PARAMETER OPTIMIZATION BY RANDOM SEARCH USING HYBRID COMPUTER TECHNIQUES *

G. A. Bekey

University of Southern California, Los Angeles, California

and

M. H. Gran, A. E. Sabroff, A. Wong

TRW Systems, Redondo Beach, California

INTRODUCTION

Optimum selection of the parameter values for a complex dynamic system usually consists of three distinct phases: (1) a proposed system configuration is selected, in which only parameter values remain as unknowns; (2) one or more performance or cost criteria for evaluation of the system are selected; and (3) a computer technique or algorithm is chosen for adjusting the system parameters until an optimum value of the criterion function is achieved. Typical algorithms are those based on relaxation or steep descent methods. However, both of these methods are primarily suited to optimization of criterion functions with unique minima or maxima. Furthermore, they may fail to converge or may converge only very slowly if the criterion function—parameter space exhibits "ridges" or if the criterion function is only piecewise differentiable or piecewise continuous. Both of these difficulties are likely to arise in connection with nonlinear systems. This paper presents an approach to finding a global optimum by means of a modified sequential random perturbation technique implemented on a hybrid computer.

Random search techniques for parameter optimization were originally proposed by Brooks. They were successfully implemented on analog computers by Munson and Rubin and Favreau and Franks. A hybrid computer implementation using only two parameters, a fixed step size, and an algebraic criterion function was studied by Mitchell. This paper extends the previous work by applying it to a nonlinear dynamic system with nine parameters. Furthermore, the effects of initial conditions and analog computer errors are included in the optimization program in a systematic way. The method is illustrated by application to the optimization of a satellite acquisition system.

PROBLEM FORMULATION

The dynamical system to be optimized is described by the differential equation:

\[ \dot{x} = f(x; t; a) \] (1)
where $\mathbf{x}$ is the $(n \times 1)$ state vector, $f$ is an $(n \times 1)$ vector function, $g$ is an $(m \times 1)$ parameter vector and $t$ represents running time. It is desired to study this dynamic system over a large class of initial conditions. Define $X_o$ as the set of all $(n \times 1)$ initial state vectors which are of interest to be studied and an element of $X_o$ as $\mathbf{x}_o$. A unique solution of (1) is solely dependent on $\mathbf{x}_o$, $g$ and $t$ and therefore can be represented as:

$$\mathbf{x} = \mathbf{x} (\mathbf{x}_o; t; g)$$  \hspace{1cm} (2)

A cost or criterion function can be written ordering the desirability of the particular choice of $g$ for a given $\mathbf{x}_o$ as:

$$J (\mathbf{x}_o; g) = \int_0^t g ((\mathbf{x}_o; t; g) dt)$$  \hspace{1cm} (3)

where $g$ is a scalar function of $\mathbf{x}_o$, $g$ and $t$. Examples of $J$ may be fuel consumed or time required for satellite acquisition for a given parameter and initial condition set.

For a given initial condition and parameter setting, equation (3) provides a scalar value describing the "quality" of the dynamic system in a quantitative fashion. As it is desired to study the effect of the parameter settings over the entire space of initial conditions, a new criterion function must be defined as:

$$F (g) = \int_{X_o} h [J (\mathbf{x}_o; g)] d X_o$$  \hspace{1cm} (4)

where $h$ is a scalar function of the functional $f(\mathbf{x}_o; g)$ and the integral is a Riemann-Stieltjes integral which allows integration of discrete $X_o$ spaces as well as continuous. This provides a single measure of the "quality" of a selection $g$ over the entire space of initial conditions. Examples of $F$ studied in this paper are:

$$F_1 = \max_{\mathbf{x}_o \in X_o} f (\mathbf{x}_o, g)$$  \hspace{1cm} (5a)

$$F_2 = \sum_{j=1}^q f (\mathbf{x}_{o_j}, g)$$  \hspace{1cm} (5b)

$$F_3 = \sum_{j=1}^q \max_{\mathbf{x}_{o_j} \in X_i} f (\mathbf{x}_{o_j}, g)$$  \hspace{1cm} (5c)

For a given criterion functional $F$, a computer algorithm is desired which finds the optimum $g$ which will be denoted by $g^*$,

$$F^* (g^*) = \min_{g} F (g)$$  \hspace{1cm} (6)

Values of $g^*$ for different criteria will, in general, be different. With nonlinear differential equations, the criterion functions cannot be assumed to possess a unique minimum or maximum, and the optimization algorithms must be designed to seek the global extremum. Since for any physical problem, the allowable range of all parameters is limited, admissible parameter vectors will be constrained to the allowable region of the parameter space. Local continuity of the criterion function will be assumed.

Exhaustive search for an optimum design, in which parameter values are quantized and all possible parameter combinations are tested, is clearly limited to systems with only a few parameters. Consequently, an organized search procedure is required.

Let the initial parameter choice be indicated by $g^{(0)}$, so that the initial criterion function is

$$F_0 = F (g^{(0)})$$  \hspace{1cm} (7)

If the parameter adjustment is made according to the gradient method, then a parameter increment is computed from

$$\Delta g^{(0)} = k \nabla F (g^{(0)})$$  \hspace{1cm} (8)

where $k$ is a scaling matrix.

Gradient methods suffer from three major disadvantages for systems of the type being considered here. First, the computation of the gradient requires $m$ trial steps to determine each component; second, the method leads only to a local minimum of the criterion function, and third, the gradient method encounters significant difficulties if the criterion function exhibits "ridges" or "narrow valleys" in the parameter space.

AN ALGORITHM FOR RANDOM SEARCH OPTIMIZATION

Strictly speaking, pure random search refers to a computation of the criterion function at a number of randomly chosen points in the parameter space, and selection of the particular parameter values ($a_1$, $a_2$) yielding the smallest value of $F(a)$. However, such a sequence of randomly selected parameter vectors does not take advantage of the local continuity.
PARAMETER OPTIMIZATION BY RANDOM SEARCH

properties of most criterion function surfaces. Consequently, the strategy to be discussed below should more properly be referred to as "sequential random scanning" or "random creep". Assume that the initial parameter vector is again designated by \( \mathbf{a}^{(0)} \). Now, choose an increment \( \Delta \mathbf{a}^{(0)} \) by selecting the individual parameters \( \Delta a_i^{(0)}, i = 1, 2, \ldots, m \) from \( m \) Gaussian sequences of random numbers with mean zero and variance \( c_i \). Then, if the \( m \) random sequences are independent, the orientation and length of the parameter increment \( \Delta \mathbf{a}^{(0)} \) will be random, and a trial value

\[
\mathbf{a}' = \mathbf{a}^{(0)} + \Delta \mathbf{a}^{(0)}
\]

is obtained. The criterion function \( F' = F'(\mathbf{a}') \) is computed and compared to \( F_0 = F(\mathbf{a}^{(0)}) \). If there is an improvement, the parameters are updated by letting \( \mathbf{a}' = \mathbf{a}' \). If there is no improvement, the trial step is abandoned and a new trial step is chosen. This basic strategy is illustrated in the flow chart of Fig. 1.

Now from the standpoint of computer implementation, the differential equations can be solved on the analog computer for each trial value \( \mathbf{a}' \). The random increments \( \Delta \mathbf{a}_i \) can be obtained by sampling analog noise generators, by generating pseudo-random sequences in the digital computer, or by construction of special devices such as shift register noise generators with several independent outputs.

**Modifications of the Basic Algorithm**

In order to take maximum advantage of the properties of the criterion surface, the basic strategy can be modified in a number of ways. Successive steps can be made correlated in such a way as to favor successful direction. The mean of the distribution of steps can be biased in the direction of a successful step, or after a specified number of successive successes. Thus, the \( j \)th trial step could be computed from

\[
\Delta \mathbf{a}^{(j+1)} = C^{(j)} \mathbf{N}^{(j)} + \mathbf{b}^{(j)}
\]

where \( \mathbf{N}^{(j)} \) is a column vector of Gaussian random samples, \( C^{(j)} \) is a diagonal matrix of variances, and \( \mathbf{b}^{(j)} \) is a bias vector, which may be altered after one or more successes (or failures).

In the present study, the term "absolute biasing" has been used to denote the repeated use of a successful random step as long as continued success is attained. That is, if \( \Delta \mathbf{a}^{(j)} \) is successful, we choose

\[
\Delta \mathbf{a}^{(j+1)} = \Delta \mathbf{a}^{(j)}
\]

and test for success. If \( \Delta \mathbf{a}^{(j)} \) was a failure, one chooses

\[
\Delta \mathbf{a}^{(j+1)} = -\Delta \mathbf{a}^{(j)}
\]

Such a technique will be referred to as "absolute positive and negative directional biasing."

It is also possible to adjust the variance of the distribution of step sizes. For example, as a local minimum is approached, the variance can be decreased in order to decrease the probability of overshooting the optimum.

The basic algorithm logically divides into two parts, concerned with the search for a local minimum and the global minimum respectively.

**Search for a Local Minimum**

This strategy, using absolute positive and negative biasing, is illustrated in Fig. 2. It consists of the following steps for the computation of the \( j \)th trial step:

1. A Gaussian random vector \( \mathbf{N}^{(j)} \) is obtained. In the present study the components \( N_i^{(j)}, i = 1, 2, \ldots, m \), were obtained from successive trial samples of an analog high-frequency noise generator. The sampling frequency was sufficiently low compared to the noise

![Figure 1. The basic algorithm.](From the collection of the Computer History Museum (www.computerhistory.org))
Figure 2. Absolutely biased local random search flow diagram.

bandwidth to insure that successive samples are essentially uncorrelated.

2. A trial step is computed with magnitude constraints imposed on all parameters.

3. The analog computer is used to compute a new value of $F = F'$.

4. $F'$ is compared with $F^{(j)}$. If $F' < F^{(j)}$, we let $F' = F^{(j+1)}$ and $a' = a^{(j+1)}$. If $F' > F^{(j)}$, the increment $-\Delta a^{(j)}$ is tried.

5. The number of trials which leads to either no change or an increase in $F$ is counted and used as a stopping criterion.

Search for the Global Optimum

Once a local optimum has been found, it must be tested to determine whether it is indeed the global optimum. The approach selected in this study is a random search. This allows much flexibility in that the statistics of the search may be adjusted to correspond to estimates of the location of other likely optima. For the present case, it was felt the space near the local optimum was the most likely location for an even better optimization criterion. The strategy then consisted of randomly sampling numbers corresponding to the parameters (as in the local random search), starting with a very small variance and expanding this variance slowly if no better points are found. If an improvement is found, a local search strategy is once again initiated. Figure 3 gives the flow detail of the global random search algorithm.

Initial Condition Set Selection

Equation (3) suggests that natural criterion functions such as fuel consumption or acquisition time are functions of the initial conditions as well as parameter values. If both the differential equation and criterion function $(J)$ were linear, a single set of parameters could be found which would optimize the $J$ for all initial conditions simultaneously. The present study is concerned with the more general case where both the differential equations and the criterion function may be nonlinear. In this case, the same set $a$ may not optimize $J$ for various initial conditions. In order to provide a single criterion function, the $J$'s must be synthesized in some fashion as indicated by Eq. (4). As the allowable I.C. space $X_0$ is generally a continuous set, an infinite number of I.C.'s exist. Apparently not all I.C.'s can be considered. The twofold problem then exists: (1) which
initial conditions should be considered, and (2) how the synthesis should be accomplished.

If the criterion space is reasonably smooth it can best be described by partitioning the space and inspecting an initial condition from each partition. For the case of six variables (and, therefore, six initial conditions), as may exist for the satellite dynamic and kinematic equations, quantizing each variable into as few as five values yields \(5^6 = 15,625\) different initial conditions. This increased cost due to high dimensions has been aptly described by Bellman as the "curse of dimensionality." 

The synthesis function \(h\) may serve to reduce the dimensionality. Frequently all that is desired is a reasonable set of initial conditions which will describe the worst possible conditions for \(F\). This is the minimax criterion:

\[
F^* (x^*) = \min_{\alpha} \max_{x_0 \in X} J (x_0, \alpha) \quad (13)
\]

In some cases, the general location of this worst case can be estimated eliminating much of the initial condition space. The same algorithm employed to minimize \(J\) with respect to \(\alpha\) can be used to maximize \(J\) with respect to the initial conditions. This approach has been programmed but the results are not included in this paper. The minimax criterion may unfairly penalize the more likely cases. For this reason, the synthesis procedure used in this paper was the most obvious linear relation—an unweighted average:

\[
F = \sum_{j=1}^{q} J (x_{j0}, \alpha) \quad (14)
\]

The most desirable aspects of Eqs. (13) and (14) can be combined by the following procedure. The I.C. space is divided into \(q\) subdivisions which we designate by \(X_1, \ldots, X_q\). For a given \(\alpha\) each of these subspaces are searched for the maximum initial condition set. These maximum I.C. points are then algebraically summed as in Eq. (14). Equation (15) expresses the operation:

\[
F = \sum_{j=1}^{q} J^* (x_{j0}, \alpha) \quad (15)
\]

where \(x_{j0} \in X_j\) and

\[
J^* (x_{j0}, \alpha) = \max_{x_{j0} \in X_j} J (x_{j0}, \alpha)
\]

As this type of search procedure for every trial \(\alpha\) is prohibitively expensive in computer time, the maximization is only performed after a given number of improved \(\alpha\)'s have been found. Experience with the acquisition simulation indicates these worst I.C.'s remain nearly the worst case for a large variation in \(\alpha\).

**A SATELLITE ACQUISITION OPTIMIZATION**

The general acquisition problem considered is that of aligning a single axis of a satellite parallel to a desired vector and driving the angular rotation about this axis to zero (one-axis acquisition problem). The reference coordinate frame is a three-dimensional, Cartesian set \((\hat{i}, \hat{j}, \hat{k})\) with the three axes defining the desired pointing direction. The kinematic representation consists of the three direction cosines of the satellite axis to be aligned with the three reference axes. The control system can consist of any collection of sensors (that can be described by the six variables), whose outputs are processed by a compensation network which can be described by the parameters to be optimized. The outputs of the networks are then used to drive angular acceleration devices (torquers). In order to provide a more concrete control equation, the sensors are assumed to be three rate sensors and two direction cosine sensors. The control laws were chosen to be proportional but saturable control. The equations expressing the acquisition dynamics, kinematics, control law, several potential optimization criteria and the end of run criterion are given in Table 1. Six initial conditions (three attitudes and three rates) were specified for each trial optimization.

**Analog Simulation**

The above-mentioned equations were simulated on a Beckman 2132 analog computer. The end of run criterion indicated in Table 1 is necessary to determine when acquisition is complete. This is especially critical when the time of acquisition is the optimization criterion. The criterion selected is the absolute value of a weighted sum of the variables biased by some fixed voltage (see "Computer Error Detection" below). The criterion is assumed satisfied when this sum becomes zero. The bias is necessary as noises and drifts by the analog would prevent an unbiased zero from ever occurring. The bias level selected is more a function of actual voltage levels than their equivalent variable units.
Table 1. Simulated Acquisition Equations

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Equations</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinematics</td>
<td></td>
<td>$\dot{a}<em>{13} = \omega</em>{13} a_{23} + \omega_{43} a_{23}$</td>
<td>$a_{13}$—Direction cosine from $i$ body axis to $3$ inertial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{a}<em>{23} = \omega</em>{13} a_{23} - \omega_{43} a_{13}$</td>
<td>$\omega_i$—Body rate about $i$ body axis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{a}<em>{33} = \omega</em>{23} a_{13} - \omega_{13} a_{23}$</td>
<td></td>
</tr>
<tr>
<td>Dynamics</td>
<td></td>
<td>$\dot{\omega}<em>i = [(I</em>{i+1} - I_i) / I_i] \omega_i \omega_3 + \gamma_i$</td>
<td>$I_i$—Inertia about $i$ principal inertia axis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{\omega}<em>2 = [(I</em>{i+1} - I_i) / I_i] \omega_2 \omega_3 + \gamma_2$</td>
<td>$\gamma_i$—Control angular acceleration about $i$ axis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{\omega}<em>3 = [(I</em>{i+1} - I_i) / I_i] \omega_3 \omega_2 + \gamma_3$</td>
<td></td>
</tr>
<tr>
<td>Control laws</td>
<td></td>
<td>$\gamma_1 = - a_1 (a_{23} + a_3 a_1)$</td>
<td>$a_i$ ($i = 1, 2, 3, 4, 5$)—Control law gain constants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\gamma_2 = - a_2 (-a_{13} + a_4 a_2)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\gamma_3 = - a_3 a_3$</td>
<td></td>
</tr>
<tr>
<td>End-of-run</td>
<td></td>
<td>$3 \sum b_i \omega_i + 2 \sum c_i a_{13} - K = 0$</td>
<td>$b_i, c_i$—Weighting and scaling constants</td>
</tr>
<tr>
<td>criterion</td>
<td></td>
<td></td>
<td>$K$—a constant</td>
</tr>
<tr>
<td>Minimum fuel</td>
<td></td>
<td>$F = \int^T_0 \sum \gamma_i \ dt$</td>
<td></td>
</tr>
<tr>
<td>criterion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum time</td>
<td></td>
<td>$F = \int^T_0 \sum \gamma_i \ dt$</td>
<td></td>
</tr>
<tr>
<td>criterion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td>$a_{13}, a_{23}$</td>
<td>Limits on magnitude of $\gamma_1, \gamma_2, \gamma_3$, respectively</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{43}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{10}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{15}, a_{25}, a_{35}$</td>
<td>Saturation values of $a_{10}$ and $a_{23}$, respectively</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Digital-Analog Interface**

The digital computer provides those functions it is best suited for. It provides the optimization algorithm, initialization search procedure, controls the analog modes (Initial Condition, Operate), provides voltages for the initial conditions of the variables and parameter settings, and reads the optimization criterion from the analog.

In order to better understand the operations of the various elements of the optimization process a logical flow diagram of the entire computer operations in the optimization phase is shown in Fig. 4. An optimization criterion is selected (for example, minimum fuel or time). A set of initial conditions for the variables is selected based on the space search that has already been run. A best initial guess for the parameters is made. The error criterion ($F$) for this initial setting will be assumed already calculated.

A new point in parameter space is selected by the optimization algorithm. For each set of variable initial conditions, three analog computer runs are made giving a $J$ for each run. The three $J$'s are compared to check parity (the reasons are explained later). If their standard deviation is within tolerance, the mean $J$ for the three is selected as the criterion for this set of I.C.'s. This is the innermost loop (loop I). This process is repeated for each set of I.C.'s (loop II). The $J$'s for each I.C. set is averaged to give $F$. This is the $F$ used to determine (loop III) whether the explored point is better than the present point. Loop III minimizes time or fuel as the error criterion.

**COMPUTER ERROR DETECTION**

The speed and accuracy of convergence of the optimization procedure is directly related to the
accuracy and repeatability of the analog simulation. Extensive testing of the analog computer found that for many runs in succession or even on separate days with identical inputs, the optimization criterion would be repeatable to within one volt unless an obvious malfunction occurred. The prime sources of malfunctions were: (1) a bit lost in the analog to digital converter, (2) a momentarily defective electronic switch, and (3) drift in the electronic multipliers. Most malfunctions were of a momentary nature so that for over 99% of the time no more than two successive runs were adversely affected. In light of this knowledge, the computer was programmed in the following manner to prevent computer malfunctions from negating an algorithm or consuming excessive time in trouble shooting.

**Overload Detection**

An overload indicates either an unstable differential equation or an equipment malfunction. The computer was programmed to halt on overload.

**Time Limit Detection**

Certain parameter combinations may result in extremely long optimization times. Occasionally, however, a computer malfunction could have the same effect. The computer was programmed to stop on a maximum time and try the problem again. If time is exceeded again, it is printed out, a large value is assigned to the optimization criterion and the optimization continues automatically.

**Standard Deviation Tests**

The use of confidence tests provides a powerful tool for improving the accuracy of analog computer studies. Assume that errors in analog computer results are normally distributed with zero mean and a known standard deviation \( \sigma \). The assumption of zero mean can be justified if the computer is balanced frequently and \( \sigma \) can be estimated from the sample variance \( \bar{V} \). Then, confidence tests can be used to determine the number of runs needed to satisfy a particular accuracy criterion. For example, suppose it is desired to be confident with .95 probability that the computer has no more noise than when \( \sigma \) was estimated. For \( N = 3 \) runs, we require that \( \sqrt{\bar{V}} \leq 3.0 \sigma \). When the allowable variance is exceeded, it is concluded that the computer is not operating properly. The computer makes three runs with the same inputs. If the standard deviation is less than one volt, operation is assumed normal and the optimization continues. If the standard deviation is greater than a volt, malfunction is assumed. In this case, the large standard deviation is printed out and another set of three runs attempted. If this set is accepted, the first set is discarded and the optimization is continued. The operator may stop the computation if he desires.

Note that if a set of runs is accepted, the three values of \( F \) are averaged, thus increasing the accuracy of the optimization criterion by \( \sqrt{3} \).

**RESULTS OF THE OPTIMIZATION STUDY**

The specific numerical values used in the optimization are given in Table 2. A maximum of nine independent parameters were studied at one time. Fig. 5 shows a typical optimization using acquisition time as the criterion for the absolutely biased local random search. The nine parameter settings are plotted versus real time. The vertical lines indicate when the computer is in reset. Three trials are simulated and compared before the case is accepted as discussed in the preceding section. The optimization runs shown considered just one initial condition set. To indicate the improvements in system performance during the optimization for
Table 2. Numerical Values for Acquisition Optimization Study

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Assumed Value or Range of Values</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega_{im}$</td>
<td>Maximum initial rate</td>
<td>$\pm 70$</td>
<td>milliradians/sec</td>
</tr>
<tr>
<td>$\omega_{ia}$</td>
<td>Average initial rate</td>
<td>$\pm 26$</td>
<td>milliradians/sec</td>
</tr>
<tr>
<td>$a_{iin}$</td>
<td>Nominal direction cosines</td>
<td>$a_{12} - a_{23} = 0, a_{33} = +1$</td>
<td></td>
</tr>
<tr>
<td>$a_{imin}$</td>
<td>Minimum initial attitudes</td>
<td>$\pm 1$ a</td>
<td></td>
</tr>
<tr>
<td>$a_{imax}$</td>
<td>Maximum initial attitudes</td>
<td>$+1, \pm 1/\sqrt{3}$ a</td>
<td></td>
</tr>
<tr>
<td>(I$_3$ - I$_1$)/I$_1$</td>
<td>Inertia ratios</td>
<td>$0.311$</td>
<td></td>
</tr>
<tr>
<td>(I$_2$ - I$_1$)/I$_2$</td>
<td>Inertia ratios</td>
<td>$-0.802$</td>
<td></td>
</tr>
<tr>
<td>(I$_1$ - I$_2$)/I$_3$</td>
<td>Inertia ratios</td>
<td>$0.643$</td>
<td></td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>Roll position gain</td>
<td>0.001 to 0.100</td>
<td>sec$^{-2}$</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>Pitch position gain</td>
<td>0.001 to 0.100</td>
<td>sec$^{-2}$</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>Roll rate to position gain</td>
<td>0.001 to 0.100</td>
<td>sec$^{-2}$</td>
</tr>
<tr>
<td>$\alpha_4$</td>
<td>Pitch rate to position gain</td>
<td>0.10 to 1.0</td>
<td>sec$^{-2}$</td>
</tr>
<tr>
<td>$\alpha_5$</td>
<td>Yaw rate gain</td>
<td>0.10 to 1.0</td>
<td>sec$^{-2}$</td>
</tr>
<tr>
<td>$\alpha_6$</td>
<td>Yaw torque limit</td>
<td>0.001 to 0.100</td>
<td>rad/sec$^2$</td>
</tr>
<tr>
<td>$\alpha_7$</td>
<td>Roll torque limit</td>
<td>0.001 to 0.100</td>
<td>rad/sec$^2$</td>
</tr>
<tr>
<td>$\alpha_8$</td>
<td>Pitch direction cosine limit</td>
<td>0.01 to 1.0</td>
<td>rad/sec$^2$</td>
</tr>
<tr>
<td>$\beta_10$</td>
<td>Roll direction cosine limit</td>
<td>0.01 to 1.0</td>
<td>sec$^{-2}$</td>
</tr>
</tbody>
</table>

* All physically realizable combinations.

Figure 5. Nine parameter minimum time optimization using absolutely biased local random search.

one I.C. set, Fig. 6 gives the response of five independent variables ($\omega_{i2}, \omega_{i3}, \omega_{i1}, a_{13}, a_{23}$) as defined in Table 1, the end-of-run criterion, time of run and fuel consumption for (1) the initial parameter settings, (2) the optimized minimum time settings, and (3) the optimized minimum fuel settings.

The major results of the study were the following:

1. Absolute biasing is an efficient way of improving the convergence of a random search process.
2. A successful strategy for changing the variance of the distribution of steps during the local search was not found. A subsequent study of USC$^{10}$ verified the conclusion that a uniform variance yielded convergence to a local optimum as rapidly as any variance adjustment strategy attempted. In this study, a variance equal to 4% of the range between maximum and minimum limits on each parameter was used.
3. The random global search technique proved to be very useful. The strategy that appears most useful is that of using the local optimum as the origin and initiating the purely random (no biases) search with a very small variance. After each search, the variance is widened until any point in the parameter space has some chance of being inspected. For the most successful strategy found in the study, the variance was initially set equal to 0.5% of the span (upper limit to lower limit) of each parameter which was incremented by a factor
A NOTE ON CONVERGENCE

It has been stated by Korn \(^7\) that the random search technique will converge whenever the gradient technique does. Clearly, for the local optimization algorithm, convergence can be assured since the strategy results in a sequence of criterion function values which is monotonically decreasing and bounded from below by zero. However, the rate of convergence is another matter. Rastrigin, who published one of the early papers on random search optimization,\(^{11}\) has also investigated its convergence properties.\(^{12,13}\) In the case of unimodal criterion functions, where constant F contours are hyperspheres in the parameter space, he shows that the mean rate of progress of the random search method in the gradient direction exceeds that of the steepest descent method when more than 3 parameters are involved. However, no such proof is available for the nonlinear case.

Global search, which in the limit samples the parameter space everywhere, will converge to the global optimum with probability one. However, any computer implementation is finite and therefore cannot insure the location of the global optimum.

CONCLUSIONS

Parameter optimization by sequential random perturbation is an efficient and easily programmed technique for the optimization of nonlinear dynamic systems. The technique is well suited to hybrid computation.

REFERENCES