE — EXIT FROM MVPACK
  X — RETURN TO NEXT HIGHER LEVEL
THE BYE AND X COMMANDS MAY BE ENTERED TO ANY SUPERVISOR
CAD COMMAND > data
DATA COMMAND > load
MVLOAD MODULE
GIVE FILE NAME > nsg.dat
DATA COMMAND > dir
MVDIR MODULE
DIRECTORY OF SYSTEM NSG
4 ENTRIES USED OUT OF 198
A      B      C      D
DATA COMMAND
```

The matrices may be displayed on the video terminal or printed on the line-printer using the DATA command "OUT". Typing "EDI" in response to the DATA supervisor's prompt lets the user edit interactively any matrix listed in the directory, as well as create new matrices. The eigenvalues and eigenvectors of the system matrix A can be easily obtained:

```
DATA COMMAND > eig
MVEIG MODULE
MATRIX NAME > a
REAL VALUES > eig-ra
IMAG VALUES > eig-ia
VECTORS > eig-va
DATA COMMAND
```

The real parts of the eigenvalues are now found in matrix EIG-RA, their imaginary parts in EIG-IA, and the corresponding eigenvectors are in matrix EIG-VA. (Other names could have been chosen here for the matrices.) The contents of these matrices are examined with module OUT. With the exception of one unstable pole ( $2.47 \times 10^{-5}$ ), all other eigenvalues are found to have negative real parts.

MVPACK may be used to simulate the process alone to show the effects of inputs on outputs and internal states:

```
DATA COMMAND > sim
MVSIM MODULE
USE PREVIOUS STRUCTURE? [Y/N] > n
TYPE O FOR OPEN-LOOP, C FOR CLOSED-LOOP STRUCTURE > o
MVEIG MODULE
MVSIM MODULE
PREVIOUS DT(s) = 0.0000 GIVE COMPUTATION STEPTIME DT(s) > 0.25
PREVIOUS OT/ODT = 0 GIVE OUTPUT SAMPLING RATIO T/DT > 4
OUTPUT SAMPLING PERIOD T(s) = 1.000
PREVIOUS TMAX(s) = .000 MAXIMUM POSSIBLE TMAX = 100.
GIVE TERMINAL TIME OF SIMULATION TMAX(s) > 100
GIVE INITIAL STATE AT THE TERMINAL?[Y/N] > n
SPECIFY STANDARD FORCING FUNCTION?[Y/N] > y
GIVE PULSE FUNCTION DATA I,H,T1,T2 > 1,0.1,0,300
DATA COMMAND > tpl
```

Module TPL was used to plot, on a Tektronix 4663 plotter, the steam generator's open-loop response to a step change of 10% in the steam valve lift and feedwater flow. See Figure 3.

Note that the initial response to the step input is often in the opposite direction to the longer term effect. This non-minimum phase characteristic, together with the fact that all four responses are significant, leads to the problems commonly associated with steam generator control.

The open-loop frequency responses from inputs to outputs can be studied with Bode or Nyquist diagrams. The FREQ supervisor's module SSGW is used to evaluate the transfer-function matrix  $G(w)$  of a state-space model. SSGW prompts for the frequency range (in Hz) and the number of desired points:

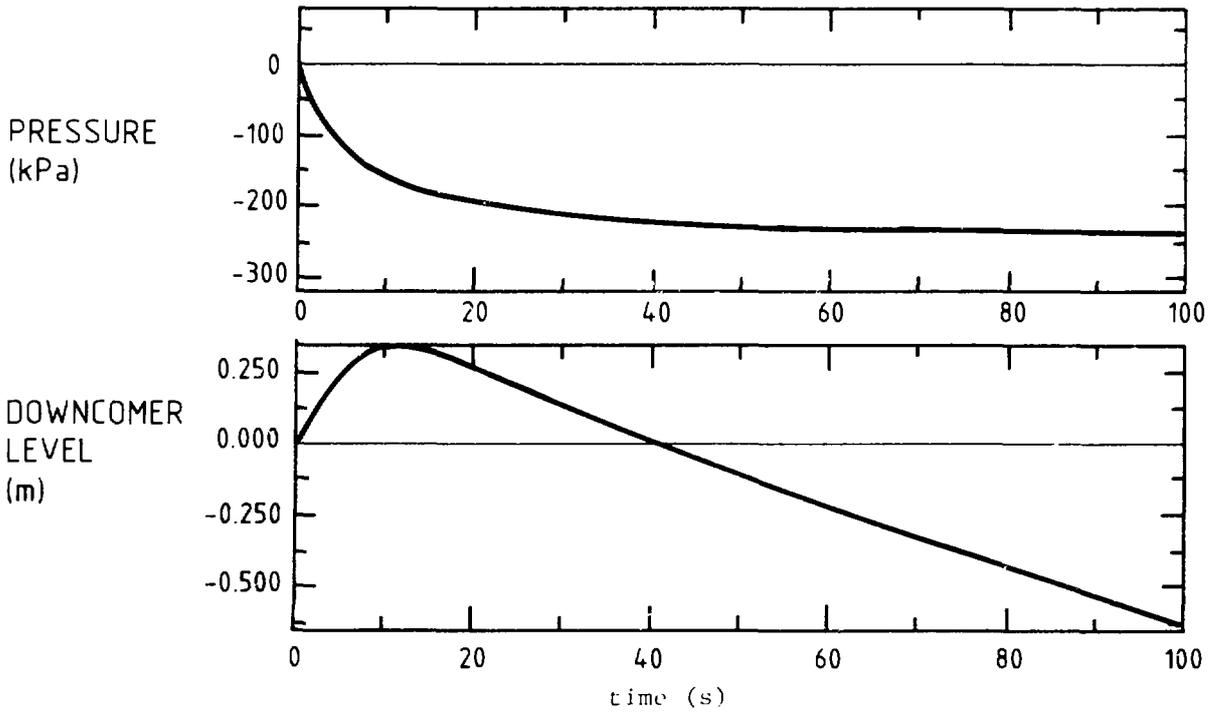


FIGURE 3a Steam Generator Open-Loop Response to a Step Change of 10% in Steam Valve Lift

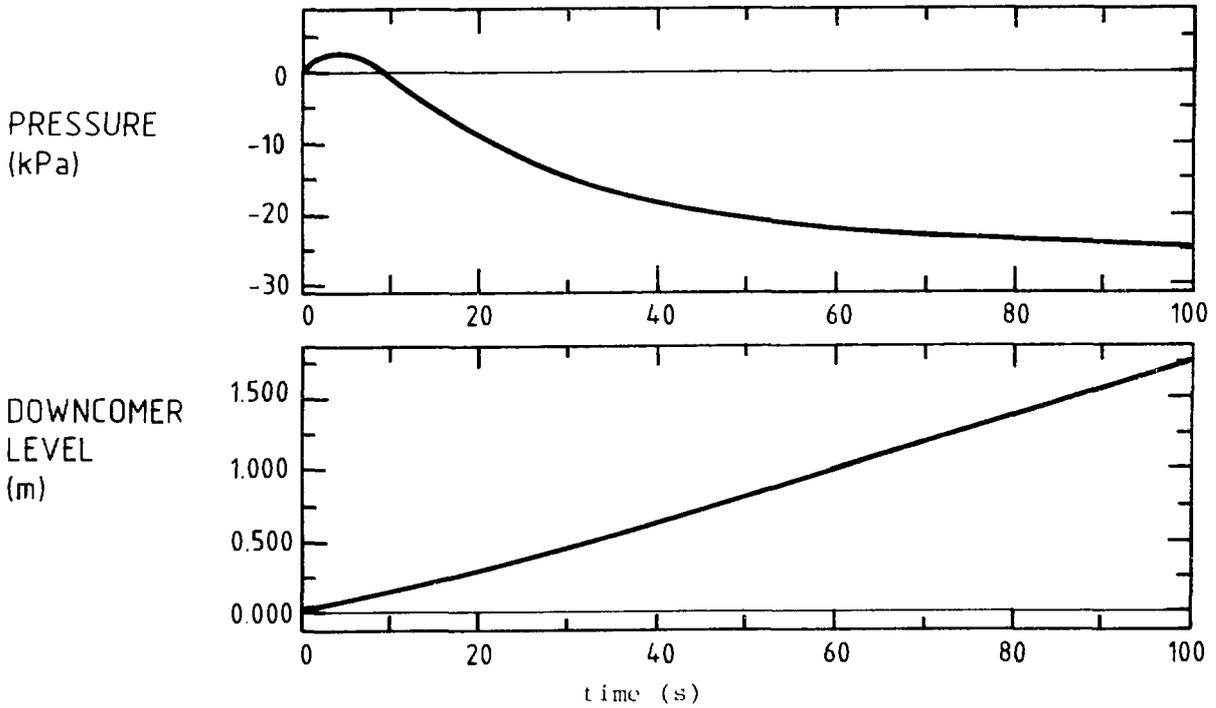


FIGURE 3b Steam Generator Open-Loop Response to a Step Change of 10% in Feedwater Flow

```
DATA COMMAND > freq
FREQ COMMAND > ssgw
MVSSGW MODULE
GIVE FMIN, FMAX, NPOINTS > .001,1,50
LINEAR SPACING? [Y/N] > n
CREATED GW, GWI AND DW
FREQ COMMAND >
```

Nyquist plots are generated with module NYQ:

```
FREQ COMMAND > nyq
NYQUIST AND INA PLOTTER
ANNOTATION LEVEL > ?
The annotation level controls plot annotation as follows
    0  Outline viewports (for testing only)
    = 1  Print bottom titles (name, date and time)
    = 5  Draw and label gain circles
    =10  Label frequencies
    =15  Print user titles
Thus the default (0) gives a normal plot, while 16 gives
only the curve, with the frequency arrows.
ANNOTATION LEVEL > 0
GIVE MATRIX NAME, NYQ COMMAND > gw,ar
GIVE FMIN, FMAX > .001,1
LOG, LINEAR, COMB OR PLAIN > log
GIVE UP TO FOUR TITLES:
TITLE > open-loop response to the
TITLE > nuclear steam generator
TITLE > <cr>
GIVE MATRIX NAME, NYQ COMMAND > CNTL-Z
FREQ COMMAND >
```

The "ar" option that appears above specifies a small plot of the complete transfer-function matrix. Bode plots are readily obtained using module BODE (see Figures 4 and 5).

### 3.3 The Design of a Multivariable Controller

It is possible to guess the structure of the controller and then, by simulating the closed-loop system's response to disturbances, iterate on the controller's parameters by trial and error to obtain a good control system design. Alternatively, the controller may be designed with the aid of one of the numerous design modules available in MVPACK.

The inverse Nyquist array (INA) is one of these design techniques. It is a graphical tool that assists the control system engineer in choosing the controller matrix. The method works in the frequency domain with the input-output transfer-function matrix, and is related to the standard Nyquist technique [7].

A block diagram representation of the complete plant with an output unity-feedback controller is shown in Figure 6.

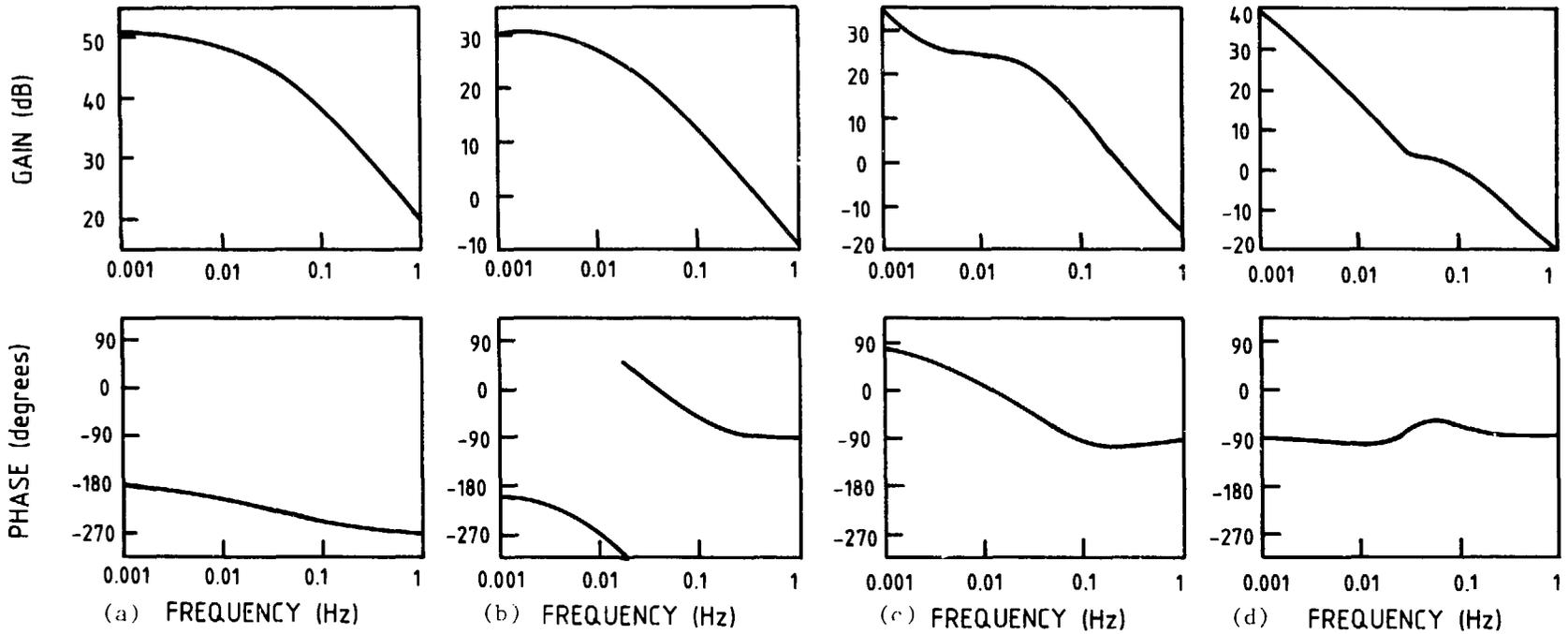


FIGURE 4 Open-Loop Frequency Response of the Nuclear Steam Generator

(a)  $G_{11} = \frac{\text{Steam Pressure}}{\text{Steam Valve Lift}}$     (b)  $G_{12} = \frac{\text{Steam Pressure}}{\text{Feedwater Flow}}$     (c)  $G_{21} = \frac{\text{Downcomer Level}}{\text{Steam Valve Lift}}$     (d)  $G_{22} = \frac{\text{Downcomer Level}}{\text{Feedwater Flow}}$

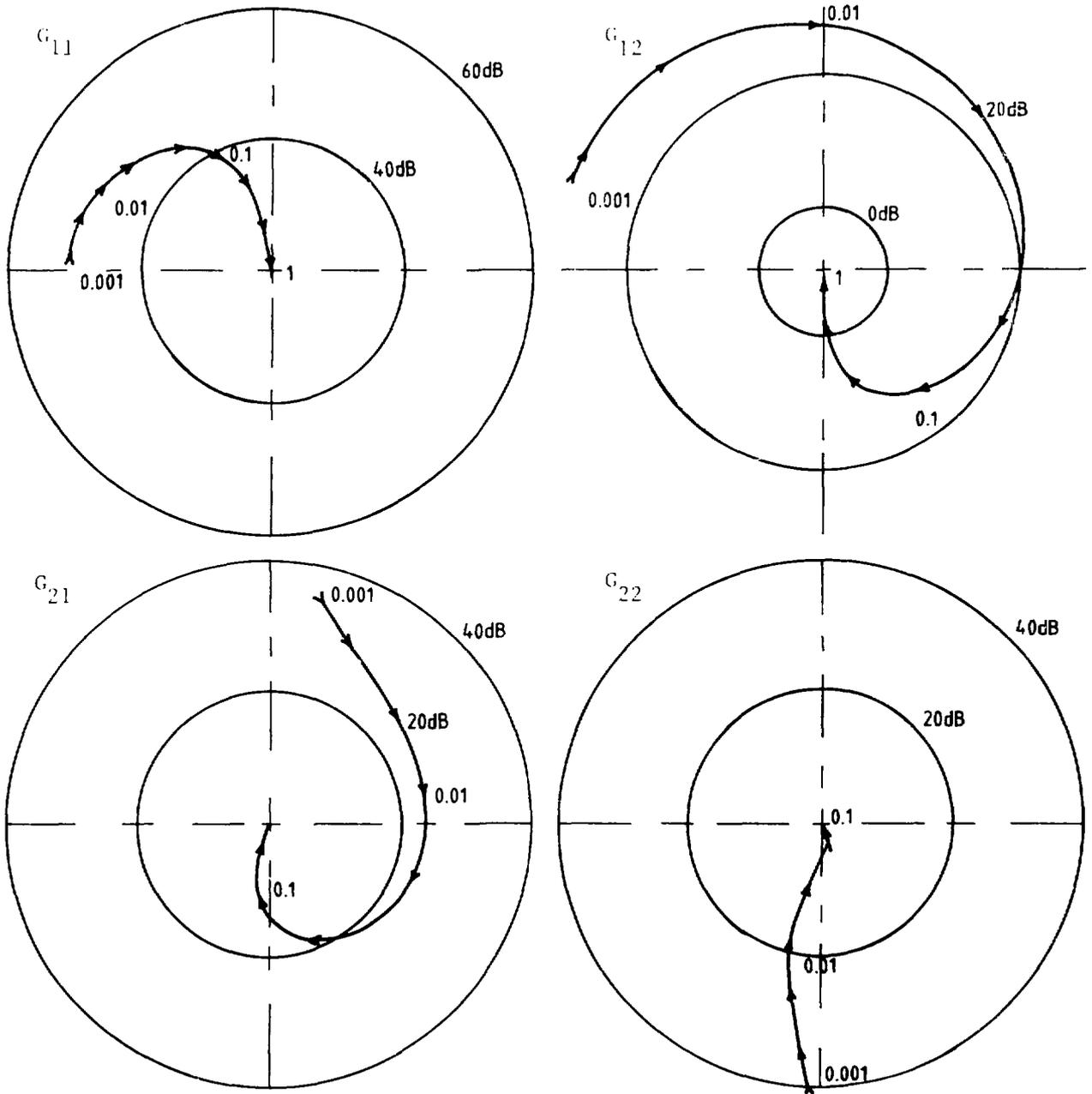


FIGURE 5 Open-Loop Response of the Nuclear Steam Generator

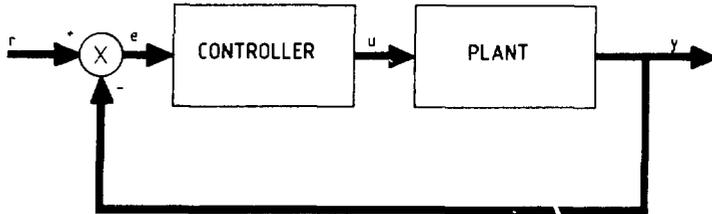


FIGURE 6 Unity-Feedback Control

Mathematically speaking, the controller is typically a gain matrix  $K$  that, for a non-interacting process, contains only diagonal elements. However, because interaction is normally present,  $K$  is usually nondiagonal.

The aim of the INA method is to decouple the interactions between the plant inputs and outputs through the addition of pre- and post-compensators. As Figure 7 shows, this enables the use of a "diagonal" controller, i.e. conventional single input-single output controllers.

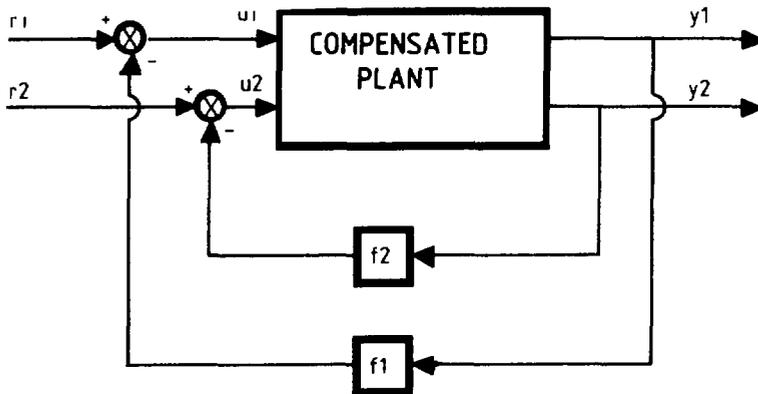


FIGURE 7 Conventional Control of Compensated Plant

Although the interactions cannot be totally decoupled, it is often possible to reduce them significantly, i.e., to achieve "diagonal dominance". This decoupling is performed by the INA module, in which the inverse of the compensated plant's transfer function,  $\hat{Q}(s)$ , is plotted with the NYQ option, and diagonal dominance is obtained via simple row and column operations.

```
FREQ COMMAND > ina
INA MODULE
CREATING QWI FROM GWI
INA COMMAND > nyq
```

The initial inverse Nyquist array plot is shown in Figure 8. As can be seen for this steam generator problem, most INA plots have amplitudes less than unity over the frequency range of 0.001 to 1 Hz.

To obtain all curves with amplitudes greater than one, row 1 is scaled by 100 and row 2 by 10:

```
INA COMMAND > row s,1,100
INA COMMAND > row s,2,10
```

Row (or column) scaling does not affect the diagonal dominance. The resulting INA plot with Gers'gorin circles is shown in Figure 9.

Note that the system is not diagonally dominant as, in general, the off-diagonal element (1,2) is greater than the diagonal element (1,1); also, the Gershgorin circles of element (1,1) encircle the origin.

Diagonal dominance can be achieved by trial and error using the methods listed in Table 5.

TABLE 5

INA COMMANDS

---

Command	Description
ROW (COLUMN) P,i,j	Permutation of rows (columns) i and j
ROW (COLUMN) S,i,	Scaling row (column) i with a constant
ROW (COLUMN) A,i,j,	Adding times row (column) j to row (column) i
ROW (COLUMN) C,i	Application of a hillclimbing algorithm to row (column) i

---

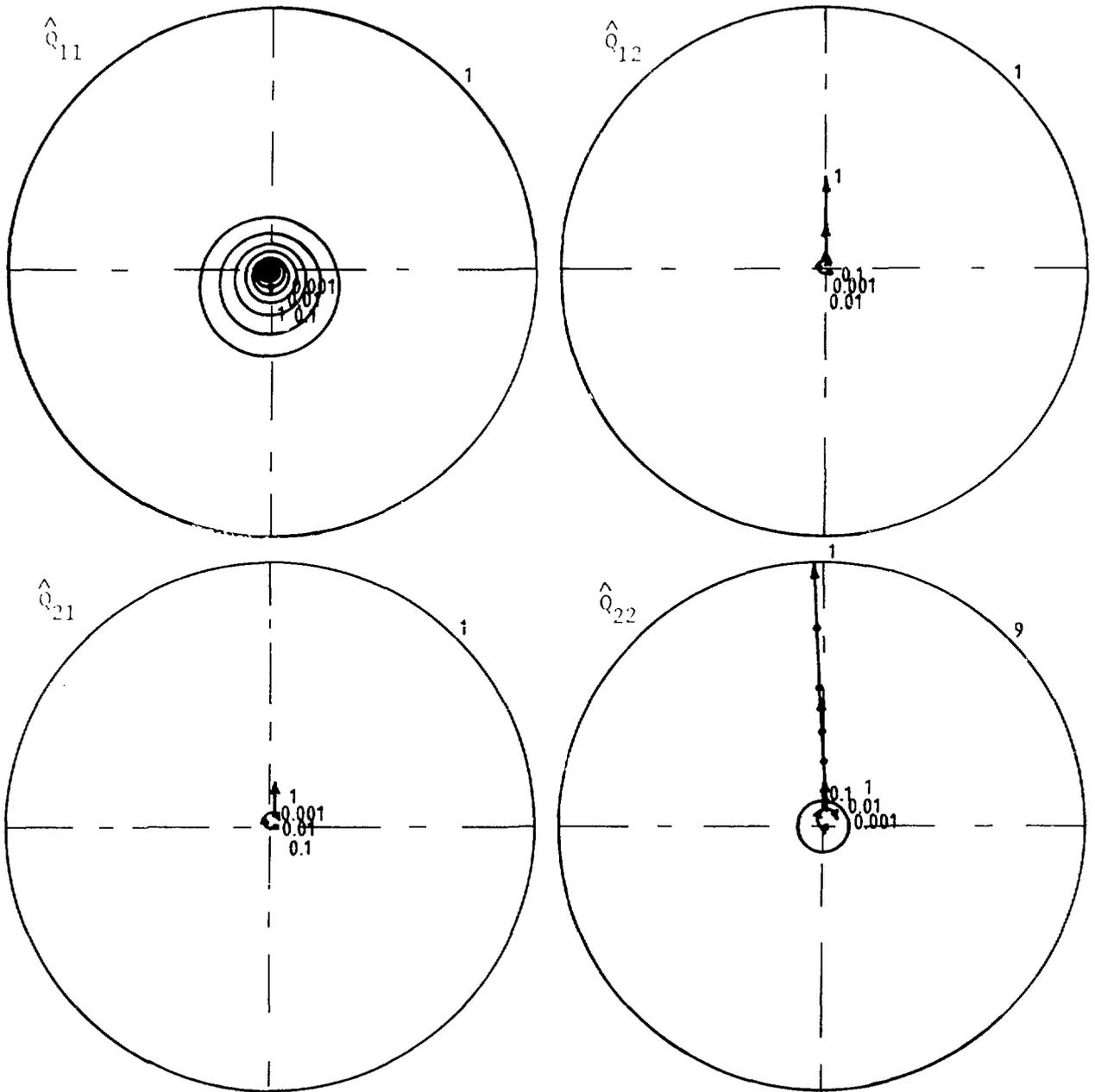


FIGURE 8 Initial INA Plot

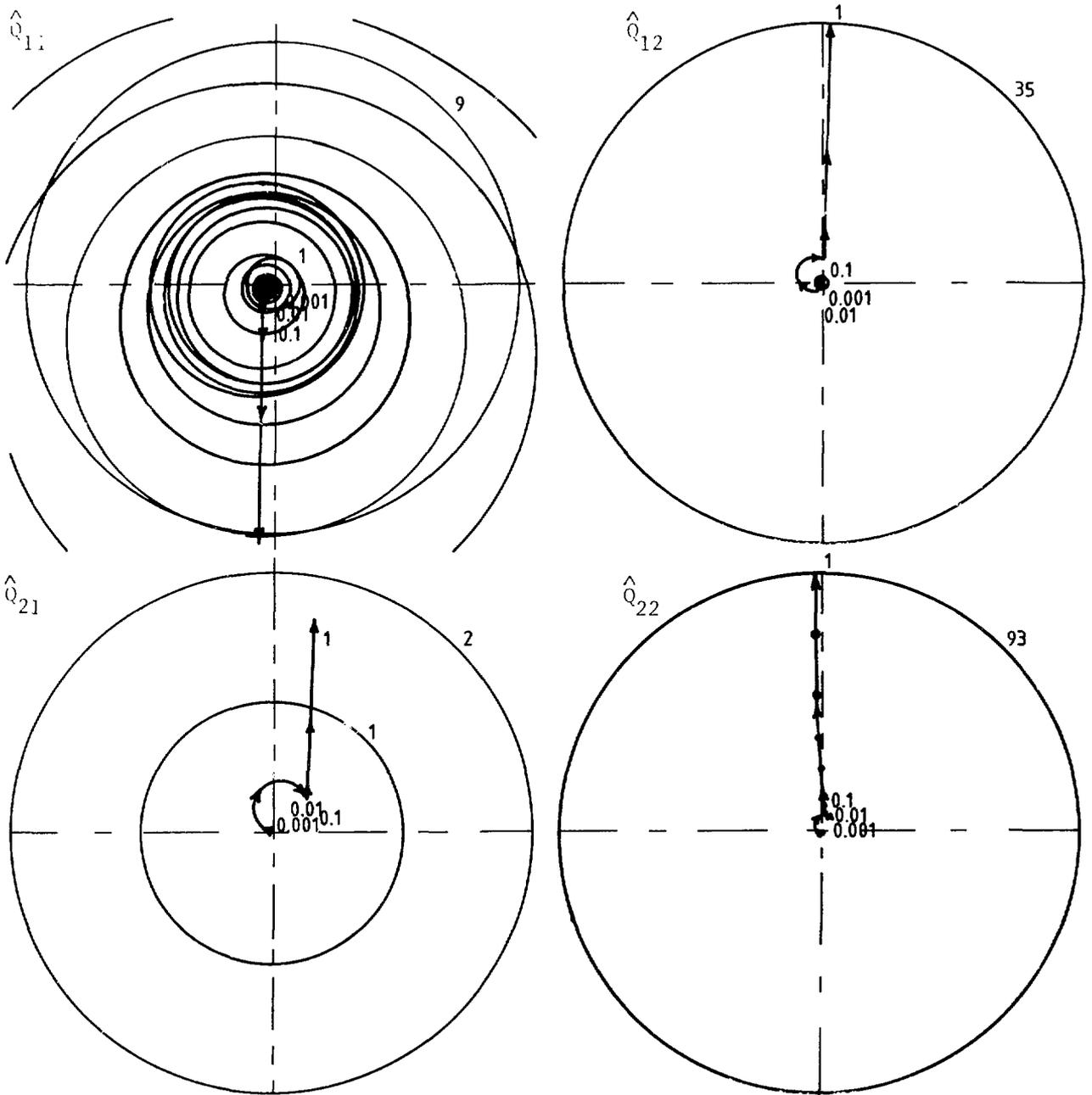


FIGURE 9 INA Plot After Initial Scaling

In this example, column 2 is scaled by 0.14 and an optimization method applied to rows 1 and 2.

```
INA COMMAND > col s,2,0.14
INA COMMAND > row c,1
INA COMMAND > row,c,2
```

These lead to the inverse Nyquist array plot of Figure 10. Note that the system is diagonally dominant.

The sequence of operations outlined above creates the inverses of the pre-compensator and post-compensator matrices, INA-MI and INA-LI, respectively. The matrix calculator, module CALC, is used next to invert these matrices:

```
FREQ COMMAND > data
DATA COMMAND > calc
INTERACTIVE MATRIX CALCULATOR
COMMAND LINE > inv ina-mi,ina-m
COMMAND LINE > inv ina-li,ina-l
```

Module OUT shows the compensators to be

$$M = \begin{bmatrix} 0.01 & 0.00293 \\ 0.00316 & 0.101 \end{bmatrix}$$

and

$$L = \begin{bmatrix} 1 & 0 \\ 0 & 7.14 \end{bmatrix}$$

The final task is to choose the values of the elements of the diagonal gain matrix F. This can be done, for example, with the aid of the modal control modules MODE and MODA. The gains of F are selected here to be as high as possible without causing a damping ratio to be less than 0.8. The gains found are  $F = \text{diag}(-0.2, 0.06)$ , and the total controller is formed by collapsing all the compensators together (see Figure 6):

$$K = MFL = \begin{bmatrix} -2.0 \times 10^{-3} & -5.859 \times 10^{-4} \\ 1.356 \times 10^{-3} & 4.325 \times 10^{-2} \end{bmatrix}$$

The matrix multiplication was performed with CALC. The response to initial offsets in the outputs of the controlled steam generator is shown in Figure 11.

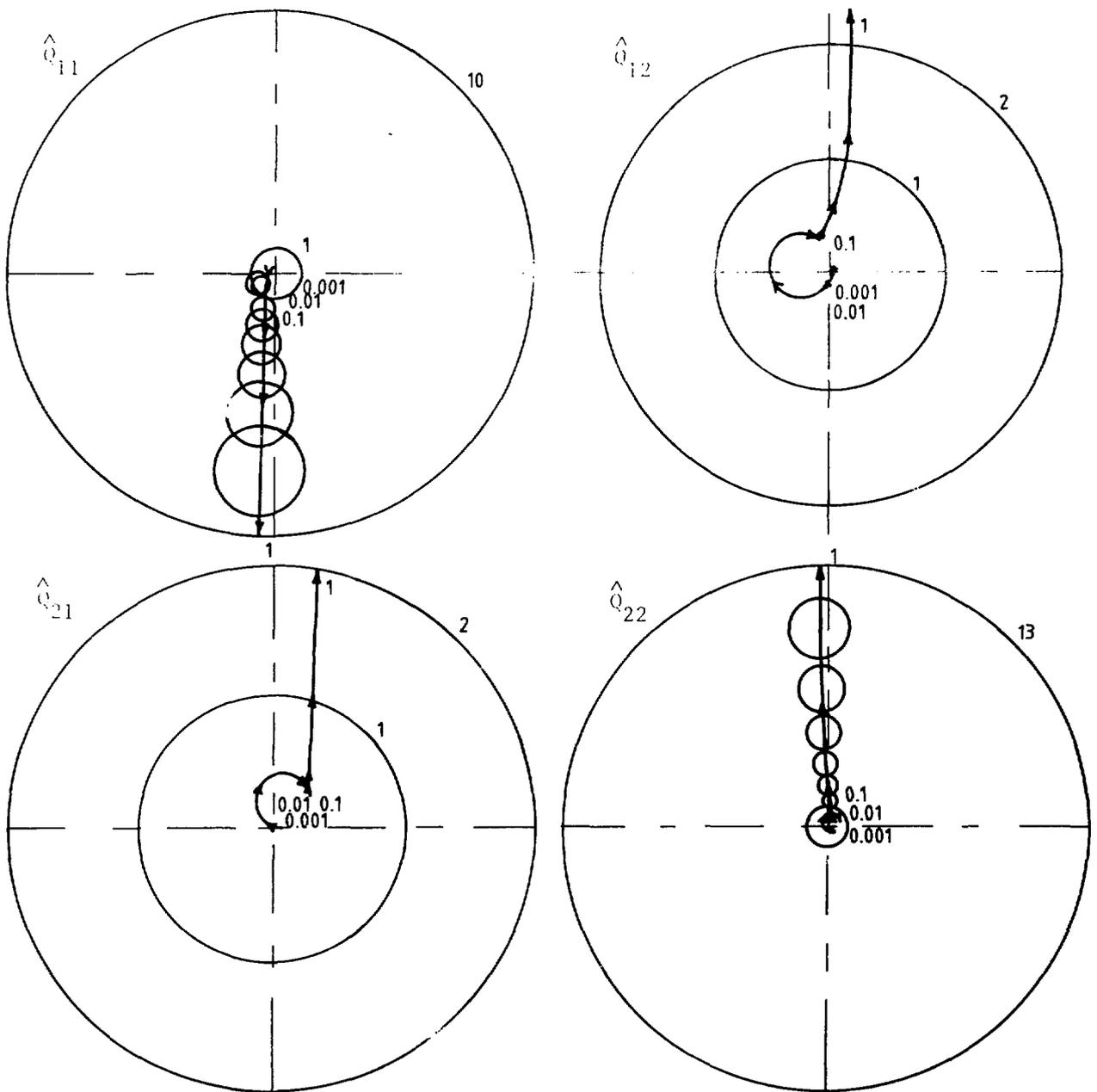


FIGURE 10 INA Plot After Achieving Dominance

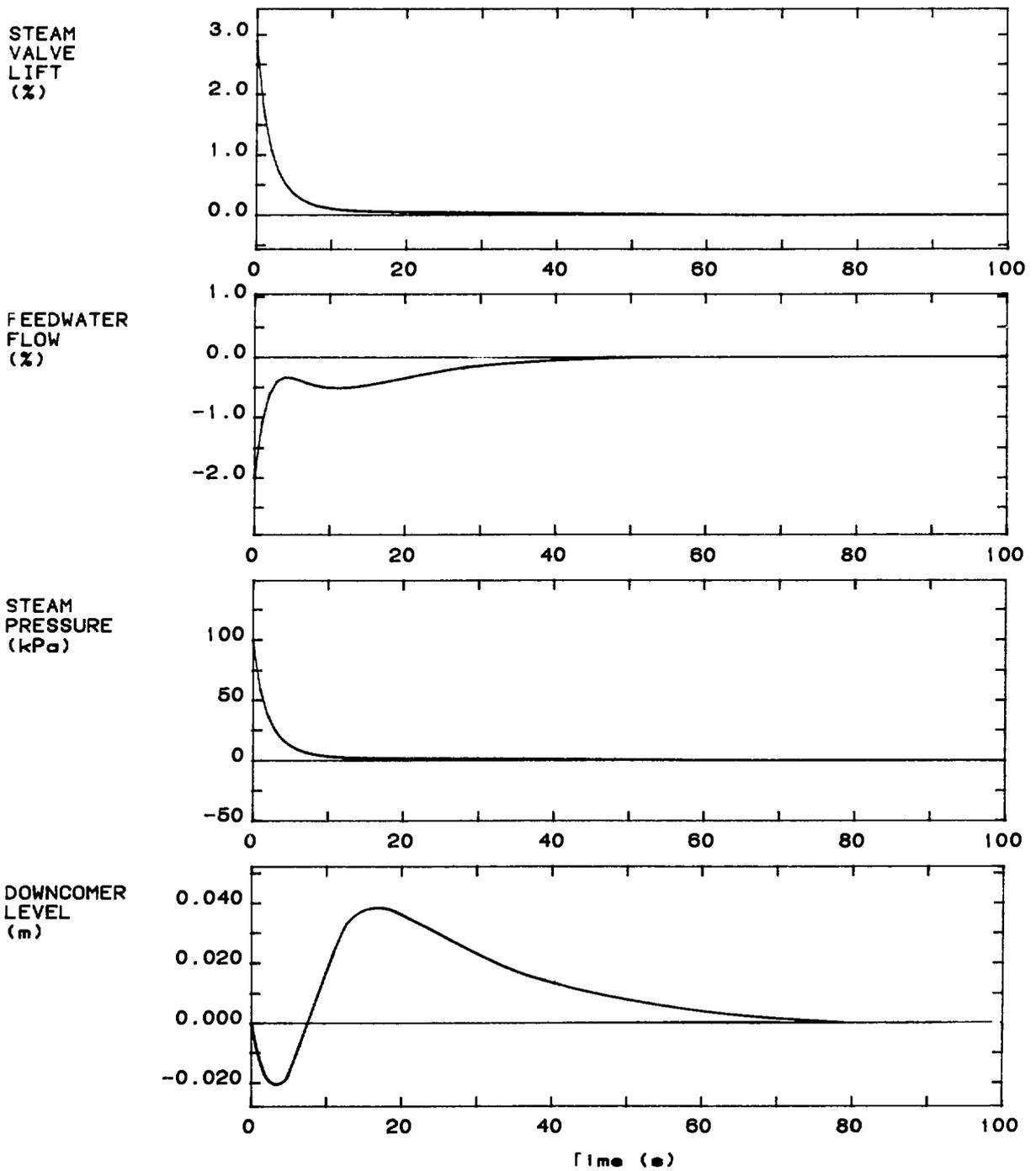


FIGURE 11a Closed-Loop Response to Initial Offset of Steam Pressure

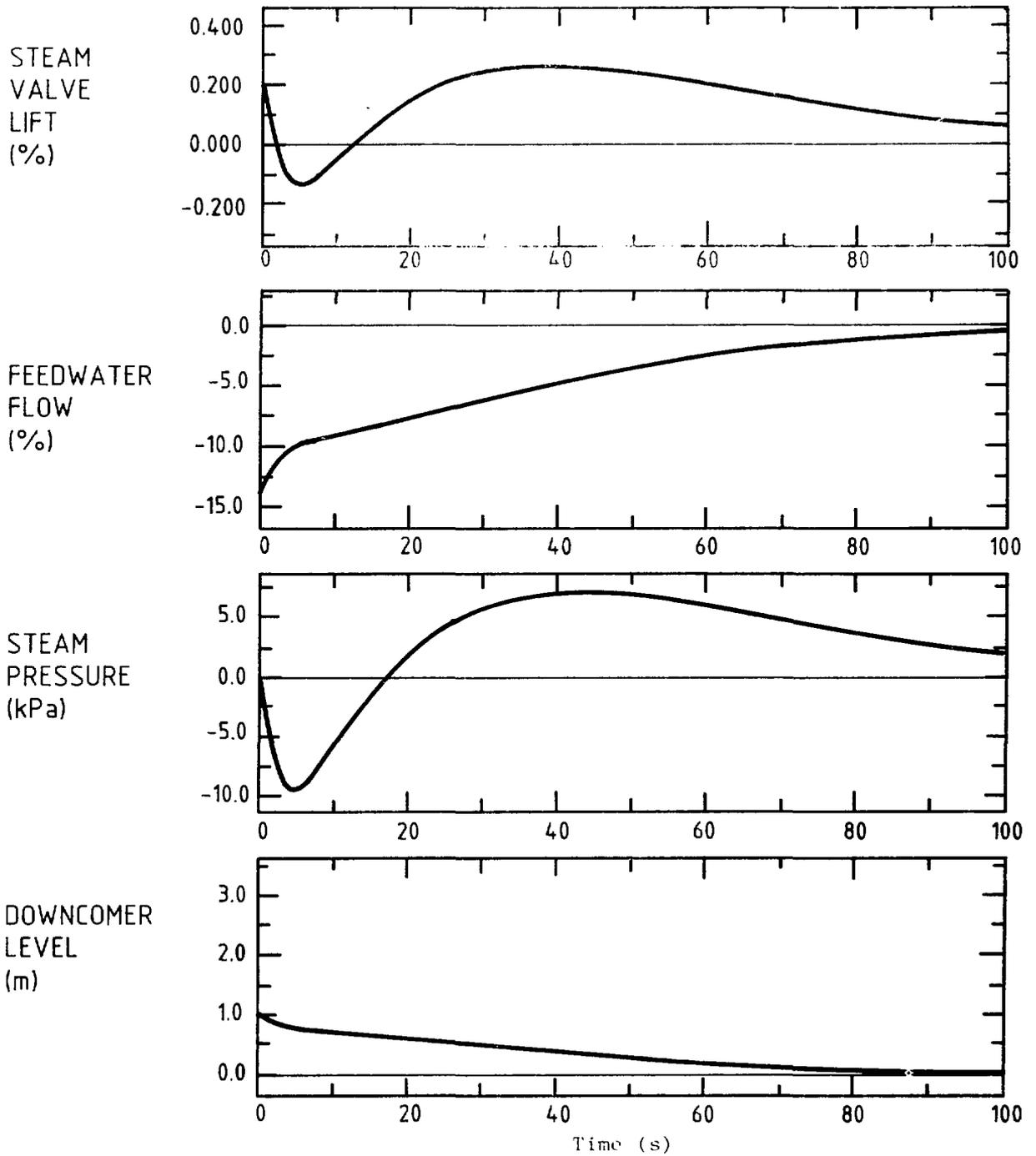


FIGURE 11b Closed-Loop Response to initial Offset of Downcomer Level

## 4. HOW MVPACK DOES IT

### 4.1 Interaction between MVPACK and User

MVPACK is an interactive program; it depends on, and prompts for, input from the user using a sequence of questions and answers. Answers may consist of a word or a series of numbers.

If the answer to a question concerning either the use of the design tools or the operation of the program is not readily apparent, there is a HELP facility that can be invoked to establish the choices of response options available. As well as being a convenient reference guide, the users' own experiences can be incorporated into the HELP facility as additional messages programmed into the computer code.

The simplicity of the dialogue between the program and the user, and the existence of the HELP facility, allow the designer to concentrate on the design problem at hand rather than the use of the program. This is especially useful for first-time users of MVPACK, or for novice designers who will be assisted by this user-friendly software.

The designer can use different modules in MVPACK to apply different design algorithms to a problem. The output from the different methods can be compared and a choice made as to which is the best. This capability can be exploited by the designer to lead to the best practical controller for a given system.

### 4.2 Mathematical Support

The design and analysis algorithms of multivariable control systems require extensive mathematical support. MVPACK has several features to meet this requirement.

#### 4.2.1 Mathematical Libraries

MVPACK includes three mathematical libraries, dedicated to real, complex and polynomial matrix arithmetic. Each library is made up of a collection of specialized subroutines. The variety and power of the functions and operations implemented in the mathematical libraries allow MVPACK design and analysis algorithms to be built rapidly with existing blocks. New subroutines can be added to the libraries and existing subroutines can be upgraded with new, improved numerical methods. Most of the facilities implemented in the libraries can also be used directly, through the CALC supervisor, by a control system designer to perform off-line matrix operations before and/or after the design of a controller, i.e., the design can be performed in a step-by-step manner.

#### 4.2.2 Error Handling

In MVPACK, errors are generally detected when either they occur or an attempt is made to use an erroneous result. All library routines check matrix validity ensuring that processing of an invalid matrix does not occur. The routines in the mathematical libraries check matrix types and

dimensions before performing any calculations. Detected errors are reported as coded numbers followed by a message and a traceback of the subroutine call chain.

These error handling mechanisms enhance MVPACK's reliability and protect the operation of the package against both programmer and user errors.

#### 4.2.3 Advanced Mathematical Facility

To speed up design and implementation of application algorithms as well as to improve programming efficiency, an advanced mathematical facility is implemented in MVPACK. This facility, under continuous development, is a collection of independent modules. Each module implements a complex operation which has potential applications within several distinct design or analysis algorithms.

The modules currently available are given in Table 6.

TABLE 6  
ADVANCED MATHEMATICS MODULES

---

Module	Function
MLUCAN, MLUEN	Transform the system defined by (A,B) to Luenberger canonical form using the method of Aplevich
ROSENH and ROSTEP	Search for the maximum of the function $U(x)$ . The algorithm is the Rosenbrock constrained minimization procedure.

---

#### 4.3 Data Structure

A comprehensive data structure is essential for the computer-aided design of a control system. The data structure must facilitate the creation and analysis of sets of matrices representing systems. It must store real, complex, polynomial, and rational polynomial matrices and permit efficient access by processing routines. Matrices must be readily transferable between modules. All information about a matrix, a system or the state of a program must be stored in the database.

The data structure of MVPACK includes in-core, disk file and task control structures. The structures chosen meet the requirements of ease of coding; efficient use of memory; natural interfacing to disk files, so data can be saved, deleted, recalled, and transferred to other programs; and flexible control of programs, with communication between them.

MVPACK is based on the methods of structured programming and top-down design. Each structure consists of stored data and functions to create, delete, copy and access the data. These functions are designed in such a way that the structure is concealed from the application program.

#### 4.3.1 In-Core Structure

There is a limited program address space available in non-virtual memory computers. This problem is overcome by allocating space for data only as required, thus eliminating the need to dimension arrays for the anticipated maximum size. When arrays are no longer needed, they are deleted and the space is reused. This dynamic memory structure consists of a directory and a large data area. The directory is a doubly-linked list so any entry can be easily deleted.

Unused directory nodes form a singly-linked list called the free list. When space is allocated, a node is removed from the free list, and is added to the end of the directory.

Important features of this dynamic memory system are efficient use of memory and inherent protection capability. It is efficient because space is allocated as required, and reused when possible. Efficiency is also improved by matrix operation functions capable of generating results in situ.

#### 4.3.2 Disk Data Structure

The disk data structure is a modular database package for efficient and reliable storage of named entities. It allows MVPACK to have a calculator-like appearance with natural transitions between modules, and permits the use of developed modules as subroutines of new ones.

The disk data structure stores sets of data representing systems, and permits recall of individual items by name. Typical named entities are real matrices, rational polynomial matrices, and sequences of complex matrices.

#### 4.3.3 Task Control Structure

The task control structure makes MVPACK easy to use and convenient for program development. It allows programs to be implemented as separate tasks, so replacement and introduction of new programs is simple. Programs can call each other like subroutines, with natural nesting to any depth.

#### 4.4 Hierarchic Modular Structure

Developments in control system theory are on-going and it is important that new methods could be incorporated into existing software as they become available. It must be simple to add and replace algorithms, as well as reuse existing code as subroutines in new modules. The modular and hierarchic arrangement of MVPACK helps meet these requirements of software flexibility and expandability, as it facilitates the development, testing and addition of new design algorithms.

As Figure 12 shows, MVPACK is a set of modules arranged hierarchically where control modules called "supervisors" accept commands interactively from the user, and call the module that will process the command. Functions specific to the operating system are localized in the top level (labelled MVPACK in the following diagram) of the hierarchy so they can be easily adapted to new operating environments.

The GEN supervisor can be used to increase the speed with which development and debugging can be carried out. The user can avoid entering processing modules through their supervisors by calling GEN, and from it, directly implement any module. The user can copy modules which require testing into GEN, test them, and when they are working satisfactorily, replace the original versions with them. GEN is a powerful tool but it is advisable that less-experienced users of MVPACK use the hierarchic structure of supervisor and processing modules.

#### 4.5 Coding of MVPACK Modules

Modules are partitioned into four levels which parallel the data structure: driver, algorithm, interface and primitive.

The top level is the driver level, typically the main program. It initializes the data structure, interacts with the control of the hierarchic structure and may interact with the user.

The algorithm level contains the actual design algorithms. It is typically coded as several subroutines, each performing part of the calculation.

To interface the computing primitives to the algorithm level, each primitive has an interface-level routine.

The primitive level is the lowest level at which the code resembles conventional FORTRAN. Existing programs can be included in MVPACK as primitives, by providing them with suitable interfaces.

MVPACK modules are coded in Ratfor, a structured preprocessor for FORTRAN. Appropriate use is made of the Ratfor control structure which facilitates modification and portability. It is recommended that users make full use of library routines. In the VAX environment, new modules may also be coded using the structured VAX FORTRAN.

#### 4.6 Graphics

The MVPACK modules that generate Nyquist, Bode and time-response plots are NYQ, BODE and TPL, respectively. To ensure the portability of MVPACK, these modules use the Tektronix PLOT 10 GKS graphics software. The Graphical Kernel System (GKS) is a standard set of high-level graphics functions accepted by both the International Standard Organization and the American National Standards Institute, and PLOT 10 GKS is an implementation of this software at level 2b. Its features include geometric transformations and picking operations, graphic input functions, full control over workstations, and a full range of text fonts for fancy labelling.

LEVELS

SUPERVISORS

MODULES

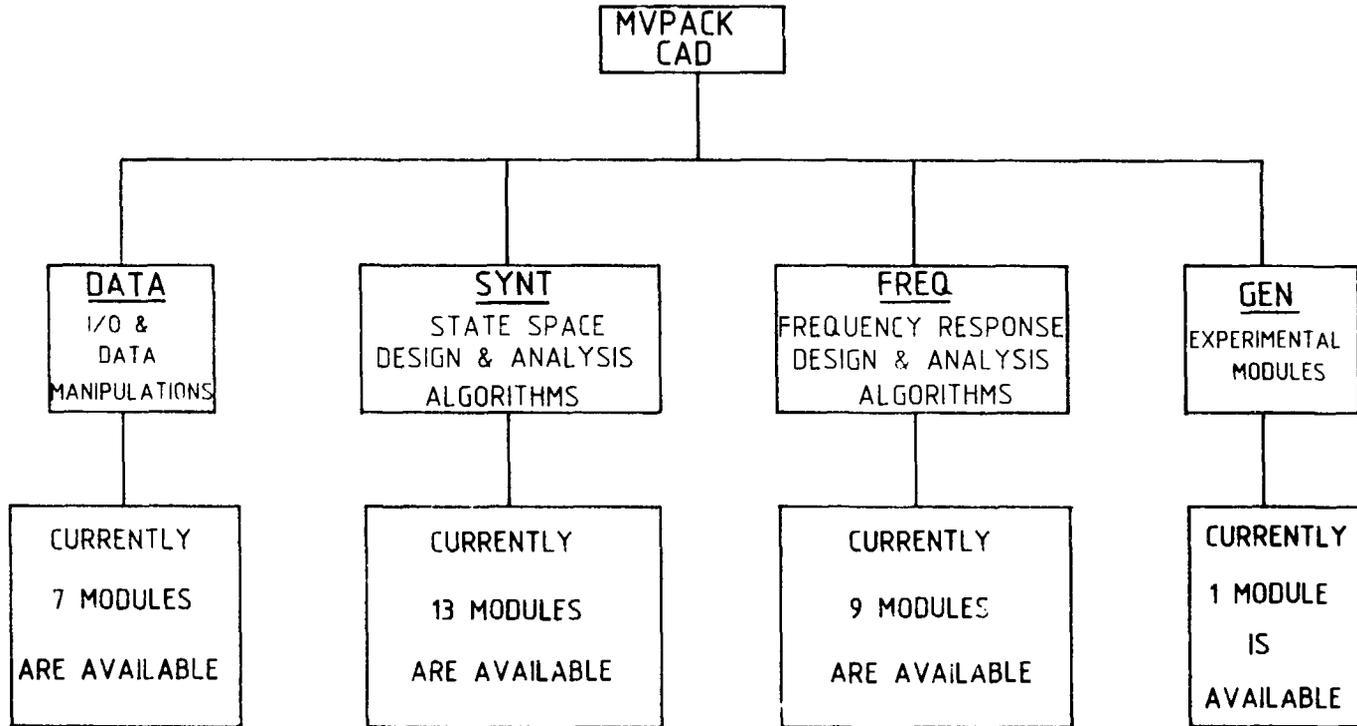


FIGURE 12. MVPACK CAD hierarchical structure.

## 5. SUMMARY

Control systems for most plants are usually designed using conventional single-variable methods that treat each control loop independently. Interactions between these loops are accounted for empirically by tuning the controllers to achieve the desired results. As plants increase in size and complexity, and are required to operate closer to maximum capacity for greatest economic returns, these dynamic interactions become more important and their existence makes the extrapolation of these techniques unreliable. However, improved process regulation can be achieved when multivariable design and analysis methods are used, which view the plant as a fundamentally interacting system with numerous inputs and outputs vying to meet often conflicting objectives.

MVPACK is a user-friendly software package that combines a powerful capability to design and analyze complex multivariable control systems with an ease of application. MVPACK is a set of interactive modules that appears to the user as a high-level calculator. It is composed of a database, interaction methods, a mathematical library, and an extensive collection of design algorithms that includes: order reduction of linear models; pole shifting with output feedback; modal control; optimal control with stochastic state estimation; and the inverse Nyquist array method.

## 6. ACKNOWLEDGEMENTS

The assistance of the technical writer, J. McMonagle, in the preparation of an early draft of this report is appreciated, as is C. Lupton's skillful laying out and typing of the final version. MVPACK was developed by P.D. McMorran and numerous coworkers.

## 7. REFERENCES

- [1] P.D. McMorran, "Multivariable Control in Nuclear Power Stations: Survey of Design Methods", AECL-6583, 1979 December.
- [2] IEEE Control Systems Magazine 2 (4), 1982 December; Proceedings IEEE 72 (12), 1984 December. (Both issues feature computer-aided control system design.)
- [3] P.D. McMorran, "Data Structures for Computer Aided Design of Multivariable Control Systems", Proceedings of the Second IFAC Symposium, West Lafayette, Indiana, 1982 September 15-17, pp. 259-266.
- [4] H.W. Hinds, "The Application of MVPACK to the Design of Multivariable Control Systems for a Nuclear Steam Generator", presented at the Power Plant Digital Control and Fault-Tolerant Microcomputers Seminar, Phoenix, Arizona, 1985 April 9-12.
- [5] P.D. McMorran and T.A. Cole, "Multivariable Control in Nuclear Power Stations: Modal Control", AECL-6690, 1979 December.
- [6] M.R.A. Ali, "Lumped Parameter, State Variable Dynamic Models for U-Tube Recirculation Type Nuclear Steam Generators", Ph.D. Thesis, The University of Tennessee, Knoxville, 1976 August.
- [7] H.H. Rosenbrock, Computer-Aided Control System Design, Academic Press, London, 1974.

ISSN 0067 - 0367

To identify individual documents in the series we have assigned an AECL- number to each.

Please refer to the AECL- number when requesting additional copies of this document

from

Scientific Document Distribution Office  
Atomic Energy of Canada Limited  
Chalk River, Ontario, Canada  
K0J 1J0

Price: \$3.00 per copy

ISSN 0067 - 0367

Pour identifier les rapports individuels faisant partie de cette série nous avons assigné un numéro AECL- à chacun.

Veuillez faire mention du numéro AECL- si vous demandez d'autres exemplaires de ce rapport

au

Service de Distribution des Documents Officiels  
L'Énergie Atomique du Canada Limitée  
Chalk River, Ontario, Canada  
K0J 1J0

Price: \$3.00 per copy

© ATOMIC ENERGY OF CANADA LIMITED, 1986

0168-86