AN OPTIMIZATION STUDY ON A HYBRID COMPUTER

by

Robert S. Gonzalez

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College
THE UNIVERSITY OF ARIZONA

1 9 6 9

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED:

Robert S. Gonzalez

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Granino A. Korn
GRANINO A. KORN
Professor of Electrical Engineering

5/1/69
Date

ACKNOWLEDGMENTS

The author wishes to express his appreciation to Dr. Granino A. Korn for his continued interest and helpful guidance during this study, and to the National Aeronautics and Space Administration for the support of this research under NASA Grants NSG-646 and NGR-03-002-024.

TABLE OF CONTENTS

	Page
LIST OF ILLUSTRATIONS	v
ABSTRACT	vi
CHAPTER	
1. INTRODUCTION	1
2. THE MAXIMUM PRINCIPLE	4
2.1 Introduction	4
2.2 Statement of the Problem	4
2.3 The Maximum Principle	5
2.4 A Simplification	6
2.5 The Boundary-value Problem	8
3. RANDOM-INCREMENT OPTIMIZATION	9
3.1 Introduction	9
3.2 Solution of the Boundary-value Problem	9
3.3 The Hybrid Computer System	10
3.4 Adaptive Random-increment Search	12
4. EXAMPLES	20
4.1 Introduction	20
4.2 Second-order System	20
4.3 Third-order System	26
5. CONCLUSION	33
APPENDIX 1. BBOPT2--OPTIMIZATION MAIN PROGRAM . . .	36
APPENDIX 2. SUBROUTINE INPUT	49
APPENDIX 3. SUBROUTINE OUTPUT	54
APPENDIX 4. SUBROUTINE ARACK	58
LIST OF REFERENCES	60

LIST OF ILLUSTRATIONS

Figure	Page
1. Flow Chart for Optimization Algorithm	13
2. Simulation of Second-order System	22
3. Time Responses for Second-order System	24
4. Cross Sections of Criterion Surfaces for the Second-order System	25
5. Simulation of Third-order System	29
6. Time Response for Third-order System	30
7. Cross Sections of Criterion Surfaces for the Third-order System	32

ABSTRACT

In this paper, the maximum principle is applied to minimum-time optimal-control problems, and an optimization algorithm is presented which can be implemented on a hybrid computer. The state and adjoint equations are set up on ASTRAC II, a high-speed analog computer capable of 1000 differential equation solutions per second. The optimization algorithm is implemented on a PDP-9, an 18-bit, digital computer. The optimization scheme has global and local search phases and uses a vector optimization criterion. Second and third-order bang-bang control systems are studied as examples.

CHAPTER 1

INTRODUCTION

This study reviews the application of the maximum principle to minimum-time optimal control problems and explores the feasibility of its implementation on an analog-digital computer.

Minimum-time optimal control problems require transfer of a system from a given initial state to some desired final state in minimum time. In addition, for the systems treated here, the control variable is constrained within certain limits.

Pontryagin's Maximum Principle is used to find the optimal control. It is assumed that the system can be described by n first-order differential equations. When the maximum principle is applied to optimal control problems, a difficult two-point boundary value problem must be solved. In addition, n adjoint equations must be solved to generate the control. The fact that time is also a variable further complicates the problem.

The boundary value problem is essentially a search for the proper initial values of the adjoint variables and for the correct time interval (minimum time). These

unknown parameters are found by iteration using an analog-digital (hybrid) computer.

The state and adjoint equations are simulated on ASTRAC II, a high-speed analog computer capable of 1000 differential equation solutions per second. Thus, all differential equations in the systems are simultaneously solved in the most efficient manner possible.

The unknown parameters which generate the optimal control are found with an optimization algorithm implemented on a PDP-9 digital computer. The optimization scheme uses a random-search strategy with adaptive features and a vector criteria concept.

All parameter-optimization problems are subject to the danger of having a bad starting point in the parameter space. The optimization algorithm used here has global and local phases which reduce the effects of this characteristic. The global phase uses a modified pure-random search to find a starting point of likely success and the local phase continues the search with a creeping-random search until the optimum point is found or until the adaptive portion of the algorithm reinitiates the global phase. At all times, the analog computer is under direct control of the digital computer.

The effectiveness of the maximum principle and the hybrid computer optimization is demonstrated with second- and third-order examples. Results which include optimal

trajectories, convergence times, and cross sections of the criterion surfaces are presented.

The digital computer and the A/D-D/A linkage interfaced with the analog computer in the University of Arizona Analog/Hybrid Laboratory are relatively new. One of the principal purposes of this optimization study is to prove the feasibility of a complete optimization process on the new hybrid computer, and thus to pave the way for more sophisticated problems.

The appendices list the optimization programs written for the given examples which may serve as a prototype for other optimization studies.

CHAPTER 2

THE MAXIMUM PRINCIPLE

2.1 Introduction

This chapter will discuss the application of Pontryagin's Maximum Principle (Pontryagin et al., 1962) to the solution of minimum-time optimal control problems. The maximum principle is applicable to a much wider range of problems and is thoroughly discussed in the literature.

2.2 Statement of the Problem

Let the dynamics of a given system be described by the set of n first-order differential equations (state equations)

$$\begin{aligned}\frac{dx_i}{dt} &= f_i(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r) \\ &= f_i(x, u) \quad (i = 1, 2, \dots, n)\end{aligned}\quad (2.2.1)$$

where $x_1(t), x_2(t), \dots, x_n(t)$ are the state variables, and $u_1(t), u_2(t), \dots, u_r(t)$ are the control variables.

The class of systems to be studied requires that a control $u(t)$ be selected for $t, 0 \leq t \leq T$, such that:

1. The control variables, $u_k(t)$, are subject to the constraints.

$$|u_k| \leq L_k \quad (k = 1, 2, \dots, r). \quad (2.2.2)$$

2. The system state variables, $x_i(t)$ are transferred from the initial state, $x(0) = [x_1(0), x_2(0), \dots, x_n(0)]$, to a final state, $x(T) = [x_1(T), x_2(T), \dots, x_n(T)]$, in such a way that the time T is minimized.

2.3 The Maximum Principle

Necessary conditions for optimal control are defined in terms of Pontryagin's Maximum Principle (Korn and Korn, 1968). Define $n+1$ adjoint variables, $p_0(t)$, $p_1(t)$, $p_2(t)$, $\dots$, $p_n(t)$ as solutions of the $n+1$ first-order differential equations (adjoint equations)

$$\frac{dp_0}{dt} = 0, \quad \dot{p}_i = - \sum_{k=1}^n p_k \frac{\partial f_k}{\partial x_i} \quad (i = 1, 2, \dots, n) \quad (2.3.1)$$

with

$$p_0(t) = \text{constant} \leq 0. \quad (2.3.2)$$

Then the optimal control minimizing the time T is given by that set of admissible control variables $u_k = u_k(t)$ which maximize the Hamiltonian function

$$H = p_0 + \sum_{i=1}^n p_i f_i \quad (2.3.3)$$

for each t between 0 and T ; moreover,

$$M \equiv \max_{(u_1, u_2, \dots, u_r)} H = 0 \quad (0 \leq t \leq T) \quad (2.3.4)$$

The usual transversality conditions associated with the adjoint variables are avoided, since in the special problem under consideration here both the initial and final values of the state variables are specified.

The maximum principle is applicable to systems where the f_i are continuous functions of the u_k , piecewise continuous functions of time, and are once continuously differentiable with respect to the x_j .

2.4 A Simplification

If a system can be described by equations of the form

$$\begin{aligned} \frac{dx_i}{dt} &= a_i(x_1, x_2, \dots, x_n) + \sum_{j=1}^r b_{ij}(x_1, x_2, \dots, x_n) u_j \\ &= a_i(x) + \sum_{j=1}^r b_{ij}(x) u_j \quad (i = 1, 2, \dots, n) \end{aligned} \quad (2.4.1)$$

where both the initial and final states are specified, the controls u_j are subject to the constraints (2.2.2), and if the functions $a_i(x)$ satisfy

$$a_i [x_1(T), x_2(T), \dots, x_n(T)] = 0 \quad (i = 1, 2, \dots, n) \quad (2.4.2)$$

then the condition that $\max_{(u_1, u_2, \dots, u_r)} H = 0$ is redundant.

The proof of this theorem is straightforward (Maybach, 1967), and the added conditions (Eq. 2.4.1 and 2.4.2) still permit the solution of a large class of systems.

The results of this simplification are:

1. H is maximized with respect to the u_j if

$$u_j = L_j \operatorname{sgn} \left[\sum_{i=1}^n p_i b_{ij}(x) \right] \quad (j = 1, 2, \dots, r). \quad (2.4.3)$$

2. The condition that $p_0(t) = \text{constant} \leq 0$ is satisfied, thus reducing the adjoint equations to

$$\frac{dp_i}{dt} = - \sum_{k=1}^n p_k \frac{\partial f_k}{\partial x_i} \quad (i = 1, 2, \dots, n). \quad (2.4.4)$$

3. The Hamiltonian function is automatically maximized to zero, thus eliminating the necessity to generate it.

The problem stated in Section 2.2 can now be restated as a boundary-value problem.

2.5 The Boundary-value Problem

The minimum-time optimal control problem reduces to the solution of $2n$ equations, n state equations and n adjoint equations, subject to $2n$ boundary conditions, the n given initial values and the n final values for the state variables. The n initial conditions can be directly set, but one must satisfy the n final conditions by properly adjusting the initial values of the adjoint variables and the time, T . The adjoint equations are homogeneous in the p_i ; therefore, if $p_1(t), p_2(t), \dots, p_n(t)$ is a solution, so is $Ap_1(t), Ap_2(t), \dots, Ap_n(t)$ for any arbitrary constant A . Thus the magnitude of the initial value of one of the adjoint variables can be fixed. However, the signs of the p_i are important because of Eq. (2.4.3)..

The two-point boundary problem has, then, been reduced from one involving an $(n+1)$ -dimensional search [over $p_1(0), p_2(0), \dots, p_n(0), T$] to one involving only an n -dimensional search plus a binary decision.

CHAPTER 3

RANDOM-INCREMENT OPTIMIZATION

3.1 Introduction

The maximum principle yields a two-point boundary-value problem which is especially amenable to hybrid computation. The state and adjoint equations are solved on the analog computer while the optimization algorithm is implemented on the digital computer. The following sections explain the boundary-value problem in more detail, describe the hybrid computer system used, and give a complete discussion of an adaptive random search program.

3.2 Solution of the Boundary-value Problem

Every reasonable set of initial conditions for the adjoint variables and a time period T will yield a solution to the adjoint equations, $p_1(t)$, $p_2(t)$, ..., $p_n(t)$ which, by Eq. (2.4.3),

$$u_j = L_j \operatorname{sgn} \left[\sum_{i=1}^n p_i b_{ij}(x) \right] \\ (j = 1, 2, \dots, r)$$

generate a control $u(t)$. This control drives the system state variables from the given initial state $x(0)$ to some arbitrary terminal state $x(T)$. The boundary-value problem

requires that one choose a set of parameters such that the desired terminal state is achieved.

By the maximum principle, the generated control $u(t)$ which gives this desired result will be the minimum-time optimal control being sought.

The unknown parameter values which generate the optimal control can be found through an adaptive random search procedure (Stewart, Kavanaugh, and Brocker, 1967) which iteratively solves the state and adjoint equations with randomly perturbed parameters.

For the cases to be considered here, the desired terminal conditions on the state variables are $x_1(T) = 0$, $x_2(T) = 0$, ..., $x_n(T) = 0$. Therefore, if a boundary-point criterion F_i is defined as the absolute value of the terminal state $x_i(T)$ for each state variable, then the iteration will converge when each performance index F_i has been minimized to within some acceptable tolerance of zero.

3.3 The Hybrid Computer System

The hybrid computing system used for the optimization consists of the ASTRAC II high-speed iterative differential analyzer interfaced with a PDP-9 digital computer (O'Grady, 1969).

ASTRAC II is an all-solid state iterative differential analyzer employing forty ± 10 volt amplifiers and

capable of solving differential equations at iteration rates as high as 1000 solutions per second. The PDP-9 general purpose digital computer is a single-address, 18-bit word, parallel binary computer with 16K core memory. The complete cycle time is 1 microsecond and the add time is 2 microseconds.

Data transfer between computers is primarily the function of four 12-bit multiplying digital-to-analog converters (MDACs) and a 12-bit analog-to-digital converter (ADC) with four multiplexed analog inputs. The MDACs require 5 microseconds per conversion and the ADC requires 150 microseconds per conversion.

The MDACs and the ADC use the 12 most significant bits of the 18-bit accumulator in the PDP-9. Therefore, in the optimization program, variables related to voltages on the ASTRAC II are multiplied (for input) or divided (for output) by 13107.2 to maintain proper scaling.

The PDP-9 exercises control over ASTRAC II through control-register output levels which are transferred from the PDP-9 accumulator to the linkage patchbay under program control. The control-register output levels are used directly as RESET/HOLD and COMPUTE/TRACK signals for the integrators. ASTRAC II exercises control over the PDP-9 through flags connected to the program-interrupt facility. Amplifier overloads are detected by one of the flags which puts all the integrators in RESET/HOLD mode.

The hybrid system is also equipped with a hardwired noise generator which can output uniformly-distributed, uncorrelated, 18-bit words once every 30 microseconds.

Input and output of all parameters are processed through an electronic keyboard and television monitor (CRT). A teletype printer is also available for hard-copy output.

Solutions of the state and adjoint equations are displayed on a four-trace oscilloscope connected to the analog computer.

3.4 Adaptive Random-increment Search

The main program of the optimization algorithm is written in assembly language, and the subroutines for input and output of user information are written in FORTRAN IV. All variables which appear in both the main program and the subroutines are stored in a 100-element array located in the main program. Calls to the FORTRAN IV subroutines consist of simple three-line calling sequences.

The optimization consists of a "global phase," which seeks a point in the parameter space for which each F_i is less than some value called delta (the delta criterion) and of a "local phase," which performs a fine local search about this point until each F_i is less than some value called epsilon (the epsilon criterion), at which time the optimization is completed. Fig. 1 shows a

Fig. 1. Flow Chart for Optimization Algorithm

general flow chart for the algorithm. Twos-complement addition (TADing) is the only arithmetic operation needed with the exception of a division used for the generation of a compute-period starting value. A test to determine "if a is less than (greater than or equal to) b" is performed by TADing to "a" the twos complement of "b" and then checking the sign of the result. This operation is used extensively throughout the program and will be implied by either of the logical relations mentioned above.

Random numbers, which are generated with the aid of a macro, are uniformly distributed about zero. The width of the distribution is specified by indicating the number of bits of magnitude desired.

The maximum distribution width is generated by specifying 17 bits of magnitude (the 18th bit is the sign bit). Therefore, if K bits of magnitude are desired, the instruction sequence in the macro transfers an 18-bit word from the hardwired noise generator into the accumulator, shifts the accumulator 17-K bits to the right (while conserving the sign), and deposits the result in any desired location.

Salient features of the program have been described; the following will now describe the actual random search and will refer to the variables by the same names used in the program (Appendix A).

After all programs have been loaded, the optimization program may be started or restarted by the CONTROL-S command on the CRT keyboard. At this time, a RESET/TRACK level is sent to the ASTRAC II and the input subroutine is entered.

SUBROUTINE INPUT displays a list of variables and options along with an associated code number (0-15). Desired parameter values for the algorithm (algorithm parameters) may be entered in any order, or all the parameters may be set to standard values which have been found the most efficient. In addition, any one of the three modes of operation may be initialized.

The optimization program has three modes of operation: normal single optimization (mode 0); averaging mode (mode 1), where averages on certain parameters are computed over any desired number of optimizations; and display mode (mode 2), which simply displays the solutions of the state and adjoint equations for any desired parameters (solution parameters).

The optimization begins by setting the algorithm parameters for the global phase of the random search. Purely random starting values are then generated for the solution parameters A_j with the random-number macro described above. The time period A_1 is generated by dividing a random number between 0 and -131072 by a scaling factor which then results in a compute period uniformly

distributed between 0 and TMAX microseconds. The remaining solution parameters are the initial conditions (ICs) on the adjoint variables. The binary decision parameter A2 is generated by testing the sign of the hardwired noise generator. The continuous ICs (A3, etc.) result as purely random values between +10 and -10 volts on the analog computer. All ICs are sent to the ASTRAC II with the MDACs as they are generated.

The analog computer run is initialized by transferring interface control-register logic levels, which determine the COMPUTE/TRACK mode to the ASTRAC II. The length of the compute period is determined with a two instruction loop by incrementing A1 (a negative value) to zero, at which time a logic level is transferred, putting the ASTRAC II back into RESET/HOLD mode. This technique for timing results in 3 microseconds resolution for T. The terminal states $x_1(T)$, $x_2(T)$, ..., $x_n(T)$ are held with HOLD circuits for the analog-to-digital conversion. If an overload occurs during the analog computer run, a program interrupt immediately aborts the run.

The first terminal state $x_1(T)$ is transferred to the PDP-9 with an ADC; converted to absolute value; stored as F1; and compared with MF1 the best-previous performance index (initially set at some arbitrary value). If F1 is greater than or equal to MF1, the run is considered a "failure" and no further ADCs are performed. If F1 is less

than MF1, the next terminal state is converted and tested as above. The run is considered a "success" if all terminal states are less than their respective best-previous performance indexes. Note that this is a logical ANDing process, which is "stricter" than the usual typical performance index employing the sum of all the absolute values of the terminal states. Since the greater majority of runs are failures, many time consuming (150 micro-seconds) ADCs are avoided in this manner.

Assuming for the moment that the analog computer run resulted in a failure or an overload, the ASTRAC II is put into RESET/TRACK mode, and an accounting operation is performed which constitutes the adaptive feature of the optimization. If CNT consecutive failures have not occurred, then the best-performance solution parameters (AS1, etc.) are randomly perturbed and another analog computer run is initiated. If CNT consecutive failures have occurred, then the noise variances for each parameter perturbation may be expanded or contracted by masking the noise words. If the noise distributions have been modified RST consecutive times, then the optimization is reinitialized by resetting all parameters for the global phase of the search.

Perturbations which are added to the solution parameters that have given the best performance are generated with the random-number macro. If the

optimization is in the local phase, the binary-decision parameter value is fixed at the value it had when the delta criterion was satisfied. After the perturbation of all parameters and the MDACing of ICs, a new ASTRAC II run is initialized.

If the ASTRAC II computer run is a success, that is, if all performance indexes are improved, the failure counters are reset to zero, and the improved performance indexes are stored as the best-previous performance values and then tested against the delta criterion. If each performance index is less than some moderately small value (called MD1, MD2, etc.), then the delta criterion is satisfied. If the delta criterion is not satisfied, then the noise variances are reset to their original values; the solution parameters are stored; and new parameters are formed by incrementing the stored parameters with the same perturbations which gave the improved performance.

If the delta criterion is satisfied, then the local phase of the optimization is initiated and the delta criterion is no longer tested. Instead, after each successful analog run, the performance indexes are tested against the epsilon criterion which is similar to the delta criterion. If each performance index is less than some tolerance of zero (called MEP1, MEP2, etc.), then the optimization is completed. If the epsilon criterion is not satisfied, then the variance of the noise distributions is

reset to the original values for the local search. The subsequent operations are identical to those performed after the delta criterion test.

When the optimal set of solution parameters is found, all pertinent data are displayed on the CRT-display with the FORTRAN IV subroutine OUTPUT and the optimal control and system trajectories are displayed on a four-trace oscilloscope.

CHAPTER 4

EXAMPLES

4.1 Introduction

The following examples illustrate the maximum principle and the optimization strategy presented above. Optimal controls are found for second- and third-order linear systems. Note should be taken of the relative ease of the analog-computer programming involved and of the desirable absence of complicated patched-digital-logic programming. All decisions and control signals are handled by the digital computer.

4.2 Second-order System

Let the second-order system be described by the differential equation

$$\frac{d^2x}{dt^2} = u \quad , \quad |u| \leq 1 \quad (4.2.1)$$

It is required that

$$\frac{dx}{dt}(T) = x(T) = 0 \quad (4.2.2)$$

with a minimum-time performance index (Sec. 2.2).

Let $x_1 = x$ and $x_2 = \frac{dx}{dt}$. Then the theory of Sec. (2.3) results in

$$\frac{dx_1}{dt} = x_2 \text{ and } \frac{dx_2}{dt} = u \quad (\text{state equations}) \quad (4.2.3)$$

$$H = p_0 + p_1 x_2 + p_2 u \quad (\text{Hamiltonian}) \quad (4.2.4)$$

$$\frac{dp_0}{dt} = 0, \frac{dp_1}{dt} = 0, \frac{dp_2}{dt} = -p_1 \quad (\text{adjoint equations}) \quad (4.2.5)$$

$$u = \text{sign}(p_2) \quad (\text{control law}) \quad (4.2.6)$$

The simplification theorem of Sec. (2.4) applies to this system; this allows the binary decision

$$p_1 = \pm 1. \quad (4.2.7)$$

The boundary-cost criteria are defined as

$$F1 = \left| a_1 x_1(T) \right| \text{ and } F2 = \left| a_2 x_2(T) \right| \quad (4.2.8)$$

where a_1 and a_2 are the scale factors used in the analog simulation described below.

The analog-computer block diagram for the state and adjoint equations is shown in Fig. 2. The control law (Eq. 4.2.6) is implemented with a comparator followed by a precision diode-bridge limiter. The limiting level is accurately adjusted to 1 volt by potentiometer 29, which constitutes a variable load resistance for the diode bridge (Korn and Korn, 1964). As indicated, the scale factors for x_1 and x_2 are $a_1 = 0.5$ and $a_2 = 1.0$.

The control-logic signals R_1 and R_2 are controlled by the digital computer (PDP-9). Logic signal R_1 provides

All integrators have gain $2 \cdot 10^4$

Fig. 2. Simulation of Second-order System

the RESET/COMPUTE logic levels for all the integrators, while R_2 provides the TRACK/HOLD logic levels for the sample/hold units.

The digital optimization program for the second-order problem is given in Appendix A. Annotation of the digital program complements the general outline given in Sec. (3.3).

The following results were obtained with the algorithm parameters set to the "standard parameters" given in subroutine INPUT (Appendix B). These are values for the algorithm parameters which have given efficient operation.

Note that MEP1 and MEP2 are proportional to 0.05v (Sec. 3.3). This means that the optimization is completed when both $F1 \leq 50$ mV and $F2 \leq 50$ mV.

Figure 3 shows the optimal time responses for two different sets of initial conditions. The trajectories in Fig. 3(a) are for $1/2x_1(0) = 0$, $x_2(0) = 5$. The optimal solution parameters are $A1 = 600$ us, $A2 = -10v$, and $A3 = -8.4v$; these values correspond to $T = 12$ sec., $p_1(0) = -1$, and $p_2(0) = -8.4$.

Cross sections through the optimal point of the criterion surfaces are shown in Fig. 4. Note that Fig. 4(a) simply represents the absolute value of the optimal time response.

The averaging mode (Sec. 3.3) of the optimization permits convenient automatic iteration of the optimization.

(a)

(b)

Fig. 3. Time Responses for Second-order System

For both (a) and (b):

First trace: $(1/2)x_1$, second line is zero, 5 v/cm.

Second trace: x_2 , fourth line is zero, 5 v/cm.

Third trace: $5u$, fifth line is 5 volts, 10 v/cm.

Horizontal: 0.1 ms/cm (computer time);

2 seconds/cm (scaled time).

(a) versus T sec.

(b) versus $p_2(0)$ volts

Fig. 4. Cross Sections of Criterion Surfaces for the Second-order System

vertical scale: 1 volt/division

$$(1/2)x_1(0) = 0, x_2(0) = 5, \text{ and } 10p_1(0) = -10$$

After the desired number of repetitions have been completed, subroutine OUTPUT (Appendix C) displays pertinent information, including the average number of ASTRAC II analog runs and time required per optimization. The time T is also given in scaled time units (20,000:1). The switching time t_s is read from the oscilloscope to within two significant digits in scaled time. The optimization was repeated 100 times for 10 different sets of initial conditions. Table 1 compares the averages for t_s and T with the theoretical values (Maybach, 1967). Also shown are the average convergence times and average number of iterations.

The switching and terminal times could be found more accurately if MEPl,2 were made smaller. The short convergence times are attributed to the efficient sequencing of digital and analog processes, to automatic interruption of analog runs which cause overloads, and to the economical use of analog-to-digital conversions accounted for by the ANDing process used for testing the performance criterion.

4.3 Third-order System

The third-order problem is described by the differential equation

$$\frac{d^3x}{dt^3} = u, \quad |u| \leq 1 \quad (4.3.1)$$

Table 1. Results for Second-order System Averaged over 100 Optimizations Using Standard Parameters.

Initial Conditions		Theoretical		Measured		Convergence Time, Seconds	Number of Iterations
$1/2x_1(0)$	$x_2(0)$	t_s	T	t_s	T		
0	5	8.52	12.07	8.4	11.94	1.2	2162
0	4	6.82	9.65	6.8	9.54	1.5	2634
0	3	5.11	7.24	5.1	6.90	1.7	3092
0	2	3.41	4.83	3.4	4.68	1.5	3034
0	1	1.70	2.41	1.7	2.40	1.8	3582
5	0	3.17	6.33	3.2	6.18	1.0	1855
4	0	2.83	5.66	2.8	5.52	1.1	2028
3	0	2.45	4.90	2.4	4.74	1.0	1812
2	0	2.00	4.00	2.0	3.90	0.8	1538
1	0	1.41	2.82	1.4	2.64	1.0	2003

It is desired that

$$\frac{d^2x}{dt^2}(T) = \frac{dx}{dt}(T) = x(T) = 0 \quad (4.3.2)$$

with a minimum-time performance index.

$$\text{Let } x_1 = x, \quad x_2 = \frac{dx}{dt}, \quad \text{and } x_3 = \frac{d^2x}{dt^2}.$$

then

$$\frac{dx_1}{dt} = x_2, \quad \frac{dx_2}{dt} = x_3, \quad \text{and } \frac{dx_3}{dt} = u \quad (\text{state equations}) \quad (4.3.3)$$

$$H = p_0 + p_1 x_2 + p_2 x_3 + p_3 u \quad (\text{Hamiltonian}) \quad (4.3.4)$$

$$\frac{dp_0}{dt} = 0, \quad \frac{dp_1}{dt} = 0, \quad \frac{dp_2}{dt} = -p_1, \quad \text{and } \frac{dp_3}{dt} = -p_2 \quad (\text{adjoint equations}) \quad (4.3.5)$$

$$u = \text{sign}(p_3) \quad (\text{control law}) \quad (4.3.6)$$

Once again the simplification theorem applies, and Eq. (4.2.7) is true.

The simulation for the third-order system is shown in Fig. 5.

A separate optimization program is used, which is similar to that used for the second-order problem with a few additions. The standard parameters are easily expanded for the new problem.

Figure 6 shows the optimal time responses for two different sets of initial conditions. The optimal

All integrators have gain of $2 \cdot 10^4$

Fig. 5. Simulation of Third-order System

(a)

(b)

Fig. 6. Time Response for Third-order System

For both (a) and (b):

First trace: $(1/25)x_1$, second line is zero, 2 v/cm.

Second trace: $(1/5)x_2$, third line is zero, 2 v/cm.

Third trace: x_3 , fifth line is zero, 2 v/cm.

Fourth trace: $5u$, sixth line is 5 volts, 10 v/cm.

Horizontal: 0.1 ms/cm (computer time);

2 seconds/cm (scaled time).

solutions of Fig. 6(a) are for $1/25 x_1(0) = 2$, and $1/5 x_2(0) = x_3(0) = 0$.

Figure 7 shows the cross sections through the optimal point of the criterion surfaces. The criterion functions are defined as

$$F1 = \left| (1/25)x_1(T) \right|, F2 = \left| (1/5)x_2(T) \right|, F3 = \left| x_3(T) \right|. \quad (4.3.7)$$

Cross sections with respect to T are the absolute value of the optimal trajectories (Fig. 6a) and therefore are not repeated.

Average convergence time for the above initial conditions is 7.8 seconds with 9922 iterations per optimization.

(a) versus $(3/2)p_2(0)$ volts

(b) versus $(3/8)p_3(0)$ volts

Fig. 7. Cross Sections of Criterion Surfaces for the Third-order System

vertical scale = 1 volt/division

$(1/25)x_1(0) = 2$, $(1/5)x_2(0) = 0$, $x_3(0) = 0$, and $10p_1(0) = -10$

CHAPTER 5

CONCLUSION

Results from the second- and third-order examples show that the maximum principle can be effectively implemented on the hybrid computer to solve optimal control problems.

The optimization strategy programmed for the digital computer proved to be very effective for parameter searches. This is especially true when one considers its comparative simplicity, no complicated multiplication and division operations are necessary. The random-perturbation scheme requires no exploration steps and is not adversely affected by parameter-surface ridges. The global phase of the algorithm relieved the dilemma of choosing a starting point. The feature of greatest aid to this dilemma was the use of the vector criterion combined with the ANDing process for determining when an improvement had been made. This feature essentially kept the search from moving in directions away from the optimum point, a phenomenon which can occur with scalar criteria.

The local optimization phase worked well and was particularly enhanced by fixing the binary-decision parameter at the value of the prior success; this

effectively reduced the volume of the parameter space by a factor of two. The above points contributed to the reduction of the number of iterations.

The use of uniformly-distributed noise directly from the noise-generator obviates time consuming processing to generate normally-distributed noise. The hardwired noise-generator remarkably decreased the convergence times relative to using a software noise-generator.

The interruption of analog computer runs which overloaded any amplifier saved much time since direct control by the PDP-9 allows the initiation of a new run immediately following the overload.

The effect of the long conversion time needed by the analog-to-digital converter was decreased by the ANDing process used to compare the result of each ADC; if an ADC did not indicate improvement the run was immediately termed a failure.

The 12-bit digital-to-analog converters provide 5 mV of resolution which is very adequate. The high resolution kept the final convergence steps free of jitter.

The accuracy of the switching times given for the second-order example is typically 2.5%. The principle limitation on accuracy is the 3 μ s resolution for the time period given by the timing loop in the optimization program. The relative resolution could be increased by a

factor of 10 by simply selecting a slower time scale on the ASTRAC II.

The absence of digital-logic patching simplifies preparations for a new problem.

The optimization study proves the feasibility for implementation of the maximum principle on the hybrid computer and indeed for the solution of more sophisticated problems.

APPENDIX 1

BBOPT2--OPTIMIZATION MAIN PROGRAM

```

/      BBOPT2
/      SECOND-ORDER BANG-BANG PROBLEM
/
/ THE PROGRAM FINDS A SET OF PARAMETERS,
/ A1,A2,A3, WHICH DETERMINE A CONTROL
/ THAT TAKES THE SECOND-ORDER SYSTEM FROM
/ SOME INITIAL STATE X(0) TO THE ZERO
/ STATE IN MINIMUM TIME.
/

```

```

      .GLOBL INPUT,OUTPUT,ARACK,.AE
      .IODEV 2,3,4,5
/

```

```

/ DEFINE MACRO FOR RANDOM WORD GENERATION
/ A UNIFORMLY DISTRIBUTED 18-BIT RANDOM
/ WORD IS SAMPLED FROM THE HARDWIRED
/ NOISE GENERATOR, SHIFTED (17-M) TIMES TO
/ THE RIGHT (WHILE CONSERVING THE SIGN),
/ AND DEPOSITED IN LOCATION AK.
/

```

```

      .DEFIN RAND,M,AK
      LAC M
      SNA
      JMP .+12
      CMA
      TAD (LRSS+22) /17 BITS OF
      DAC .+2      /MAGNITUDE
      706412      /GET RANDOM WORD
      XX
      SNA          /ELIMINATE ZERO
      JMP .-3
      SPA          /ELIMINATE 400000
      TAD (1       /RECREATE ZERO
      DAC AK
      .ENDM
/

```

```

/ DEFINE MACRO FOR A-D CONVERSION & TEST
/ ANALOG-TO-DIGITAL CONVERSION OF ADC-
/ CHANNEL "INDEX" TRANSFERS THE VALUE OF
/ XK(T) TO THE PDP-9, THE ABSOLUTE VALUE
/ IS TAKEN AND DEPOSITED IN "FK"; IF "FK"
/ IS NOT LESS THAN "MFK" IN MAGNITUDE THE
/ RUN IS A "FAILURE". THE COMPARISON IS
/ ACCOMPLISHED BY TAD-ING "FK" WITH "MFK"
/ (A NEGATIVE NUMBER) AND CHECKING THE
/ SIGN OF THE RESULT.
/

```

```

      .DEFIN ADCON,INDEX,FK,MFK
      LAC INDEX      /SELECT CHANNEL

```

```

AMPXL
AADC          /START CONVERSION
AADS F        /CHECK ADC FLAG
JMP .-1
AADR          /TRANSFER C(ADC)
GSM           /TAKE ABSOLUTE
SZLICLL       /VALUE
TAD (I
DAC FK
TAD MFK       /COMPARE TO BEST
SMA           /PREVIOUS VALUE
JMP FAIL
.ENDM

```

```

/
/DEFINE MACRO FOR CRITERIA CHECK
/ THIS MACRO IS USED FOR BOTH THE DELTA
/CRITERIA & THE EPSILON CRITERIA. IF "FK"
/IS LESS THAN "MCRIT" IN MAGNITUDE, "MFK"
/IS SET EQUAL TO "MCRIT". IF NOT, THEN THE
/RUN HAS FAILED THE CRITERIA TEST AND A
/ZERO IS DEPOSITED IN "DECK".
/

```

```

.DEFIN FCRT,FK,MCRIT,DECK,MFK
LAC FK        /COMPARE FK
TAD MCRIT     /WITH CRITERION
SPA
JMP .+3       /IF PASS CRITERION
DZM DECK      /IF FAIL CRITERION
JMP .+3
LAC MCRIT     /FIX BEST-PAST FK
DAC MFK       /AT MCRIT
.ENDM

```

```

/
/INSERT "JMP FAIL" AT END OF SKIP CHAIN
/ THE FOLLOWING OPERATION IS PERFORMED
/ONLY AS A PRECAUTION. THE REFERRED TO
/SYSTEM OCCASSIONALLY EMITS SPURIOUS
/PROGRAM INTERRUPTS. THE FOLLOWING
/MODIFIES THE PROGRAM-INTERRUPT SKIP
/CHAIN TO ALLOW THE OPTIMIZATION TO
/CONTINUE WITHOUT DELAY. THE INTERRUPTED
/RUN IS TREATED AS A FAILURE.
/

```

```

SEARCH LAC* (2      /STORE START OF
DAC POINT#         /SKIP CHAIN
LOOP   ISZ POINT
LAC* POINT
SAD (IORS          /FIND END OF

```

```

        JMP FOUND      /SKIP CHAIN
        JMP LOOP
FOUND   LAC (JMP* 23    /INSERT JUMP
        DAC* POINT
        LAC (FAIL
        DAC* (23
        LAC (JMP START
        DAC SEARCH
/
/.SETUP FOR OVERLOAD SERVICE ROUTINE
/ WHENEVER AN ASTRAC II COMPUTER RUN
/OVERLOADS, A PROGRAM-INTERRUPT IS IMME-
/DIATELY ISSUED WHICH PUTS THE ASTRAC II
/INTO RESET/TRACK MODE. THE RUN IS THEN
/TREATED AS A FAILURE.
/
        .SETUP 0, ASFI, FAIL
        DBK
/
/ALL PARAMETERS ARE ENTERED INTO THE
/MACRO MAIN PROGRAM THRU THE FORTRAN IV
/SUBROUTINE "INPUT". THE MODE OF OPERA-
/TION IS ALSO SELECTED IN THE SUBROUTINE
/
START   LAC (400        /DEPOSIT INITIAL
        DAC AD3         /PERTURBATIONS
        DZM IOMODE      /NORMAL MODE
INTAKE  ACRL+10         /RESET ASTRAC II
        ION
        JMS* INPUT      /INPUT PARAMETERS
        JMP .+2
        .DSA VARBL
        JMS* ARACK       /PUT AC#4 UP FOR
        JMP .+2         /PARAMETER CHECK
        .DSA VARBL
        .INIT 2,0,PRINT /ENABLES THE
        IOF             /CONTROL-P
        LAC (FCHEK      /COMMAND
        DAC TERM1#
        LAC IOMODE
        SAD (2
        JMP PRINT       /DISPLAY MODE
/
/START OF THE OPTIMIZATION
/
/THE GLOBAL PHASE OF THE OPTIMIZATION
/SEEKS A SET OF PARAMETERS WHICH WILL
/SATISFY THE DELTA CRITERIA. THE FOLLOW-

```

/ING SECTION RESETS CERTAIN ALGORITHM
/PARAMETERS FOR THIS PHASE OF THE SEARCH

```

RESET  LAC MIF1      /INITIAL
        DAC MF1      /BEST-PREVIOUS
        LAC MIF2     /PERFORMANCE
        DAC MF2      /VALUES
        LAC WBITS1   /INITIAL
        DAC BITS1    /WIDTHS OF NOISE
        LAC WBITS3   /DISTRIBUTIONS
        DAC BITS3
        LAC WINK1    /INITIAL
        DAC DBIT1    /MODIFICATIONS
        LAC WINK3    /FOR NOISE
        DAC DBIT3    /DISTRIBUTIONS
        LAC (NEWAD2  /INITIAL
        DAC TERM2#   /TRANSFER
        LAC (DELCK   /POINTS
        DAC TERM3#
        LAC CNT1     /INITIAL
        DAC MFRUNS#  /COUNTER
        LAC RST1     /SETTINGS
        DAC MIRUNS#
        DZM FRUNS#
        DZM IRUNS#
        DZM RUNS3

```

/THE STARTING VALUES FOR THE SOLUTION
/PARAMETERS A1, A2, & A3 ARE CHOSEN
/PURELY AT RANDOM. THE MACRO IS
/DESCRIBED ABOVE.

```

RAND1  RAND (21),A1  /RANDOM COMPUTE
        SPA          /PERIOD
        JMP .+3
        CMA
        TAD (1
        JMS* .AE
        LAC TSF
        DAC A1
        DAC AS1
        SNA
        JMP RAND1
RAND2  706412        /RANDOM BINARY IC
        SPA
        JMP .+3
        LAC PBIN
        SKP

```

BBOPT2

PAGE 5

```

    LAC MEIN
    DAC A2
    DAC AS2
    ADAØLIADAØX /MDAC
RAND3 RAND (21),A3 /RANDOM IC
    DAC AS3
    ADAILIADAIX /MDAC
    DZM ADI
/
/ THE ASTRAC II COMPUTER IS CONTROLLED
/ BY THE PDP-9. THE FOLLOWING SECTION
/ CLEARS OVERLOAD INDICATORS; SENDS THE
/ COMPUTE/TRACK LOGIC LEVELS; COUNTS FOR
/ THE COMPUTE PERIOD; AND SENDS THE
/ RESET/HOLD LOGIC LEVELS.
/
COMPUT ISZ RUNS1 /TALLY ASTRAC II
    NOP /COMPUTER RUNS
DISRST ASFI /CHECK OVERLOAD
    SKP /FLAG
    JMP DISRST /IF FLAG NOT CLEAR
    ION
    LAC A1
    DAC TIME
    LAC (17 /START ASTRAC II
    ACRL
    ISZ TIME /INCREMENT FOR
    JMP .-1 /COMPUTE PERIOD
    LAC (60 /HOLD TERMINAL
    ACRL /STATE VALUES &
    IOF /RESET INTEGRATORS
    JMP* TERM1 /JMP: FCHEK, DISPLA
/
/ THE FOLLOWING SECTION PERFORMS THE ADC
/ AND COMPARES THE VALUE OF THE PRESENT
/ PERFORMANCE AGAINST THE BEST-PREVIOUS
/ PERFORMANCE. THE MACRO IS DESCRIBED ABOVE
/
FCHEK ADCON (0),F1,MF1
    ADCON (1),F2,MF2
    ACRL+10 /RESET ASTRAC II
/
/ IF ALL PRESENT PERFORMANCE VALUES SHOW
/ IMPROVEMENT THEY ARE STORED AS THE
/ BEST-PREVIOUS PERFORMANCE VALUES.
/
SUCCES ISZ RUNS3 /TALLY SUCCESSES
    NOP

```

```

DZM FRUNS      /RESET FAILURE
DZM IRUNS      /COUNTERS
LAC F1         /STORE F1
CMA
TAD (1
DAC MF1
LAC F2         /STORE F2
CMA
TAD (1
DAC MF2
JMP* TERM3     /JMP:DELCK,EPCK

```

```

/
/ THE FOLLOWING "DELTA CRITERIA" COMPARES
/ "FK" WITH "MDK". IF THE CRITERIA IS
/ SATISFIED THE LOCAL PHASE OF THE OPTI-
/ MIZATION IS INITIATED. THE MACRO IS
/ DESCRIBED ABOVE.
/

```

```

DELCK  LAC (1
        DAC DCK#
        FCRIT F1,MD1,DCK,MF1
        FCRIT F2,MD2,DCK,MF2
        LAC DCK
        SZA
        JMP SEPCK      /IF CRITERIA
        LAC WBITS1     /SATISFIED
        DAC BITS1
        LAC WBITS3
        DAC BITS3
        JMP SSTEP

```

```

/
/ IF THE "DELTA CRITERIA" IS SATISFIED
/ THE FOLLOWING SETS CERTAIN ALGORITHM
/ PARAMETERS FOR THE LOCAL SEARCH.
/

```

```

SEPCK  LAC CNT2
        DAC MFRUNS
        LAC RST2
        DAC MIRUNS
        LAC (NEWAD3    /FIX BINARY
        DAC TERM2     /PARAMETER
        LAC (EPCK      /DELTA CRITERIA
        DAC TERM3     /IS NOT CHECKED
        LAC LINK1     /IN LOCAL PHASE
        DAC DBIT1
        LAC LINK3
        DAC DBIT3

```


/THE FOLLOWING CHECKS THE "EPSILON
/CRITERIA".IF SATISFIED THE OPTIMIZATION
/IS COMPLETED.

/

```

EPCK  LAC C1
      DAC ECK#
      FCRT F1,MEP1,ECK,MF1
      FCRT F2,MEP2,ECK,MF2
      LAC ECK
      SZA
      JMP PRINT      /IF CRITERIA
      LAC LBITS1      /SATISFIED
      DAC BITS1
      LAC LBITS3
      DAC BITS3

```

/

/THE PRESENT SOLUTION PARAMETERS (WHICH
/GAVE A PERFORMANCE IMPROVEMENT) ARE
/STORED.THE SEARCH NOW PIVOTS ABOUT
/THESE PARAMETER VALUES.THE PARAMETERS
/ARE PERTURBED IN THE SAME DIRECTION
/THAT GAVE THE IMPROVEMENT.

/

```

SSTEP  LAC A1
      DAC AS1
      TAD AD1
      DAC A1
      SPA
      JMP .+4
      DZM AD1
      LAC AS1
      DAC A1
      LAC A2
      DAC AS2
      LAC A3
      DAC AS3
      TAD AD3
      DAC A3
      ADA1LIADA1X    /MDAC
      JMP COMPUT

```

/

/IF THE ASTRAC II COMPUTER RUN RESULTS
/IN A FAILURE THE FOLLOWING ACCOUNTING
/IS PERFORMED.IF "CNT" CONSECUTIVE
/FAILURES HAVE NOT OCCURRED THE PARA-
/METERS ARE PERTURBED.IF "CNT" CONSECU-
/TIVE FAILURES HAVE OCCURRED THE WIDTHS
/OF THE NOISE DISTRIBUTIONS ARE MODI-

/FIED. IF THE NOISE DISTRIBUTIONS HAVE
 /BEEN MODIFIED "RST" CONSECUTIVE TIMES.
 /THE OPTIMIZATION IS REINITIALIZED.
 /NOTE: MFRUNS=CNT1 & MIRUNS=RST1 FOR THE
 /GLOBAL PHASE OF THE OPTIMIZATION AND
 /MFRUNS=CNT2 & MIRUNS=RST2 FOR THE LOCAL
 /PHASE.

/

FAIL	ACRL+10	/RESET ASTRAC II
	ISZ FRUNS	/CHECK NUMBER OF
	LAC FRUNS	/CONSECUTIVE
	TAD MFRUNS	/FAILURES
	SPA	
	JMP NEWAD1	
	ISZ IRUNS	/CHECK NUMBER OF
	LAC IRUNS	/NOISE-DISTRIBUTION
	TAD MIRUNS	/MODIFICATIONS
	SMAISZA	
	JMP RESET	/RESTART GLOBAL
	LAC BITS1	/SEARCH
	TAD DBIT1	
	DAC BITS1	
	LAC BITS3	
	TAD DBIT3	
	DAC BITS3	
	DZM FRUNS	

/

/THE FOLLOWING GENERATES THE PERTURBA-
 /TIONS WHICH ARE ADDED TO THE SOLUTION
 /PARAMETERS THAT HAVE GIVEN THE BEST
 /PERFORMANCE. NOTE: THE BINARY-DECISION
 /PARAMETER IS NOT VARIED DURING THE
 /LOCAL PHASE. THE MACRO IS DESCRIBED
 /ABOVE.

/

NEWAD1	RAND BITS1, AD1	/RANDOM TIME
	TAD ASI	/PERTURBATION
	DAC A1	
	SMA	
	JMP NEWAD1	
	JMP* TERM2	/JMP: NEWAD2, NEWAD3
NEWAD2	706412	/RANDOM
	SPA	/BINARY IC
	JMP .+3	
	LAC PBIN	
	SKP	
	LAC MBIN	
	DAC A2	

```

      ADAØL!ADAØX      /MDAC
NEWAD3 RAND BITS3,AD3      /RANDOM IC
      TAD AS3          /PERTURBATION
      DAC A3
      ADAIL!ADAIX      /MDAC
      JMP COMPUT

```

```

/
/ THE FOLLOWING DISPLAYS THE OPTIMAL
/ TRAJECTORIES AND CONTROL OR MAY DISPLAY
/ SOLUTIONS TO THE STATE EQUATIONS FOR
/ ANY DESIRED SOLUTION PARAMETERS WHICH
/ CAN BE READ IN USING DISPLAY MODE.
/

```

```

DISPLA LAC (DISPLA      /SETUP DISPLAY
      DAC TERM1        /CYCLE
      LAW -333         /WAIT DELAY
      DAC DELAY
      ISZ DELAY
      JMP .-1
      LAC A2
      ADAØL!ADAØX      /MDAC
      LAC A3
      ADAIL!ADAIX      /MDAC
      JMP DISRST

```

```

/
/ THE OPTIMIZATION RESULTS ARE DISPLAYED
/ ON THE CRT-DISPLAY WITH THE FORTRAN IV
/ SUBROUTINE "OUTPUT". THE OPTIMAL SOLU-
/ TIONS ARE SIMULTANEOUSLY DISPLAYED ON
/ A FOUR-TRACE OSCILLOSCOPE. A TELETYPE
/ COPY OF THE OPTIMIZATION RESULTS IS
/ OBTAINED WITH A "CONTROL-P" COMMAND ON
/ THE CRT-KEYBOARD.
/

```

```

PRINTI LAC (3          /FOR TELETYPE
      DAC IOMODE        /PRINTOUT
PRINT ACRL+1Ø          /RESET ASTRAC II
      ION
      JMS* OUTPUT      /DISPLAY PARAMETERS
      JMP .+2
      .DSA VARBL
      IOF
      LAC IOMODE
      SAD (Ø
      JMP DISPLA
      JMP INTAKE      /AVERAGING MODE

```

```

/
/ ALL THE PARAMETERS ARE STORED IN THE

```

BEOPT2

PAGE 10

/MACRO MAIN PROGRAM IN A 100-ELEMENT
 /ARRAY.THIS FACILITATES COMMUNICATION
 /WITH THE FORTRAN IV SUBROUTINES.

/

/PARAMETER STORAGE

/ IN THE FOLLOWING, THE NUMBER TO THE
 /EXTREME RIGHT OF EACH VARIABLE IS THE
 /SUBSCRIPT OF THE CORRESPONDING ELEMENT
 /OF THE J-ARRAY IN SUBROUTINE INPUT AND
 /SUBROUTINE OUTPUT.

/

	.DSA 144	/ARRAY DESCRIPTOR
	.DSA 0	/BLOCK
	.DSA 0	
VARBL	.DSA A1	
A1	0	/1
A2	0	
A3	0	
A4	0	
A5	0	/5
AS1	0	
AS2	0	
AS3	0	
AS4	0	
AS5	0	/10
AD1	0	
AD2	0	
AD3	0	
AD4	0	
AD5	0	/15
BITS1	0	
BITS2	0	
BITS3	0	
BITS4	0	
BITS5	0	/20
WBITS1	0	
WBITS2	0	
WBITS3	0	
WBITS4	0	
WBITS5	0	/25
LBITS1	0	
LBITS2	0	
LBITS3	0	
LBITS4	0	
LBITS5	0	/30
WINK1	0	
WINK2	0	
WINK3	0	

BBOPT2

PAGE 11

WINK4	0	
WINK5	0	/35
LINK1	0	
LINK2	0	
LINK3	0	
LINK4	0	
LINK5	0	/40
F1	0	
F2	0	
F3	0	
F4	0	/44
MF1	0	
MF2	0	
MF3	0	
MF4	0	/48
MIF1	0	
MIF2	0	
MIF3	0	
MIF4	0	/52
MD1	0	
MD2	0	
MD3	0	
MD4	0	/56
MEP1	0	
MEP2	0	
MEP3	0	
MEP4	0	/60
DBIT1	0	
DBIT2	0	
DBIT3	0	
DBIT4	0	
DBIT5	0	/65
.BLOCK 5		
PBIN	0	/71
MBIN	0	
TIME	0	/73
TSF	0	
TIMAX	0	
RUNS1	0	/76
RUNS2	0	
RUNS3	0	
RUNS4	0	
IOMODE	0	/80
CNT1	0	/81
CNT2	0	
RST1	0	/83
RST2	0	
.BLOCK 3		

BBOPT2

PAGE 12

REPI 0 /88
REPI5 0
DELAY 0 /90
 .BLOCK 12
 .END SEARCH

APPENDIX 2

SUBROUTINE INPUT

INPUT

PAGE 1

```

SUBROUTINE INPUT(J)
C SECOND-ORDER EXAMPLE
  DIMENSION J(100),R(10)
  DATA SF/13107.2/
  IF(J(80).EQ.1)GO TO 500
  NEXT=2
1  WRITE(2,2)
  READ(3,10)IKODE
  WRITE(2,3)
  IKODE=IKODE+1
  GO TO(400,300,20,30,40,50,60,
1    70,80,90,100,110,120,
2    130,140,150),IKODE
20 WRITE(2,21)
  READ(3,10)J(26),J(28)
  GO TO(30,1),NEXT
30 WRITE(2,31)
  READ(3,10)J(36),J(38)
  GO TO(40,1),NEXT
40 WRITE(2,41)
  READ(3,10)J(21),J(23)
  GO TO(50,1),NEXT
50 WRITE(2,51)
  READ(3,10)J(31),J(33)
  GO TO(60,1),NEXT
60 WRITE(2,61)
  READ(3,10)J1,J2
  J(81)=-J1
  J(82)=-J2
  GO TO(70,1),NEXT
70 WRITE(2,71)
  READ(3,10)J1,J2
  J(83)=-J1
  J(84)=-J2
  GO TO(80,1),NEXT
80 WRITE(2,81)
  READ(3,11)R1,R2
  J(49)=-R1*SF
  J(50)=-R2*SF
  GO TO(90,1),NEXT
90 WRITE(2,91)
  READ(3,11)R1,R2
  J(53)=-R1*SF
  J(54)=-R2*SF
  GO TO(100,1),NEXT
100 WRITE(2,101)
  READ(3,11)R1,R2
  J(57)=-R1*SF

```



```
J(53)=-R2*SF
GO TO(110,1),NEXT
110 WRITE(2,111)
    READ(3,11)TIMAX
    J(75)=TIMAX
    J(74)=131072.0*3.0/TIMAX
    GO TO(120,1),NEXT
120 WRITE(2,121)
    READ(3,11)R1
    J(71)=R1*SF
    J(72)=-J(71)
    NEXT=2
    GO TO 1
C SET STANDARD PARAMETERS
130 J(26)=5
    J(28)=J(29)=11
    J(36)=J(38)=-2
    J(21)=0
    J(23)=J(24)=13
    J(31)=0
    J(33)=J(34)=4
    J(81)=-100
    J(82)=-1000
    J(83)=-1
    J(84)=-2
    J(49)=J(50)=-5.00*SF
    J(53)=J(54)=-1.00*SF
    J(57)=J(58)=-0.05*SF
    J(75)=1000
    J(74)=131072.0*3.0/FLOAT(J(75))
    J(71)=10.0*SF
    J(72)=-J(71)
    WRITE(2,131)
    DO 132 K=1,80000
132 CONTINUE
    GO TO 1
140 WRITE(2,141)
    DO 143 K=1,3
    WRITE(2,142)K
143 READ(3,11)R(K)
    J(1)=-R(1)/3.0
    J(2)=R(2)*SF
    J(3)=R(3)*SF
    J(80)=2
    RETURN
150 WRITE(2,151)
    NEXT=1
    GO TO 20
```

INPUT.

PAGE 3

```

300  WRITE(2,301)
      READ(3,10)J(88)
      J(89)=J(88)
      J(80)=1
      GO TO 450
400  J(80)=0
      J(89)=1
450  WRITE(2,451)
      CALL CLKON
500  J(76)=0
      J(77)=0
      J(78)=0
      RETURN

```

C FORMAT STATEMENTS

```

2  FORMAT(*-$$$ OPTIMIZATION *
   1 *PARAMETERS $$$*//
   1 9X* 0.SINGLE RUN */
   1 9X* 1.AVERAGING */
   1 9X* 2.LBITS1,3 */
   1 9X* 3.LINK1,3 */
   1 9X* 4.WBITS1,3 */
   1 9X* 5.WINK1,3 */
   1 9X* 6.CNT1,2 */
   1 9X* 7.RST1,2 */
   1 9X* 8.MIF1,2 */
   1 9X* 9.MD1,2 */
   1 9X*10.MEP1,2 */
   1 9X*11.MAXIMUM TIME*/
   1 9X*12.BINARY DECISION*/
   1 9X*13.STANDARD PARAMETERS*/
   1 9X*14.DISPLAY ONLY*/
   1 9X*15.CONSECUTIVE INPUT*//
   1 *+ IKODE=*)
3  FORMAT(*-*)
10  FORMAT(4I7)
11  FORMAT(4F10.0)
21  FORMAT(*+LBITS1,3=*)
31  FORMAT(*+ LINK1,3=*)
41  FORMAT(*+WBITS1,3=*)
51  FORMAT(*+ WINK1,3=*)
61  FORMAT(*+ CNT1,2=*)
71  FORMAT(*+ RST1,2=*)
81  FORMAT(*+ MIF1,2=*)
91  FORMAT(*+ MD1,2=*)
101 FORMAT(*+ MEP1,2=*)
111 FORMAT(*+MAXIMUM *
1  *TIME(MICRO-SEC)=*)
121 FORMAT(*+MAGNITUDE OF *

```

INPUT

PAGE 4

```
1 *BINARY DECISION=*)
131  FORMAT(////////10X*STANDARD *
1 *PARAMETERS*//15X*ARE SET*)
141  FORMAT(3X*TEST PARAMETERS*//)
142  FORMAT(*+      A*I1*=*)
151  FORMAT(7X* $$$ CONSECUTIVE *
1 *INPUT $$$*/)
301  FORMAT(*+REPETITIONS=*)
451  FORMAT(////////9X*$*18X*$*//
1 13X*OPTIMIZATION*//
1 13X*IN  PROGRESS*//
1 9X*$*18X*$*)
END
```

APPENDIX 3

SUBROUTINE OUTPUT

OUTPUT

PAGE 1

```

      SUBROUTINE OUTPUT(J)
C SECOND-ORDER EXAMPLE
      DIMENSION J(100),AV(10),
1         R(10),F(15)
      DATA AV(1),AV(2),AV(3),AV(4),
1         AV(5),AV(6),AV(7),AV(8),
1         AV(9),AV(10)/10*0.0/
      DATA SF/13107.2/
      IOMODE=J(80)+1
      GO TO(50,100,200,300),IOMODE
C
C CRT DISPLAY OUTPUT
50    CALL CLKRD(ITIME)
      SEC=FLOAT(ITIME)/60.
10    WRITE(2,11) J(89),J(76),J(78),SEC
60    J13=-J(1)*3
      WRITE(2,12) J13
      DO 13 K=2,3
        R(K)=FLOAT(J(K))/SF
13    WRITE(2,14) K,R(K)
C IF DISPLAY MODE,EXIT
      IF(J(80).EQ.2)GO TO 316
      WRITE(2,17)
      DO 70 K=1,12
        KP=K+48
70    F(K)=-FLOAT(J(KP))/SF
      WRITE(2,71) F(1),F(2)
      WRITE(2,72) F(5),F(6)
      WRITE(2,73) F(9),F(10)
      WRITE(2,9)
      WRITE(2,18) J(21),J(23)
      WRITE(2,19) J(26),J(28)
      WRITE(2,20) J(31),J(33)
      WRITE(2,21) J(36),J(38)
      J81=-J(81)
      J82=-J(82)
      WRITE(2,22) J81,J82
      J83=-J(83)
      J84=-J(84)
      WRITE(2,23) J83,J84
      WRITE(2,24) J(75)
      DO 30 K=1,10
30    AV(K)=0.0
      RETURN
C
C AVERAGING OF PARAMETERS
100   DO 35 K=1,3
35    AV(K)=AV(K)+FLOAT(J(K))

```

OUTPUT

PAGE 2

```

      AV(6)=AV(6)+FLOAT(J(76))
      AV(7)=AV(7)+FLOAT(J(78))
      J(88)=J(88)-1
      IF(J(88).EQ.0)GO TO 150
      RETURN
150   CALL CLKRD(ITIME)
      SEC=FLOAT(ITIME)/60.
      REP=FLOAT(J(89))
      SEC=SEC/REP
      J(1)=(AV(1)/REP)+0.5
      DO 160 K=2,3
160   J(K)=AV(K)/REP
      J(76)=AV(6)/REP
      J(78)=AV(7)/REP
      J(80)=0
      GO TO 10
C
C CRT DISPLAY OF TEST PARAMETERS
200   WRITE(2,201)
      GO TO 60
C
C TELETYPE OUTPUT
300   WRITE(4,9)
      WRITE(4,11) J(89),J(76),J(78),SEC
      WRITE(4,12) J13
      DO 313 K=2,3
313   WRITE(4,14) K,R(K)
      WRITE(4,9)
      WRITE(4,17)
      WRITE(4,71) F(1),F(2)
      WRITE(4,72) F(5),F(6)
      WRITE(4,73) F(9),F(10)
      WRITE(4,9)
      WRITE(4,18) J(21),J(23)
      WRITE(4,19) J(26),J(28)
      WRITE(4,20) J(31),J(33)
      WRITE(4,21) J(36),J(38)
      WRITE(4,22) J81,J82
      WRITE(4,23) J83,J84
      WRITE(4,24) J(75)
316   J(80)=0
      RETURN
C FORMAT STATEMENTS
9     FORMAT(* *)
11    FORMAT(*- $$$ FINAL SOLUTION *
1     *PARAMETERS $$$*//* AVERAGED OVER
1     *I4* OPTIMIZATIONS WITH:*/1X16
1     * ASTRAC II RUNS;*I4* SUCCESSES;*
```

OUTPUT

PAGE 3

```

1  /* AND*F7.1* SECONDS PER *
1  *OPTIMIZX*A1=*I7* MICRO-SEC*)
12  FORMAT(8X*A1=*I7* MICRO-SEC*)
14  FORMAT(8X*A*I1=*F7.2* VOLTS*)
17  FORMAT(/* $$$ OPTIMIZATION *
1  *PARAMETERS $$$*/)
71  FORMAT(* MIF1,2=*
1  F6.3*,*F6.3)
72  FORMAT(* MD1,2=*
1  F6.3*,*F6.3)
73  FORMAT(* MEP1,2=*
1  F6.3*,*F6.3)
18  FORMAT(* WBITS1,3=*
1  I6*,*I6)
19  FORMAT(* LBITS1,3=*
1  I6*,*I6)
20  FORMAT(* WINK1,3=*
1  I6*,*I6)
21  FORMAT(* LINK1,3=*
1  I6*,*I6)
22  FORMAT(8X*CNT1,2=*I6*,*I6)
23  FORMAT(8X*RST1,2=*I6*,*I6)
24  FORMAT(* MAXIMUM COMPUTE PERIOD=*
1  I5* MICRO-SEC*)
201  FORMAT(*-*///8X*$$$ TEST *
1  *SOLUTION $$$*/)
END

```

APPENDIX 4

SUBROUTINE ARACK

ARACK

PAGE 1

```
      SUBROUTINE ARACK(J)
      DIMENSION J(100)
      CALL SSWTCH(4,L)
      IF (L.EQ.1)GO TO 99
      WRITE(2,5)
5      FORMAT(*-ARACK*)
      DO 100 K=1,5
      READ(3,10)M
10     FORMAT(I7)
      WRITE(2,5)
      DO 100 I=1,20
      N=(K-1)*20+I
      WRITE(2,500)N,J(N)
500    FORMAT(* J(*,12,*)=*,I8)
100    CONTINUE
      READ(3,10)M
99     RETURN
      END
```

LIST OF REFERENCES

Korn, Granino A., and Theresa M. Korn

1964 Electronic Analog and Hybrid Computers, McGraw-Hill Book Company, New York.

1968 Mathematical Handbook for Scientists and Engineers, McGraw-Hill Book Company, New York.

Maybach, R. L.

1967 The Solution of Optimal Control Problems on an Iterative Hybrid Computer, Ph.D. Dissertation, Department of Electrical Engineering, The University of Arizona.

O'Grady, E. P.

1969 Interface, Design, Test, and Application of a Hybrid-computer. Ph.D. Dissertation, Department of Electrical Engineering, The University of Arizona.

Pontryagin, L. S., B. G. Boltyansky, R. V. Gamkrelize, and E. M. Mischenko

1962 The Mathematical Theory of Optimal Processes, John Wiley and Sons, Inc., New York.

Stewart, E. C., W. P. Kavanaugh, and D. H. Brocker

1967 Study of a Global Search Algorithm for Optimal Control, Presented at the Fifth Congress of the International Association for Analog Computation at Lausanne, Switzerland.