(NASA-CR-146531) ALTERNATIVES FOR JET ENGINE CONTROL Final Technical Report, 1 Mar. 1975 - 29 Feb. 1976 (Notre Dame Univ.) 42 p HC A03/MF A01 CSCL 21E N77-12058 Unclas G3/07 56703

Department of
ELECTRICAL ENGINEERING
UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

Revised Final Technical Report

to the

National Aeronautics and Space Administration

on

NASA Grant NSG-3048

ALTERNATIVES FOR JET ENGINE CONTROL

(March 1, 1975 - February 29, 1976)

Under the Direction of

Dr. R. J. Leake
Dr. M. K. Sain

Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ALTERNATIVES FOR JET EINGINE CONTROL

NASA Grant NSG-3048

ABSTRACT

The objective of the research under this Grant has been the development of approaches which are alternatives to current design methods which rely heavily on linear quadratic and Riccati equation methods. The main alternatives were classified into two broad categories:

Local Multivariable Frequency Domain Methods

Global Nonlinear Optimal Methods

The progress made toward the original research objectives within the one year grant period is described briefly in this report. More detailed information is available in the following works, which were essentially completed during the grant period.

(1) T. C. Brennan and R. J. Leake, "Simplified Simulation Models for Control Studies of Turbojet Engines," Notre Dame Electrical Engineering Technical Report No. EE-757, 88 pages, November 1975.

(2) J. C. Shearer, "An IBM 370/158 Installation and User's Guide to the DYNGEN Jet Engine Simulator," Notre Dame Electrical Engineering M.S.E.E. Thesis, 44 pages, November 1975.

(3) R. J. Leake and J. G. Allen, "Optimal Regulators and Follow-Up Systems Determined from Input-Output Signal Monitoring," Notre Dame Electrical Engineering Memorandum No. 7403, 16 pages, October 1975.

(4) R. R. Gejji and M. K. Sain, "Polynomial Techniques Applied to Multivariable Control of Jet Engines," Notre Dame Electrical Engineering Technical Report No. EE-761, 118 pages, March 1976.

(5) V. Seshadri and M. K. Sain, "Multivariable System Compensation Including Characteristic Methods for Jet Engine Control," Notre Dame Electrical Engineering Technical Report No. EE-763, 114 pages, May 1976.

(6) M. K. Sain, et. al., "Alternative Methods for the Design of Jet Engine Control Systems," Proceedings of the 1976 Joint Automatic Control Conference, Purdue University, pp. 133-142, July 1976.

CONTENTS

PART I

MULTIVARIABLE FREQUENCY-DOMAIN ALTERNATIVES FOR JET ENGINE CONTROL

a. Polynomial Techniques
R. R. Gejji and M. K. Sain

b. Characteristic Loci Techniques
V. Seshadri and M. K. Sain

Ia. POLYNOMIAL TECHNIQUES

Completion of the first year's work on Grant NSG-3048, occurring February 29, 1976, was simultaneous with the completion of preliminary investigations on polynomial techniques applied to jet engine control. This work was carried out by R. R. Gejji under the direction of M. K. Sain and served also as Mr. Gejji's Master of Science thesis activity. As a research activity, the effort was jointly supported by this grant and by the National Science Foundation under Grant GK-37285. Effort under this grant was focused upon the development of a realistic, practical application for the polynomial design method in the area of jet engine control, while the NSF support was used to provide some flexibility for theoretical and algorithmic extensions of polynomial concepts as they became needed.

Before presenting an abstract of this work, it may be useful to provide a brief notion of the types of problems and methods involved.

Basically, polynomial techniques have to do with the solution of transfer function equations of the form

$$A(s)X(s) = B(s)$$

where A(s) and B(s) are known matrices each of whose elements is a ratio of polynomials in s, with real coefficients. X(s) is an unknown matrix in the same class and is to be determined. For feedback control of jet engines, A(s) and B(s) are determined from engine data around the operating point of interest and from specifications on the performance of the feedback control system; and X(s) embodies information about the control compensations required to satisfy the specifications. If it were merely desired to solve exhaustively the linear equation above, it would be possible to take advantage of the field structure of ratios of polynomials and proceed according to the classical introductory methods of linear equations. Unfortunately, this is not the case, since in control problems there is also k-y interest in

(i) whether X(s) is physically realizable,

(ii) where the poles of X(s) are located in the complex plane,

(iii) and, if more than one X(s) is possible, how many dynamical elements are required to construct X(s).

These issues immediately remove the solution from an elementary classification. One approach which has developed in recent years to handle issues such as (i), (ii), and (iii) is what we have been calling the polynomial technique. The idea can be simply illustrated by writing

$$A(s) = N_A(s)/a(s),$$

$$B(s) = N_B(s)/b(s),$$

$$X(s) = N(s)D^{-1}(s)$$

where N_A, N_B, N and D are matrices of polynomials and a and b are polynomials. The representation for X(s) is called a matrix fraction and is quite helpful in answering advanced questions about X. With these conventions, the design equation becomes

$$[b(s)N_A(s) \vdots -a(s)N_B(s)]\begin{bmatrix} N(s) \\ D(s) \end{bmatrix} = [0]$$

so that pairs of corresponding columns from N and D can be understood algebraically as solutions to a homogeneous equation having a polynomial coefficient matrix.

The work discussed in the following abstract concerns the specification of A(s) and B(s), the calculation of homogeneous solutions to this polynomial equation, and the control theoretic interpretation of these solutions.

Abstract

R. R. Gejji and M. K. Sain, "Polynomial Techniques Applied to Multivariable Control of Jet Engines, Technical Report No. EE-761, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana 46556, March 22, 1976.

One way to approach the design of linear multivariable control systems is by means of exact model matching, where desired closed loop performance is expressed in terms of a specified transfer function matrix. One questions which is often raised about such an approach involved the practicality of specification of such a desired, closed loop performance matrix. Another, related, question concerns the possibility of determining the existence of compensators to achieve such a performance, as well as giving a finite enumeration of them and selecting those which are, for example, minimal or stable.

This study addresses itself to these points, in reverse order. Research by Sain in 1975 has established that determination of the existence of compensators and a finite enumeration of them is possible within the context of free modules over polynomial rings. Algorithms to accomplish this in theory are also available. The first purpose of this work was to construct straightforward computer software to check the workability of these theoretical algorithms. This has been accomplished in both FORTRAN and PL1, with listings for the former included herein. It was found that these programs were successful on the sort of small problems which often appear in the literature. The second purpose of the work was to evaluate these first-generation software efforts on a realistic practical problem. This has been accomplished in the context of multivariable control of models derived from the F100 turbofan

engine by Michael and Farrar. It was established that specification of desired closed loop performance matrices is a reasonable idea for this application, and, in fact, that in at least one case it has already been done. This means that exact model matching is not a vacuous idea, at least for jet engines. Efforts were then expended to determine if the first generation software would compute compensators, which had been proven to exist. The results here were promising but inconclusive. Though the software would yield answers of the degree expected (a definitely nontrivial achievement), eigenvalue tests and other checks initiated to verify accuracy were decidedly disappointing.

Two conclusions are possible: (1) that some oversight remains undetected in the software or (2) that straightforward computer software is unequal to this application. The second of these possibilities appears to be the more probable at this time, though the first cannot be eliminated with certainty. Studies are under way to develop more sophisticated software.

In any event, the results in terms of problem formulation are promising enough, and the computer results close enough to suggest that further work on this problem would be worthwhile.

Remark

Contained in this report are two results which are ready for presentation to the technical community. The first result, on discovering and fully formulating what seems to be an important exact model matching problem in jet engine control, is being prepared for the Nineteenth Midwest Symposium on Circuits and Systems, which will take place in the summer of 1976. The second result, on development of FORTRAN and PL1 software for this class of problems, is being prepared for the Fourteenth Allerton Conference on Circuit and System Theory in the fall of 1976.

Remark

The sorts of academic polynomial equations previously solved in the literature have, in the experience of the authors, taken at least an hour or two to solve by hand. Moreover, hand solution is tedious and error-proven. The fact that this software works successfully on such problems is, thus, nontrivial and represents an improvement of roughly two orders of magnitude in solution time for such cases. The necessity for additional software development for the jet engine example can be regarded as a temporary delay rather than as an essential obstacle. Further results in this regard are expected to be achieved over the summer.

Remark

There is also one result on polynomial methods in the next section.

Ib. CHARACTERISTIC LOCI TECHNIQUES

In a manner similar to the studies on polynomial techniques, completion of the first year's grant work occurred almost simultaneously with the completion of preliminary investigations on characteristic loci techniques applied to jet engine control. This work was carried out by V. Seshadri under the direction of M. K. Sain and also served as Mr. Seshadri's Master of Science thesis activity. As a research activity, the effort was jointly supported by this grant and by the National Science Foundation under Grant GK-37285. Effort under this grant was focused on the application of the characteristic loci technique to realistic numerical data associated with jet engine models, while the NSF support allowed some basic investigation into the nature of characteristic loci themselves.

A word or two on the meaning of characteristic loci may be helpful before a reading of the abstract which follows.

Recall the Nyquist method for analysis of a plant having one input and ont output. The strategy is to study the behavior of the plant transfer function g(s) as s travels an appropriately chosen Nyquist contour roughly enclosing the right half of the s-plane. To gain appreciation for MacFarlane's generalization of Nyquist's idea, it is convenient to observe that the transfer function g(s) actually defines the eigenvalue of the plant operator, as can be seen from

$$[g(s)][1] = g(s)[1],$$

where [1] is the eigenvector of the operator [g(s) and G(s) is its eigenvalue. In generalizing to the case of a multivariable plant G(s), having an equal number of inputs and outpus, MacFarlane determined to make Nyquist plots of the eigenvalues of G(s). These plots are called characteristic loci.

Abstract

V. Seshadri and M. K. Sain, "Multivariable System Compensation Including Characteristic Methods for Jet Engine Control," Technical Report No. EE-763, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana 46556, May 6, 1976.

This study addresses two possible approaches to the problem of linear multivariable control system design. The first approach belongs to the class of exact model matching designs; the second approach makes use of the characteristic methodologies of MacFarlane.

In the exact model matching case, we consider the problem of finding stable solutions G(s) to the transfer function matrix equation

$$G_1(v)G(v) = G_2(v)$$

where all matrices have their elements in the quotient field R(v) formed from the ring R[v] of polynomials in the indeterminate v with coefficients in the real number field R. Our viewpoint on this equaton is the free R[v]-module structure presented by Sain in 1975. Previous work has made clear how to answer questions such as whether G(v) can be proper and minimal. Herein we present some tools and partial results oriented to the question of whether G(v) can be stable.

In the MacFarlane methodologies case, we have considered the problem of compensating a two-input, two-output, five-state jet engine model in the class described by Michael and Farrar. Inputs chosen were main burner fuel flow and jet exhaust area; outputs were thrust and high turbine inlet temperature. Software was developed for constructing characteristic loci and a number of related parameters, and an effort was made to use the ideas of noninteraction, integrity, and accuracy to achieve an introductory jet engine control design.

The results are positive in nature, although we feel that more attention is needed in this theory toward the actual design of reasonable compensators.

Remark

This work was essentially complete by February 29, 1976. The May 6 date merely reflects delays induced by allowing time for thesis readers to provide remarks and for scheduling of the thesis defense.

Remark

The main part of this work concerns the characteristic locus; the other result on polynomial methods relates to the previous section and consumed a smaller time of the research.

Remark

Two portions of the work contained in this report are being prepared for presentation to the technical community. The first part, about assignment of the poles of the solution, is being prepared for the Fourteenth Allerton Conference on Circuit and System Theory. The second part, dealing with the characteristic loci studies, is being prepared for the Nineteenth Midwest Symposium on Circuits and Systems. A very interesting feature of this presentation, in the opinion of the authors, is the conjecture that difficulties in applying the basic method to jet engine control are intuitively due to interaction among the inputs and outputs. There seems to be a possibility that such interactions can be related to the fact that certain jet engine characteristic loci do not separate into self-contained sub-loci as sometimes occurs in less complicated examples. If this should indeed later turn out to be the case, it would represent a substantial new involvement of topological concepts in feedback control systems, which have traditionally been more heavily influenced by algebra.

Remark

The conclusions that more attention to compensator design is needed in this theory, at least for the jet engine examples under study, are not very prohibitive. New investigations to overcome this deficiency are now being planned.

PART II

GLOBAL NONLINEAR METHODS FOR JET ENGINE CONTROL

a. Analog Simulation of Single Spool Turbojet
T. C. Brennan and R. J. Leake

b. Installation of DYNGEN Jet Engine Simulator
J. C. Shearer and R. J. Leake

c. Optimal Control Studies
R. Basso and R. J. Leake

IIa. ANALOG SIMULATION OF SINGLE SPOOL TURBOJET

This work represents an attempt to characterize the essential dynamical characteristics of a simple single-spool turbojet engine through simulation of low order system models on an Electronic Associates TR-48 analog computer. The objective was to gain insight into the most important dynamical constraints of such an engine, which can be applied to control studies of more advanced engines.

Included is a derivation of a seventh order simplified simulation model, a derivation of an even simpler third order model, and simulation results from each. Details are given in reference (1) listed in the abstract of this report.

The control problem studies is one of getting from "Windmill" (zero fuel flow equilibrium) to "Design" (a high thrust equilibrium) while taking into account surge margin and turbine inlet temperature constraints. Several control schemes were investigated.

The principal constraints were imposed by the limited nonlinear equipment available on the TR-48 analog computer. The main simplifications involved the extensive use of linear approximations, single stage compressor dynamics, a linear compressor map, assumptions of a choked nozzle condition, and certain empirical relations based on design point equilibrium data available. Additional simplification was achieved by limiting the model to a condition of 20,000 ft. altitude at Mach number .8. We shall term this model the "Drone" system.

The Drone System

Throughout the development, P stands for pressure in newtons/meter^2, T stands for temperature in degrees Kelvin, $\dot{\omega}$ represents mass flow rate in kilograms/second, and ρ is density in kilograms/meter^3. N is rotor speed in revolutions/minute. Using a single stage compressor model the main dynamical equations are

Auxiliary Relations

$$P_2' = R\rho_c T_2'$$

$$P_4 = R\rho_B T_4$$

$$\Delta P_4 = K_B \dot{\omega}_4^2 / P_3$$

$$\dot{\omega}_3 = \dot{\omega}_4$$

Non-Linear Functions

Turbine

$$\dot{\omega}_5 = f_1(N, T_4, P_4, P_5)$$

$$T_5 = f_2(T_4, P_4, P_5, N)$$

Nozzle

$$\dot{\omega}_8 = f_3(T_5, P_5)$$

Compressor

$$T_3 = f_4(T_2', N, P_4)$$

$$P_3 = f_5(P_2', T_2', \dot{\omega}_3, \dot{\omega}_2)$$

Normalized Seventh Order Model

Using a number of assumptions detailed in reference (1) of the abstract

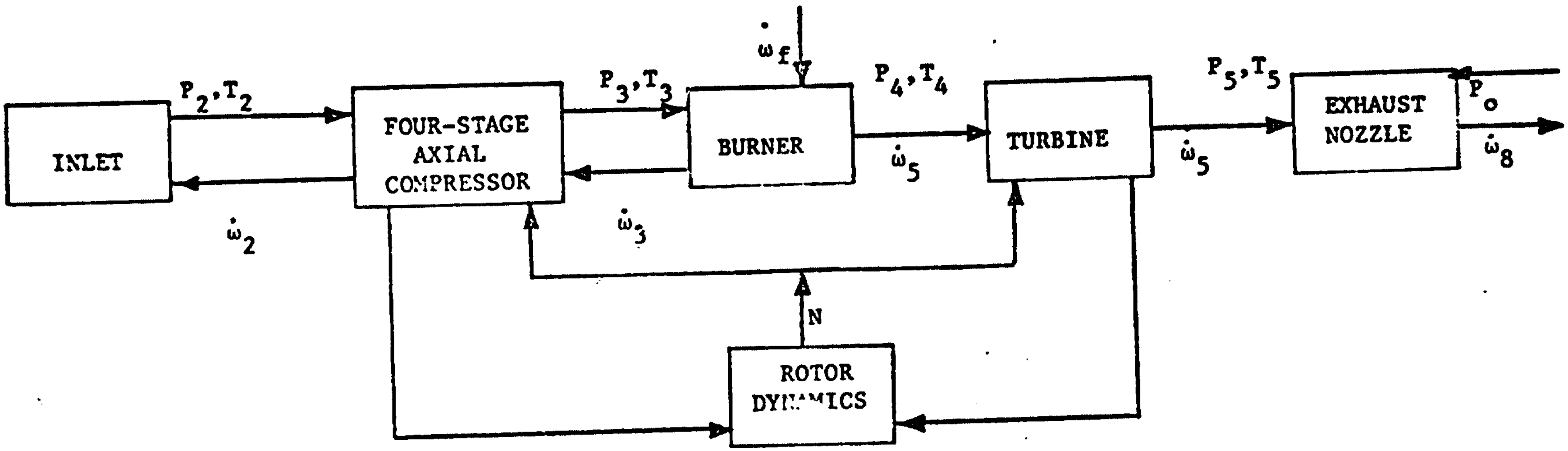

FIGURE 1. ROUGH SIMULATION DIAGRAM FOR DRONE SYSTEM

of this report, and normalizing about a selected design point the seventh order numerical model becomes:

State Equations:

$$\frac{d\hat{P}_2'}{dt} = 51.52\ (\hat{\omega}_2 - \hat{\omega}_3)$$

$$\frac{d\hat{\omega}_2}{dt} = 2188.27\ (1 - \hat{P}_2')$$

$$\frac{d\hat{\omega}_3}{dt} = 8191.36\ [1.05787\hat{P}_3 - \hat{P}_4 - .05787\ \frac{\hat{\omega}_3{}^2}{\hat{\rho}_3}]$$

$$\frac{d\hat{P}_4}{dt} = .93586\ \frac{\hat{P}_4\hat{\omega}_f}{\hat{\rho}_B} + 31.486\hat{\omega}_f + 21.435\hat{\omega}_3\hat{T}_3 - 53.86\ \frac{\hat{P}_4{}^2}{\hat{\rho}_B}$$

$$\frac{d\hat{\rho}_B}{dt} = 37.78\hat{\omega}_3 - 38.448\ \hat{P}_4 + .66849\hat{\omega}_f$$

$$\frac{d\hat{P}_5}{dt} = 61.97\ \frac{\hat{P}_4}{\hat{\rho}_B}\ [\hat{P}_4 - \hat{P}_5\hat{\Theta}]$$

$$\frac{d\hat{N}}{dt} = \frac{1}{.305\hat{N}}\ \{\frac{\hat{P}_4}{\hat{\rho}_B}\ [3.12\hat{P}_4 - 2.7361\hat{P}_5\hat{\Theta}]$$

$$+ [.688013\hat{\omega}_2 - 1.0715\ \hat{T}_3\hat{\omega}_3]\}$$

Other Relations:

$$\hat{\omega}_3 = \hat{\omega}_4 \qquad \hat{P}_4 = \hat{\rho}_B\hat{T}_4 \qquad \hat{P}_2 = 1 \qquad \hat{T}_2 = \hat{T} = 1$$

$$\hat{\dot{\omega}}_5 = \hat{P}_4$$

$$\hat{\dot{\omega}}_8 = \hat{P}_5\Theta$$

$$\hat{T}_5 = \hat{T}_4$$

$$\hat{T}_3 = .64212 + .35788\hat{N}^2$$

$$\hat{P}_3 = 4.394\hat{N} - 3.394\hat{\dot{\omega}}_3$$

$$F = \hat{\Theta}\ \{1.5486\hat{P}_5 - .5486\}$$

$$V_1 = V_4 = V_5 = A_1 = A_4 = .1 \qquad I = .0305$$

Normalized Third Order Simulation Model

State equations:

$$\frac{d\hat{P}_4}{dt} = \hat{\dot{\omega}}_f\ (.93586\hat{P}_4/\hat{\rho}_B + 31.486) + 21.435\hat{\dot{\omega}}_3\hat{T}_3 - 53.86\ \hat{P}_4^2/\hat{\rho}_B$$

$$\frac{d\hat{\rho}_B}{dt} = 37.78\hat{\dot{\omega}}_3 - 38.448\hat{P}_4 + .66849\hat{\dot{\omega}}_f$$

$$\frac{d\hat{N}}{dt} = \frac{1.258}{\hat{N}}\ (\hat{P}_4{}^2/\hat{\rho}_B - \hat{\dot{\omega}}_3 N^2)$$

Other Relations:

All relations equations in the previous seventh order model are valid, plus

$$\hat{P}_2' = 1 \qquad \hat{\dot{\omega}}_2 = \hat{\dot{\omega}}_3 \qquad \hat{P}_4 = \hat{P}_5\hat{\theta}$$

$$\hat{\dot{\omega}}_3 = 1.3009\hat{N} - .13982(\hat{P}_4 - \sqrt{\hat{P}_4^{\,2} + .41688\hat{N}^2 - .0899\hat{P}_4\hat{N}})$$

Equilibrium Conditions

Equilibrium conditions for the normalized seventh order and third order models are the same. The "Design" equilibrium occurs when all normalized state variables are unity, and this corresponds to a design point of Mach .8 and 20000 ft. Other equilibrium points must be calculated using a successive approximation procedure to solve the nonlinear equations which result by setting all derivatives zero. In particular, we define "Windmill" as the equilibrium which occurs when fuel flow $\dot{\omega}_f = 0$ and nozzle area $\hat{\theta} = 1$.

The algorithm used to calculate equilibrium conditions is as follows:

1. Set values of $\dot{\omega}_f$ and θ (the controls).
2. As initial estimates, set

$$\hat{P}_2 = 1,\ \hat{N} = \hat{P}_4 = .5(\hat{\dot{\omega}}_f + 1)$$

3. Set in order

$$\hat{\dot{\omega}}_3 = 1.3009\hat{N} - .13982(\hat{P}_4 - \sqrt{\hat{P}_4^{\,2} + .41688\hat{N}^2 - .0899\hat{P}_4\hat{N}})$$

$$\hat{T}_3 = .64212 + .35788\hat{N}^2$$

$$\hat{P}_3 = 4.394\hat{N} - 3.394\hat{\dot{\omega}}_3$$

$$\hat{P}_4 = (\hat{\dot{\omega}}_f + 56.515\hat{\dot{\omega}}_3)/57.515$$

$$\beta_B = (53.86\hat{P}_4^{\,2} - .93586\hat{P}_4\hat{\dot{\omega}}_f)/(21.435\hat{\dot{\omega}}_3\hat{T}_3 + 31.486\hat{\dot{\omega}}_f)$$

$$\hat{T}_4 = \hat{P}_4/\rho_B$$

$$\hat{P}_5 = \hat{P}_4/\hat{\Theta}$$

4. Set

$$\hat{N} = \sqrt{\hat{P}_4^{\,2}/\hat{\rho}_B\hat{\dot{\omega}}_3}$$

and return to 3, until convergence is achieved.

This algorithm gave five place convergence in twenty iterations and showed a very nearly linear operating line relation of approximately

$$\hat{P}_3 \cong 1.0263\hat{\omega}_3 - .0263$$

or equivalently (using the linear compressor relation $\hat{P}_3 = 4.394\hat{N} - 3.394\hat{\omega}_3$)

$$\hat{\omega}_3 \cong .99405\hat{N} - .0095$$

Compressor Map and Surge Lines

The equation used to determine the approximate compressor map of Figure 2 is

$$P_3/P_2' = 5N/10^4 - 2.94215\omega_3\sqrt{\Theta_2}/\delta_2$$

where

$$\Theta_2 = T_2/288.3 = 280.6/288.3 = .97329$$
$$\delta_2 = P_2/10.1325 \times 10^4$$

Using these relations and the equilibrium values

$$P_{3E} = 28.076 \times 10^4$$

$$P_{2E} = 7.09 \times 10^4$$

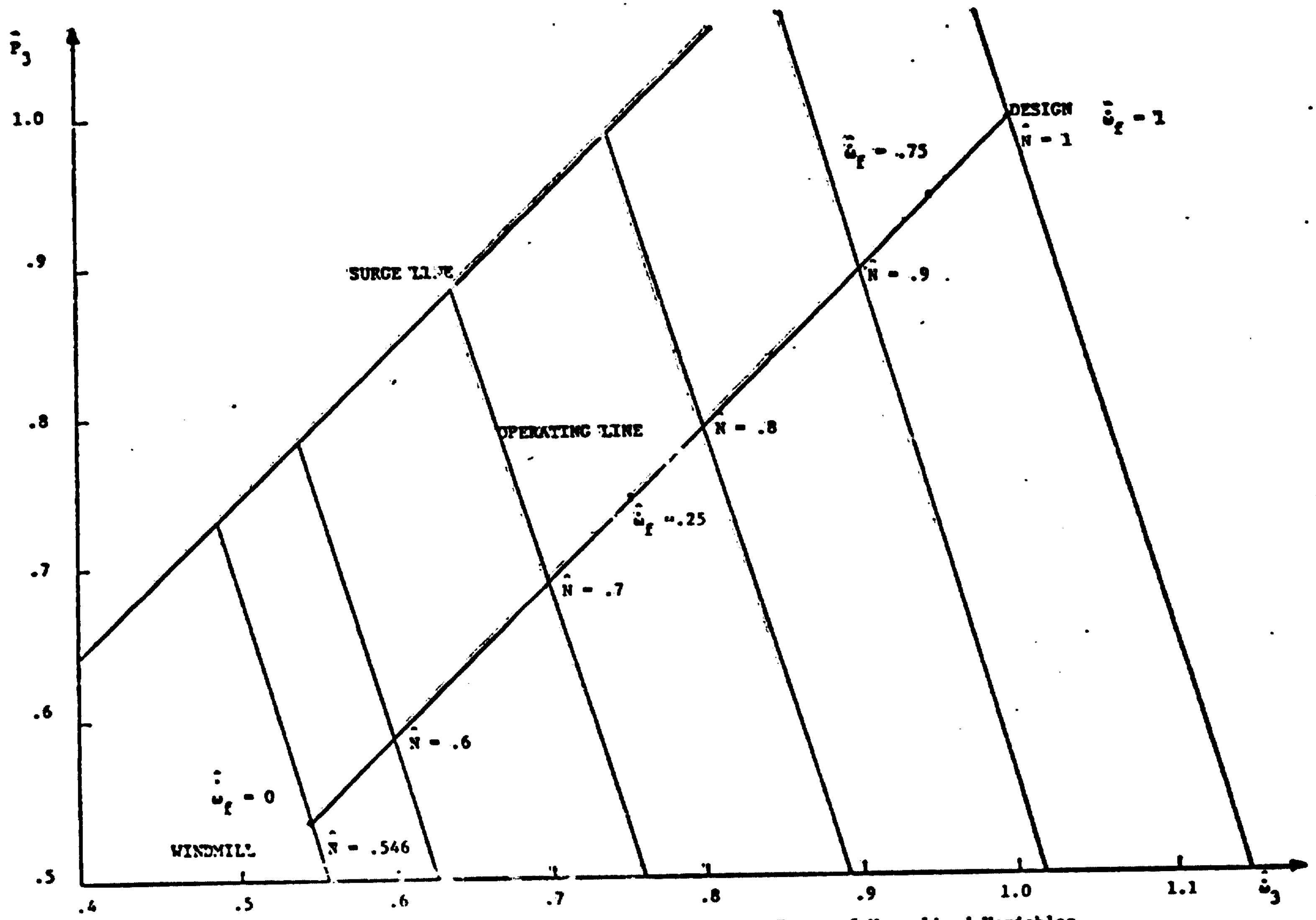

Figure 2. Linearized Compressor Map in Terms of Normalized Variables.

$$\hat{\omega}_{3E} = 3.24$$

$$N_E = 3.48 \times 10^4$$

we obtain

$$\hat{P}_3 = 4.394\hat{P}_2'\hat{N} - 3.394\hat{\omega}_3$$

This relation is graphed in Figure 2 for $\hat{P}_2' = 1$ along with the operating line of equilibrium points.

We have chosen to define the surge line as a line running parallel to and above the operating line. This is rather arbitrary, but preserves the qualitative nature of the problem of getting from Windmill to Design without transgressing on the surge characteristic. Indicated on Figure 2 is a surge line specified by the relation

$$\hat{P}_3 = 1.0263\hat{\omega}_3 + .24105$$

Linear Normalized Systems

In order to estimate the dynamical modes of the various models, and to provide models for linear control studies, linear representations of the form

$$\frac{d\delta\hat{x}}{dt} = A\delta\hat{x} + B\delta\hat{u}$$

were determined, where $A = \frac{\delta f}{\delta x}$ and $B = \frac{\delta f}{\delta u}$. Using the seventh order normalized equations and linearizing about the design equilibrium

$$\hat{x}_E = (1, 1, 1, 1, 1, 1, 1)$$

we have

A =

	$\hat{P}_2'$	$\hat{\dot{\omega}}_2$	$\hat{\dot{\omega}}_3$	$\hat{\dot{P}}_4$	$\hat{\rho}_B$	$\hat{P}_5$	$\hat{N}$
$\hat{P}_2'$	0	51.52	-51.52	0	0	0	0
$\hat{\dot{\omega}}_2$	-2188.3	0	0	0	0	0	0
$\hat{\dot{\omega}}_3$	40159	0	-31967	-8191.4	0	0	40159
$\hat{P}_4$	0	0	21.436	-106.72	52.893	0	15.343
$\hat{\rho}_B$	0	0	37.78	-38.448	0	0	0
$\hat{P}_5$	0	0	0	61.97	0	61.97	0
$\hat{N}$	0	2.2558	-3.5131	11.488	-1.2587	-8.9708	-2.5146

B =

	$\hat{\dot{\omega}}_f$	$\hat{\theta}$
$\hat{P}_2'$	0	0
$\hat{\dot{\omega}}_2$	0	0
$\hat{\dot{\omega}}_3$	0	0
$\hat{P}_4$	32.4219	0
$\hat{\rho}_B$	.66849	0
$\hat{P}_5$	0	-61.97
$\hat{N}$	0	-8.9708

The corresponding third order normalized linearizations are

$$A = \begin{array}{c} \\ \hat{P}_4 \\ \hat{\rho}_B \\ \hat{N} \end{array} \begin{array}{c} \begin{array}{ccc} \hat{P}_4 & \hat{\rho}_B & \hat{N} \end{array} \\ \begin{bmatrix} -112.27 & 52.924 & 42.26 \\ -48.12 & 0 & 47.444 \\ 2.8377 & -1.258 & -4.096 \end{bmatrix} \end{array}$$

$$B = \begin{array}{c} \\ \hat{P}_4 \\ \hat{\rho}_B \\ \hat{N} \end{array} \begin{array}{c} \hat{\dot{\omega}}_f \\ \begin{bmatrix} 32.4219 \\ .66849 \\ 0 \end{bmatrix} \end{array} .$$

These matrices were obtained for the values of $V = V_i$, $A = A_i$, and I given on page 18. Other gain systers were studied by changing V, A, and I. A summary of digital computer calculations of the eigenvalues and dominant time constants for various gain systems is presented below.

<u>Gain System A</u>. ($V = .1$, $A = .1$, $I = .00305$.) To obtain A and B matrices multiply bottom rows of all matrices by 10.

7th order eigenvalues: ($\tau \gtrsim .0425$)	-31852, -141.43, -60.278 -44.907 ± j329.24 -8.5923 ± j21.913
3rd order eigenvalues: ($\tau \gtrsim .0546$)	-89.288, -45.638, -18.304

Gain System B. (V = .1, A = .1, I = .0305) A and B matrices exactly as shown in equations.

7th order eigenvalues: ($\tau \gtrsim .296$)	-31892, -88.732, -61.218 -25.204, -3.3818 -33.738 ± j333.68
3rd order eigenvalues: ($\tau \gtrsim .352$)	-81.219, -32.309, -2.8386

Gain System C. (V = .01, A = .1, I = .0305) to obtain A and B matrices, multiply all rows except the bottom one by 10.

7th order eigenvalues: ($\tau \gtrsim .339$)	-318962, -814.82, -618.54 -310.44, -2.9529 -325.38 ± j3339.7
3rd order eigenvalues: ($\tau \gtrsim .343$)	-807.68, -316.2, -2.9166

Gain System D. (V = .01, A = .1, I = .0305) to obtain A and B matrices, multiply all upper rows by 10 and the bottom one by 1/10.

7th order eigenvalues: ($\tau \gtrsim 3.423$)	-318966, -806.7, -619.58 -316.34, -.29218 -324.17 ± j3340
3rd order eigenvalues: ($\tau \gtrsim 3.420$)	-807.25, -315.57, -.2924

Simulation

The actual simulation was conducted at the University of Notre Dame's Analog Computer Laboratory. An Electronic Associates TR-48, and a few small

TR-20 and TR-10 analog comptucers were used. Patching diagrams were obtained with the aid of ANSIR 3, a digital computer program which generates all pertinent analog simulation information, given the differential equations.

Control Problem

As stated earlier, the control problem is to schedule fuel in order to accelerate the engine operating state from Windmill to Design equilibrium conditions. However, maximum acceleration potential is limited by compressor surge or stall. As fuel is increased to accelerate the engine, pressure ratio, P_3/P_2', increases and airflow, $\hat{\omega}_2$, decreases. When stall is reached the pressure will drop and will cause a similar decrease in compressor efficiency. It is desirable to design controls which will cause the compressor to approach the stall line, however, since these controls tend to give a faster engine response.

Four different fuel control concepts were investigated on the simulation. They consist of an open loop control, linear feedback design, and two nonlinear controls developed by empirical methods. Their effects on engine performance and surge constraint were indicated by time response curves and trajectory plots.

Control 1. Here we use the term open loop to refer to this control. It simply involves letting

$$\hat{\omega}_f = 1$$

This corresponds to the value of $\hat{\omega}_f$ at design equilibrium.

Control 2. Control 2 fuel scheduling is a linear combination of compressor and turbine pressures given by

$$\hat{\dot{\omega}}_f = .99329 \quad \hat{P}_3 + .00608 \ \hat{P}_2'$$

Control 3. This control is the product of compressor airflow and rotor speed. It is defined as

$$\hat{\dot{\omega}}_f = \hat{\dot{\omega}}_3 \hat{N}$$

Control 4. This nonlinear control is given by

$$\hat{\dot{\omega}}_f = \hat{P}_4^2/\hat{\rho}_B$$

These controls were studied for both the seventh order and third order models given above and detailed results are given in reference (1) listed in the abstract.

The exercise was generally successful, yielding valuable experience in determining the essential dynamical characteristics of jet engines. The simulation results were reasonably representative of the types of response one would obtain using more sophisticated models.

IIb. INSTALLATION OF DYNGEN

This study is concerned with the setup and utilization of DYNGEN, a FORTRAN program for analyzing steady state and transient performance of jet engines. DYNGEN is based on three earlier computer programs; SMOTE, GENENG, and GENENG II. These programs allowed the user to analyze the steady state behavior of several different engine configurations. However, as the need to predict the transient performance of these engines became more important in preliminary design and control studies, DYNGEN was developed to provide this added capability. The result is a digital program which can perform both steady state and transient calculations on a wide variety of engine configurations without any significant modifications to the program itself.

Because of DYNGEN's size (slightly less than 32,000 words of computer storage) certain decisions had to be made concerning the storage, editing and running the program. It was expected that DYNGEN would be used quite often in subsequent control studies. This made the reading and compiling of its 5,000 plus cards too cumbersome and costly. Furthermore, the control subroutines FCNTRL, NOZCTR, and DISTRB would have to be edited, as new control schemes were devised. Again, rereading and recompiling the entire program for each new control would prove nonfeasible. To overcome these and other difficulties, the Notre Dame PURT (Program Utility Routines for Tape) system available on the IBM SYSTEM/370 MODEL 158 was utilized. A description of user options and the JCL (Job Control Language) required to initiate them can be found in more detail in reference (2) listed in the abstract of this report.

Once DYNGEN was made readily accessible, a manual providing the information required to run the program was written. Because it was assumed that only one engine configuration would be used in further studies, the information in the original DYNGEN report could be vastly reduced and simplified. As a result, the time and effort required of a beginner to learn the particulars of the program were kept to a minimum.

Tape Initialization

The initial decision as to how to store and access DYNGEN was rather crucial. Ease of handling and editing and minimizing compiling and execution times were important considerations. Cards proved too slow, cumbersome, and costly to be used effectively. Disk storage is fast and convenient, but the storage requirements of the program were rather excessive. Magnetic tape seemed to be the best alternative. Tapes are reasonably fast, cheap to use and, with the available PURT system, quite easily accessed and edited.

The DYNGEN Program was received from NASA on a seven-track tape with a 556 bpi density. This had to be transferred onto a more commonly used nine-track, 1600 bpi tape. After securing a 600 foot tape from the Computing Center (tape number M0151, code name RAT) and submitting the NASA seven-track tape to the operator, the following JCL program transfers the information from the seven-track tape to the new nine-track tape.

```
//TAPE JOB (CF,E009,,8),STUDENTID,TIME=09,MSGLEVEL=(1,1)
/*ROUTE PRINT LOCAL
/*JOBPARM LINES=50
/*SETUP             M0151,RAT
/*SETUP             DYNGEN,DISPTH
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=DD,UNIT=TAPE7,
// DCB=(DEN=1,TRTCH=ET,RECFM=FB,LRECL=80,BLKSIZE=800),VOL=SER=DYNGEN,
// LABEL=(1,BLP),DISP=(OLD,PASS)
//SYSUT2 DD DSN=DYNGEN,DISP=(NEW,KEEP),
// LABEL=(1,SL),VOL=SER=M0151,UNIT=TAPE,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSIN DD *
 GENERATE MAXFLDS=1
 RECORD FIELD=(80,1,HE)
/*
//
```

Once the new nine-track tape was prepared, the entire FORTRAN program was punched out on cards. This was to make correcting easier and to provide a backup copy in the event the tapes were accidently misplaced or destroyed. The following JCL program performs the punching operation.

```
//PUNCH JOB (CF,E069),STUDENTID,TIME=01,MSGLEVEL=(1,1)
/*JOBPARM CARDS=5000
/*ROUTE  PUNCH LOCAL
/*SETUP          M0151,RAT
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=DYNGEN,DISP=(OLD,PASS),LABEL=(1,SL),VOL=SER=M0151,
// UNIT=TAPE
//SYSUT2 DD SYSOUT=B
//SYSIN DD *
 PUNCH TYPORG=PS
/*
//
```

The resulting source deck was divided into four separate source modules. The first three modules consist of the three control subroutines: FCNTRL, NOZCTR and DISTRB. The remaining source module contains the rest of the subroutines, dummy mainline, and control functions. Each of these source modules is read in turn into the compiler and translated into four sets of machine instructions called object modules. In order for these object modules to be executed, they must be processed by the loader. The loader accepts all the object modules from core, resolves the external references, searches the subroutine library, and combines everything into an executable form. Control then passes to the program for execution.

The advantage of segmenting DYNGEN in this manner is straightforward. If any of the source modules has to be corrected or replaced, only that module has to be recompiled. The new object code from the edited source module can then be combined with the remaining old object modules by the loader and executed. Because the bulk of DYNGEN is put into one module which should never need revision, the savings in time and computer costs is quite significant. Another added feature of this system is that new source and object modules can be loaded and executed without destroying the old modules. The programmer then has the option of which modules to save.

The following is the JCL program needed to segment the entire DYNGEN program into the four source modules, compile each, and store both the source and object modules back on the tape. It should be noted that this program should only be used to segment the original DYNGEN.

```
//BREAKUP JOB (CF,F081),STUDENTID,TIME=09,MSGLEVEL=(1,1),REGION=192K
/*JOBPARM LINES=50
/*SETUP          M0151,RAT
//STEP1 EXEC PORTINIT
//STEP2 EXEC PORTIN,TAPE=M0151
//STEP3 EXEC PORTSOUR
//SOURCE.SYSIN DD *
./ ADD NAME=DYNGEN1
./ NUMBER NEW1=100,INCR=100

        *********************************
        *                               *
        *  ENTER ALL CARDS EXCEPT THE   *
        *   THREE CONTROL SUBROUTINES.  *
        *                               *
        *********************************

//STEP4 EXEC PORTFORM,NAME=DYNGEN1
//STEP5 EXEC PORTSOUR
//SOURCE.SYSIN DD *
./ ADD NAME=DISTRB
./ NUMBER NEW1=100,INCR=100

        *********************************
        *                               *
        *   ENTER THE DISTRB SUBROUTINE. *
        *                               *
        *********************************

//STEP6 EXEC PORTFORM,NAME=DISTRB
//STEP7 EXEC PORTSOUR
./ ADD NAME=FCNTRL
./ NUMBER NEW1=100,INCR=100

        *********************************
        *                               *
        *   ENTER THE FCNTRL SUBROUTINE. *
        *                               *
        *********************************

//STEP8 EXEC PORTFORM,NAME=FCNTRL
//STEP9 EXEC PORTSOUR
./ ADD NAME=NOZCTR
./ NUMBER NEW1=100,INCR=100

        *********************************
        *                               *
        *   ENTER THE NOZCTR SUBROUTINE. *
        *                               *
        *********************************

//STEP10 EXEC PORTFORM,NAME=NOZCTR
//STEP11 EXEC PORTOUT,TAPE=M0151
//
```


The tape is now in its final form. It can be executed, edited or both with just a few simple JCL cards. Each option the user desires requires one or more JCL cards in "block" form. Furthermore, each desired option may require several preliminary JCL "blocks" for initialization - a tape cannot be edited until it has first been read into the main core memory. It is important, then, for the user to understand how each "block" works as well as be able to reproduce on cards the JCL programs that follow.

Program Initialization

Before anything can be done with the tape, it must first be put on the tape drives and read into core. This is accomplished by the following JCL cards.

```
//DYNGEN JOB (CF,F081),STUDENTID,TIME=02,MSGLEVEL=(0,0),REGION=320K
/*JOBPARM LINES=20
/*SETUP        M0151,RAT
//STEP1 EXEC  PURTINIT
//STEP2 EXEC  PURTIN,TAPE=M0151
```

The emphasis here will be on how to execute DYNGEN and how to edit the control subroutines. If the main body of subroutines, located in the module named DYNGEN1 is to be revised, TIME on the JOB card should be set to at least three minutes for editing purposes alone.

Editing

Once the source and object modules are loaded into core, they can be easily edited. When DYNGEN was divided into four parts, each module had its statements numbered in increments at 100 starting with 100. By specifying any individual statement number, any statement can be changed or deleted without affecting the rest of the source module. Also, the entire source module can be changed by submitting a new deck of source statements. Regardless of the editing method chosen, each revised module must be recompiled and, if it is to be saved, read back onto the tape. To get a text of all statements and their assigned sequence numbers, refer to the latest compiled listing or have the necessary modules printed out.

There are two print options available to obtain a listing. First, a listing of everything on the tape can be obtained by the card:

```
//STEP4 EXEC PURTLIST
```

If only individual modules listings are desired, the following JCL cards are required.

```
//STEP4 EXEC PURTLIST
//PURTLIST.SYSIN DD *
     ***********************************************
     *                                             *
     *   ENTER THE NAME OF THE LOAD MODULES        *
     *   TO BE LISTED ON CONSECUTIVE CARDS         *
     *   STARTING IN THE FIRST COLUMN              *
     *                                             *
     ***********************************************
/*
```

Once the sequence numbers of the statements to be edited are known, it is a simple matter to make revisions. First, the corrected FORTRAN statements should be punched on cards with the appropriate sequence numbers in columns 73 to 80. If a statement is to be inserted between two other statements, the sequence number on the new card must be some number between the sequence numbers of the other two statements. The following JCL cards are required to make statement changes.

```
//STEP5 EXEC PURTSOUR
//SOURCE.SYSIN DD *
./ CHANGE NAME=(ENTER MODULE NAME)
     *********************************
     *                               *
     *   ENTER ALL NEW STATEMENTS    *
     *   WITH THE SEQUENCE NUMBER IN *
     *   COLUMNS 73-80.              *
     *                               *
     *********************************
/*
```

The following JCL cards are needed to delete a statement or series of statements from a module.

```
//STEP7 EXEC PURTSOUR
//SOURCE.SYSIN DD *
./ DELETE SEQ1=N,SEQ2=M
/*
```

This will delete statements n through m. If only one statement is to be deleted, let n - m.

Sometimes, it is more convenient to resubmit the entire revised module rather than edit each individual statement. To change an entire module, a new source deck is punched out on cards and is entered in the following manner:

```
//STEP6 EXEC PURTSOUR
//SOURCE.SYSIN DD *
./ REPL NAME=(ENTER MODULE NAME)
./ NUMBER NEW1=100,INCR=100
     **********************************
     *                                *
     *   ENTER SOURCE MODULE DECK.    *
     *                                *
     **********************************
/*
```

To compile a newly edited source module, the following statement is required.

```
//STEP8 EXEC PURTFORM,NAME=(ENTER MODULE NAME)
```

If more than one module has been revised, the above card must be supplied for each. If the DYNGEN1 module is being edited, an extra card may be necessary. This is to insure that there is sufficient disk space available for the compilation.

```
//STEP8 EXEC PURTFORM,NAME=DYNGEN1
//FORT.SYSLIN DD SPACE=(CYL,(1,1))
```

To execute DYNGEN after all source and object modules are in their desired form requires the following cards:

```
//STEP9 EXEC PURTFGO,NAME=DYNGEN1
//GO.FT06F001 DD DISP=(NEW,PASS),SPACE=(TRK,(10,10)),UNIT=SYSDA
//GO.SYSIN DD *
     **********************************
     *                                *
     *   ENTER DYNGEN INPUT CARDS.    *
     *                                *
     **********************************
/*
```

Control then goes to the dummy mainline program in the DYNGEN1 module.

The editing techniques explained thus far change only the modules as they appear in core. To transfer the corrections back to the tape is the function of the PURTOUT card below.

```
//STEP 10 EXEC PURTOUT,TAPE=M0151
```

It is important to remember that if this card is not used, what was originally on the tape will remain, even if changes had been made to the source and object modules. Also, when more than one option is used during a particular run, the /* should be omitted between blocks. It should, however, be used after the last block.

Further discussion on the installation of DYNGEN, together with some simple control studies is contained in reference (2) listed in the abstract of this report.

IIc. OPTIMAL CONTROL STUDIES

The analog and digital simulation sutdies of the previous two sections are aimed at establishing models of jet engines for the purpose of experimentation with various control system designs. We are concerned with both linear multivariable and global nonlinear approaches as alternatives to linear quadratic Riccati methods. Here, we describe some preliminary studies by R. Basso and R. J. Leake on the global nonlinear approach. The work is incomplete as of the end of the grant period and will be completed in the summer of 1976. Some details are given in reference (6) listed in the abstract of this report, and will be presented at the 1976 Joint Automatic Control Conferences.

The linear multivariable approaches of the previous sections aim at various criteria common to good regulation about design equilibrium points. They depend on scheduled nominal inputs for large excursions which are often part of the engine design itself. A global nonlinear approach actually takes scheduling and local control action into account simultaneously by specifying the total control, and in this manner, overlaps the control and engine design problems.

As a preliminary global study, the problem is taken as one of going from idle to a high thrust design equilibrium point in minimum time, while observing side constraints on surge margin and high turbine inlet temperature.

The usual approaches of Optimal Control have been considered, but only within the more modern, more systematic, and more general framework of Mathematical Programming. If side constraints are included through penalty and barrier methods, the control problem can be viewed as an unconstrained minimization of an objective function $J(\underline{u})$ where

$$\underline{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}$$

and

$$J(\underline{u}) = K(x(N)) + \sum_{t=0}^{N-1} L(x(t),u(t),t)$$

with the differential equation approximation

$$x(t+1) = x(t) + f(x(t),u(t)).$$

Standard algorithms such as the Fletcher-Reeves Conjugate Gradient and the modified Davidson-Fletcher-Powell quasi-Newton algorithm with self-scaling and automatic restart can be applied directly to this problem with $J(\underline{u})$ being evaluated for $t = 0, 1, \ldots, N-1$. The gradient

$$\nabla_{\underline{u}} J = [\nabla_{u(0)} J \ \nabla_{u(1)} J \ \cdots \ \nabla_{u(N-1)} J]$$

is determined by solving the adjoint equations

$$\lambda(t) = \lambda(t+1) + \lambda(t+1) \ \nabla_x f(x(t), \ u(t) + \nabla_x L(x(t), \ u(t),t)$$

$$\lambda(N) = \nabla_x K(x(N))$$

for $t = N - 1, \ldots, 2, 1$ and observing that

$$\nabla_{u(t)} J = \lambda(t+1) \ \nabla_u f(x(t), \ u(t)) + \nabla_u L(N(t), \ u(t),t)$$

Each of the algorithms mentioned above requires a line search for minimization

$$h(a) = J(\underline{u}+a\underline{d}), \quad h'(0) = \nabla_{\underline{u}} J(\underline{u})$$

along directions $\underline{d}$, (usually not the gradient direction) determined by the particular algorithm. A cubic approximation of h(a) depending on h(0), h'(0) and two other values, h(a), h(b), as suggested by Lasdon, seems to be an efficient basis for carrying out a fairly simple line search in one step. Refinements of this approach are being sought, as it is essential that the number of evaluations of $J(\underline{u})$ and $\nabla_u J(\underline{u})$ be kept to a minimum due to the amount of computation required.

If the variables are normalized about a high thrust design point and certain simplifications are assumed for the nonlinear functions involved, a complete numerical model which has the essential characteristics of the engine is given in Section IIa of this report.

The single control variable is $W\hat{F}B$ and the problem is to drive the system from idle (actually windmill here with $W\hat{F}B = 0$)

$$\hat{\rho}_B = 1.774 \qquad \hat{P}_4 = .5384 \qquad \hat{N}_C = .5461$$

to design (with $W\hat{F}B = 1$) and

$$\hat{\rho}_B = 1 \qquad \hat{P}_4 = 1 \qquad \hat{N}_C = 1$$

in minimum time subject to surge margin and turbine inlet temperature constraints approximated respectively by

$$\hat{P}_4 \leq 1.25\ \hat{N}_C$$

$$\hat{P}_4 \leq 1.25\ \hat{\rho}_B.$$

Notice that a good second order approximation to the problem can be obtained by taking $\hat{P}_4$ as a control variable. Once the optimal $\hat{P}_4$ is obtained, it is used to determine the $W\hat{F}B$ necessary to yield $\hat{P}_4$. This technique was used to obtain a Dynamic Programming solution to the problem with time step size $\nabla t = .002$. Generally, there are three regions of control:

(1) $\hat{P}_4$ rides the surge margin constraint boundary ($\hat{P}_4 = 1.25\ \hat{N}_C$).

(2) $\hat{P}_4$ switches and rides the inlet temperature constraint boundary ($\hat{P}_4 = 1.25\ \hat{\rho}_B$).

(3) $\hat{P}_4$ leaves the inlet temperature constraint boundary and adjust to get all states to final design equilibrium value of unity.

The optimal time from idle to design is slightly less than one second.

A preliminary open loop solution, using a conjugate gradient algorithm with time variable penalty functions to account for state constraints and terminal condition, has been obtained but there are some constraint violations and the solutions are, as yet, unsatisfactory.

The constraint violations which occur in $\hat{P}_4$ are due to exterior penalty functions. Feasible [illegible]ction and barrier approaches are being studied to remedy this effect.

The fact that the problem is time optimal and the optimal number of steps N is initially unknown does not appear to offer any real difficulty as a few rough computational runs quickly give one a good estimate of N.

Future studies will involve experimentation with various conjugate gradient and quasi-Newton algorithms in combination with penalty, barrier, and augmented Lagrangian techniques. Also, it appears that the state constraints can be taken to be of very simple form (such as $x_i(t) \geq 0$) and this should open the way for the development of simple feasible direction methods to handle the hard side constraints.

III. ONGOING AND RELATED ACTIVITIES

In addition to the activities described in the preceding two sections, other grant activities, ongoing and related projects, are taking place.

1976 Joint Automatic Control Conference

With the assistance of J. Zeller and B. Lehtinen of the NASA Lewis Research Center, the investigators have arranged a special sponsored session at the Joint Automatic Control Conference, Purdue University, Lafayette, Indiana, July 28, 1976. A brief description of the session follows:

Application Session 2
Theme----------Application of Control
Title----------"Jet Engine Control"
Chairman-------M. K. Sain, University of Notre Dame
Vice-Chairman--B. Lehtinen, NASA Lewish Research Center

Paper No. 1
"The Role of Multivariable Control Techniques in the Design of Turbine Engine Control Systems"
C. Skira, Aero Propulsion Laboratory
Wright Patterson Air Force Base, Ohio

Paper No. 2
"Multivariable Control Design Principles with Application to the F100 Turbofan Engine"
R. L. DeHoff and W. E. Hall, Systems Control, Inc.
Palo Alto, California.

Paper No. 3
"Simulation of a Turbofan Engine for Evaluation of Multivariable Optimal Control Concepts"
K. Seldner, NASA Lewish Research Center
Cleveland, Ohio

Paper No. 4
"Application of Multivariable Optimal Control Techniques to a Variable Area Turbine Engine"
E. C. Beattie and W. R. Sprock, Pratt & Whitney Aircraft, Inc.
East Hartford, Connecticut

Paper No. 5
"Alternative Methods for the Design of Jet Engine Control Systems"
M. K. Sain, R. J. Leake, R. Basso, R. Gejji, A. Maloney, and
V. Seshadri, University of Notre Dame
Notre Dame, Indiana

This session is officially sponsored by the Control Systems Society of the Institute of Electrical and Electronics Engineers.

Dominance and Inverse Nyquist Studies

Mr. A. Maloney, a teaching assistant not receiving direct support under this grant, has been developing software to help in the evaluation of the methods of H. Rosenbrock as applied to jet engine models. Selected curves from this work have been used in Paper No. 5 of the JACC session described above. The preliminary work is expected to be complete during the coming summer.

Classroom Evaluation

Jet engine models were used extensively as project models in the course EE 555, Multivariable Control Systems, at the University of Notre Dame in Spring 1976. Several insights into various frequency domain techniques were obtained in this way, and one or two further projects may possibly grow out of this class work.

In the course EG 551, Mathematical Programming, various algorithms were investigated for computation of time optimal controls for the simple jet engine model discussed in Parts IIa and IIc of this report.