Robust Control System Design
Using Simulated Annealing
T. Motoda, R. F. Stengel, Y. Miyazawa

Reprinted from
Journal of Guidance, Control, and Dynamics
Volume 25, Number 2, Pages 267-274

A publication of the
American Institute of Aeronautics and Astronautics, Inc.
1801 Alexander Bell Drive, Suite 500
Reston, VA 20191-4344
Robust Control System Design Using Simulated Annealing

Toshikazu Motoda*
National Space Development Agency of Japan, Minato-ku, Tokyo 105-8060, Japan
Robert F. Stengel†
Princeton University, Princeton, New Jersey 08544
and
Yoshikazu Miyazawa‡
National Aerospace Laboratory of Japan, Chofu, Tokyo 182-8522, Japan

Design parameters of a flight control system are optimized by a probabilistic method. Simulated annealing is applied for the optimization, and the downhill-simplex method is added to generate new design vector candidates. The cost function to be minimized is chosen as the probability of violating the design criteria, and it is derived by Monte Carlo evaluation that incorporates various uncertainties. Thus, the designed system is robust against these uncertainties. The feasibility of the algorithm is demonstrated by designing a control system for a simplified model. The results show that simulated annealing is more effective than the downhill-simplex method for parameter optimization, and it requires less computational time than the genetic algorithm. The Automatic Landing Flight Experiment unpowered reentry vehicle provides a second example. Simulated annealing is shown to produce a more robust longitudinal flight control design than that used in the 1996 flight experiment.

Nomenclature

\[ a, b, K, \tau \] = uncertain system parameters
\[ d \] = design parameter vector
\[ d^* \] = optimized design parameter vector
\[ e_i \] = unit vector whose ith element is 1
\[ H, X, Y \] = position of Automatic Landing Flight Experiment vehicle, m
\[ J \] = cost function (probability of mission failure)
\[ J_{\text{crit}} \] = one of the current cost functions
\[ J_{\text{high}}, J_{\text{high}}, J_{\text{low}} \] = highest, second highest, and lowest value among current cost functions
\[ J_{\text{new}}, J_{\text{new}} \] = generated new cost functions
\[ K_D, K_I, K, K \] = design parameters
\[ K_{\text{max}}, L, \alpha \] = constants for simulated annealing cooling schedule
\[ N_{\text{fail}}, N_{\text{MCE}} \] = number of failures
\[ n \] = number of simulations for Monte Carlo evaluation
\[ p \] = probability that no failure occurs
\[ P_{\text{init}}, p \] = actual probability of failure
\[ T \] = temperature (parameter for metropolis algorithm)
\[ V_G, \dot{Z} \] = ground speed and sink rate, m/s
\[ \beta_G \] = side-slip angle of inertial velocity vector, deg
\[ \| \Delta d \|_{\text{max}} \] = maximum Euclidean norm of two current design vectors
\[ \Delta J_{\text{max}} \] = maximum difference of two current cost functions
\[ \delta_i, y_i, \gamma_i \] = constants for stopping condition
\[ \lambda \] = constant for initial design vector
\[ \Phi, \Theta, \Psi \] = vehicle attitude, deg

Subscript

0 = initial value

I. Introduction

The development of spacecraft and aircraft, preflight evaluation of the flight control system is essential for confirming both flight safety and performance. Numerical simulation is an efficient tool for control system design and development because it allows many cases to be investigated within a short period of time. There are many uncertainties in the real flight, such as aerodynamics, actuator dynamics, sensor dynamics, environmental conditions, and inertial characteristics. The designed system must be robust against these uncertainties. Monte Carlo evaluation is a powerful tool for the evaluation of system robustness. Though the computational load is heavy, Monte Carlo evaluation gives quite rewarding results. It has been applied to the analysis of linear-time-invariant dynamic systems1–3 and to an Automatic Landing Flight Experiment (ALFLEX) conducted in 1996 (Ref. 4). In ALFLEX, the probability of mission achievement was estimated by Monte Carlo evaluation, and touchdown states of the experimental vehicle were estimated. The estimated distributions were consistent with the flight-test results.

The application of Monte Carlo evaluation was extended to flight control design in Refs. 5–9. The Monte Carlo result is directly fed back to control parameter tuning, and the stochastic model, which incorporates various uncertainties, is optimized (stochastic parameter optimization). The designed system is robust against the uncertainties. Genetic algorithms were applied to optimize control parameters, and Monte Carlo evaluation was used to derive cost functions.5–7 The mean tracking method was applied to an unpowered reentry vehicle, ALFLEX, in which the computational load for full flight simulation was quite heavy.8,9 The major contribution of the mean tracking method is that computational time is reduced in comparison to earlier methods. Though the mean tracking method is a local search technique, this approach extended the possibility of application in the real development of spacecraft. Parallel processing,9,10 in which many computers are used for the optimization, is also effective for the reduction of computational time.

In this paper, a simulated annealing algorithm is applied for stochastic parameter optimization. The objective of this approach...
is to find feasible control parameters that minimize the probability of violating design constraints within a reasonable time. Simulated annealing is a global search method that minimizes a cost function probabilistically. To improve search speed, the downhill-simplex method is combined with simulated annealing, as suggested in Ref. 11. Simulated annealing combined with the downhill-simplex method is different from the downhill-simplex method alone. In the reminder of this paper, reference to simulated annealing implies incorporation of the downhill-simplex method as well. The optimized result and computational time for simulated annealing are compared with those obtained by a genetic algorithm. The comparison verifies that simulated annealing reduces computational load while obtaining the same level of optimization as the genetic algorithm in a simple example.

Furthermore, the simulated annealing is applied to the ALFLEX longitudinal flight control system. Retaining the structure of the original control system, the presented algorithm improves the likelihood of satisfactory landing performance by revising the values of design parameters.

II. Stochastic Parameter Optimization

In stochastic parameter optimization, design parameters are tuned, whereas the cost function is derived by Monte Carlo evaluation, as shown in Fig. 1. The advantage of Monte Carlo evaluation is that the nonlinear system is evaluated directly, and the result is fed back to design parameter tuning. For a highly nonlinear system, the flight path is usually divided into several flight phases, and linear control theory is applied for each phase. As an alternative, the integrated nonlinear system can be optimized by stochastic parameter optimization. Another advantage is that various uncertainties can be incorporated as physical values, and so the result reflects the influence of the incorporated uncertainties. Design parameters are automatically tuned by an optimization algorithm such as the simulated annealing considered here.

In Monte Carlo evaluation, the distribution of each uncertainty is predetermined. Uncertain system parameters are generated depending on their assumed distributions. All generated uncertainties are incorporated into the nominal system model, and flight simulation is performed. The probability of mission success can be estimated by a number of simulations that are performed with different sets of uncertainties. System robustness against various uncertainties is statistically quantified. The cost function reflects the probability that all requirements will be satisfied. If any requirement is violated, the trial is considered unsatisfactory.

When all requirements are satisfied,

\[ z_i = 0 \]  

(1a)

Otherwise,

\[ z_i = 1 \]  

(1b)

When \( N_{MCE} \) simulations are performed, the cost function \( J \) to be minimized is determined as follows:

\[
J = \frac{\sum_{i=1}^{N_{MCE}} z_i}{N_{MCE}}
\]

(2)

In stochastic parameter optimization, the design parameter vector that minimizes the probability of unsatisfactory performance is determined. To obtain an accurate estimate of the cost function, \( N_{MCE} \) should be large. On the other hand, to reduce computational load, \( N_{MCE} \) should be small. It is important to choose an appropriate value of \( N_{MCE} \), and this is described in Sec. III.C.

III. Simulated Annealing

Simulated annealing is the numerical analog of the process by which a heated metal is cooled to become less brittle; gradual cooling allows atoms to align themselves in crystals that represent the minimum energy state of the material.

Generally, the search speed of simulated annealing is slow because a new design vector is generated randomly. In this paper, the downhill-simplex method is applied for the generation of a new design vector instead of random generation, and a structured search is added (generation). After a new design vector is generated, the corresponding new system is evaluated and a cost function is derived (evaluation). Then the current and new cost functions are compared, and one of them is accepted (comparison). The generation/evaluation/comparison procedure is described in Sec. III.B.

The manner in which cost functions are compared is the unique point of simulated annealing. The new cost \( J_{new} \) may be accepted even when \( J_{new} \) is worse (larger) than \( J_{cur} \) with a certain probability. On the other hand, in a local search technique \( J_{new} \) is accepted only when \( J_{new} \) is better (lower) than \( J_{cur} \). Because of this characteristic, the design vectors generated by simulated annealing can escape from local minima. The acceptance probability is determined by the difference of two cost functions, \( \exp((J_{cur} - J_{new})/T) \), and the parameter \( T \) that is called temperature. \( T \) is determined by the designer, and is decreased monotonically as the optimization proceeds. The scheduling of \( T \) (cooling schedule) is determined in Sec. III.A. The comparison algorithm is as follows:

If \( J_{new} \leq J_{cur} \)

\[ J_{cur} = J_{new} \]

Else if \( \exp((J_{cur} - J_{new})/T) > U(0, 1) \)

\[ J_{cur} = J_{new} \]

\( U(0, 1) \): random variable from a uniform distribution between 0 and 1

When a new cost is worse (\( J_{cur} < J_{new} \)), it is not likely to be accepted as \( (J_{cur} - J_{new}) \) and \( T \) decrease because the acceptance probability, \( \exp((J_{cur} - J_{new})/T) \), becomes small. This comparison is called the Metropolis algorithm.

A. Cooling Schedule

In simulated annealing, the numerical analog temperature \( T \) reduces as the search proceeds. The probability of accepting \( J_{new} (> J_{cur}) \) decreases as \( T \) decreases. For an effective search, the initial value of \( T_0 \) must be large enough to accept almost all generated design vectors, and \( T \) should decrease slowly. In this paper, the cooling schedule is chosen as follows:

\[ T(k) = T_0 \cdot (1 - k/K_{max})^w \]  

(4)

where \( k \) is the iteration index and \( K_{max}, w \), and \( T_0 \) are constants chosen by the designer. \( K_{max} \) determines the total number of iterations that drive \( T(k) \) to zero. If larger values of \( w \) or are used, more search iterations occur at lower temperature, meaning that increased values of \( J \) are less likely to be accepted. In the right term of Eq. (4), \( k \) may be fixed and renewed every \( L \) times. Once \( T \) reaches zero, Eq. (4) is disregarded, and only a lower value of \( J \) is accepted.

Acceptance ratio is determined as a ratio of the accepted number of generated higher cost functions to the total number of generated higher cost functions. When acceptance ratio is large enough, it means many higher cost functions are accepted and simulated annealing is working well. Otherwise, the cooling schedule should be reconsidered.
Simplex at the beginning

\[ J_{\text{init}} \]

a) Reflection

Away from the highest point

K1, K2: Design parameters

\( \bullet \): new design vector

\( \bigcirc \): current design vector

Contraction along one dimension from the highest point

b) Reflection and expansion

Contraction along all dimensions towards the lowest point
c) Contraction

d) Multiple contraction

Fig. 2 Downhill-simplex method (generation of new design vector).

Reflection: New design vector \( d_{\text{new}} \). Evaluation: \( J_{\text{new}} \).

Comparison: \( J_{\text{new}} \) and \( J_{\text{init}} \).

If \( J_{\text{new}} \) is not accepted, \( J_{\text{new}} = J_{\text{init}} \).

\( J_{\text{new}} \) is lower than \( J_{\text{init}} \)

Reflection and Expansion: \( d_{\text{new2}} \) is generated.

Evaluation: \( J_{\text{new2}} \).

Comparison: \( J_{\text{new2}} \) and \( J_{\text{new}} \).

\( J_{\text{new2}} \) is accepted.

\( J_{\text{new2}} \) is NOT accepted

Multiple Contraction

\( J_{\text{new2}} \) is NOT accepted

Multiple Contraction

New Simplex is created

Fig. 3 Procedure of generation/evaluation/comparison.

B. Generation, Evaluation, and Comparison

The downhill-simplex method finds the minimum of a function of \( n \) variables without calculating gradients.\(^{11,12}\) Given an \( n \)-dimensional design space, a simplex is formed by \( (n + 1) \) trial values of the design vector (Fig. 2), each associated with a different design cost. The algorithm searches for an \( (n + 1) \)nd design vector that yields lower cost than any of the points in the simplex using one of four operations: reflection, reflection and expansion, contraction, or multiple contraction, as illustrated by the example for \( n = 2 \) in Fig. 2. Because of this, a new design vector is efficiently generated by the downhill-simplex method compared to random generation.

The downhill-simplex procedure is incorporated into simulated annealing as shown in Fig. 3. The new \( (n + 1) \)nd design vector is generated by reflection, and the corresponding cost \( J_{\text{new}} \) is evaluated by Monte Carlo simulation. Then, \( J_{\text{new}} \) and \( J_{\text{high}} \) are compared by the Metropolis algorithm. When \( J_{\text{new}} \) is accepted, the design vector corresponding to \( J_{\text{high}} \) is replaced in the simplex by the new one. If \( J_{\text{new}} \) is not accepted, \( J_{\text{high}} \) is treated as \( J_{\text{new}} \) for one of the remaining procedures. \( J_{\text{new}} \) is compared with the current lowest cost \( J_{\text{low}} \). If \( J_{\text{new}} \) is better than \( J_{\text{low}} \), the next design vector is generated by reflection and expansion. The corresponding new cost function \( J_{\text{new2}} \) is evaluated and compared with \( J_{\text{low}} \) by the Metropolis algorithm. One of the two cost functions is accepted, and the corresponding design vector is retained. If \( J_{\text{new}} \) is worse than the second highest cost \( J_{\text{high}} \), the new design vector is generated by contraction. The corresponding new cost function \( J_{\text{new2}} \) is compared with \( J_{\text{new}} \). If \( J_{\text{new2}} \) is not accepted, multiple contraction is performed.

The difference between the simulated annealing and the downhill-simplex method is the comparison of two cost functions.

If lower cost function is always accepted in the procedure shown in Fig. 3, the optimization becomes the downhill-simplex method.

C. Number of Monte Carlo Evaluation

The number of numerical trials \( N_{\text{MCE}} \) should be increased as the optimization proceeds because the \( N_{\text{MCE}} \) required to estimate the probability at a given confidence interval increases as the actual probability becomes close to zero, as noted in Ref. 1. A guideline for the value of \( N_{\text{MCE}} \) is obtained from the binomial distribution.\(^{16}\) When \( N_{\text{MCE}} \) simulations are performed, the probability of detecting no failure \( P_{\text{no fail}} \) is obtained by

\[
P_{\text{no fail}} = N_{\text{MCE}} C_0 \cdot p^0 \cdot (1 - p)^{N_{\text{MCE}} - 1} = (1 - p)^{N_{\text{MCE}}}
\]

Therefore,

\[
N_{\text{MCE}} = \frac{\log P_{\text{no fail}}}{\log(1 - p)}
\]

Where \( p \) is an actual probability of failure. The value of \( N_{\text{MCE}} \) obtained by Eq. (6) is rounded up to the nearest integer. When the \( N_{\text{MCE}} \) derived by Eq. (6) is applied to Monte Carlo evaluation, the probability of occurrence of at least one failure is \( (1 - P_{\text{no fail}}) \).

When Eq. (6) is applied, the minimum value of cost function that appeared in the search \( J_{\text{min}} \) is used instead of \( p \) because the actual probability is unknown. The renewed value, \( N_{\text{MCE}}(k + 1) \), is given by

\[
N_{\text{MCE}}(k + 1) = \frac{\log P_{\text{no fail}}}{\log(1 - J_{\text{min}})}
\]
\( P_{\text{final}} \) is a constant determined by the designer. When \( J_{\text{max}} \) is zero, the minimum positive cost obtained by the current value of \( N_{\text{MCE}}(k) \) is used for \( J_{\text{min}} \):

\[
J_{\text{min}} = 1/N_{\text{MCE}}(k)
\]  

(8)

When \( N_{\text{MCE}} \) is increased, additional Monte Carlo evaluation is necessary for all current \((n+1)\) design vectors.

Equation (7) shows that \( N_{\text{MCE}} \) becomes large if a very small value of \( J_{\text{min}} \) is obtained. This means that a large number of simulations are necessary in the optimization when the optimized cost function is very good (small).

### D. Stopping Condition

When the cost functions are close enough to each other, the numerical search can be stopped. The stopping condition is determined both by the difference of the cost functions and by the distance of the design vectors, as follows,

\[
\Delta J_{\text{max}} \leq \gamma_1 \quad \|\Delta d\|_{\text{max}} \leq \delta_1 \cdot \|\Delta d\|_{\text{max}0}
\]  

(9)

where \( \gamma_1 \) and \( \delta_1 \) are constants determined by the designer.

The search also is stopped regardless of \( \|\Delta d\|_{\text{max}} \) when the cost function has a sufficiently low value. If Eq. (9) is not satisfied, the stopping condition can be softened to

\[
\Delta J_{\text{max}} = 0 \quad J \leq \gamma_2
\]  

(10)

where \( \gamma_2 \) is a constant determined by the designer.

Even when \( J > \gamma_2 \) and \( \|\Delta d\|_{\text{max}} > \delta_1 \cdot \|\Delta d\|_{\text{max}0} \), condition \( \Delta J_{\text{max}} = 0 \) sometimes occurs if \( N_{\text{MCE}} \) is not large enough to distinguish current cost functions. In this case, an appropriate new design vector is not generated because the gradient information of the current cost functions is unavailable. To manage this problem, \( N_{\text{MCE}} \) is increased by a small amount, and additional Monte Carlo evaluations are performed. First, the number of failures that gives the current value \( N_{\text{MCE}}(k) \) is derived. In Eq. (5), \( N_{\text{MCE}}(k) \) and \( N_{\text{fail}}/N_{\text{MCE}}(k) \) are replaced by \( N_{\text{MCE}} \) and \( p \), respectively:

\[
N_{\text{MCE}}(k) = \frac{\log P_{\text{final}}}{\log (1 - N_{\text{fail}}/N_{\text{MCE}}(k))}
\]  

\[
N_{\text{fail}} = N_{\text{MCE}}(k) \cdot \left[ 1 - P_{\text{final}} \left( 1/N_{\text{MCE}}(k) \right) \right]
\]  

(12)

When \( J_{\text{min}} < N_{\text{fail}}/N_{\text{MCE}}(k) \) in Eq. (7), \( N_{\text{MCE}} \) increases. To obtain the increased number of Monte Carlo evaluations, \( N_{\text{fail}} \) is rounded down to the nearest integer, and \( J_{\text{min}} = N_{\text{fail}}/N_{\text{MCE}}(k) \) is used in Eq. (7).

In addition to the stopping conditions described, the maximum number of simulations to be performed in the search can be determined for a stopping condition to avoid excessive calculation time.

### E. Initial Design Vector

When the design vector is a \((n+1)\) dimensional design vector is necessary to start the downhill-simplex procedure. In the real development of flight vehicles, a relatively good initial design vector \( d_0 \) often can be based on conventional engineering metrics and earlier experience. The remaining \( n \) initial design vectors, \( d_i, i = 1, \ldots, n \), are determined as follows:

\[
d_i = d_0 + (\lambda \cdot d_0(i)) \cdot e_i \quad i = 1, \ldots, n
\]  

(13)

where \( d_0(i) \) is an \( i \)th element of the vector \( d_0 \) and \( \lambda \) is a constant determined by the designer.

### F. Summary

The simulated annealing method is summarized in Fig. 4. First, \( d, T, \) and \( N_{\text{MCE}} \) are initialized. After initialization, the operations of generation, evaluation, and comparison are performed as shown in Fig. 3. The optimization procedure is terminated if the stopping condition is satisfied. Otherwise, \( T \) and \( N_{\text{MCE}} \) are renewed, and the process is returned to the generation.

Quite a few design vectors are generated in the optimization process, and every vector is assessed by Monte Carlo evaluation using the same set of uncertainties. In the optimization process, the same design vector is sometimes generated more than once. To avoid repeated evaluation of the same design vector, all generated design vectors and their corresponding results are stored in a buffer. If the algorithm selects a design vector that has been evaluated, the results are taken from the buffer, and Monte Carlo evaluation is skipped, reducing the computational burden.

### IV. Example of Simple Model

#### A. System Model

The algorithm is demonstrated by designing a proportional-integral-derivative (PID) controller for the simple model shown in Fig. 5. The model loosely represents simplified pitch attitude control of a flight vehicle. The uncertain parameter vector is \( x = [K_r \ a \ b]^T \), whose elements correspond to the loop gain, control delay time, static stability, and dynamic stability of the flight vehicle model. Normal distributions are assumed for these uncertainties, and their means and variances are given as follows:

\[
K : N(0, 3^2) \quad \tau : N(0.2, 0.05^2)
\]

\[
a : N(0, 0.5^2) \quad b : N(0, 0.3^2)
\]  

(14)

The variances shown in Eq. (14) correspond to the standard deviations of 3 dB of gain, 50 ms of delay time, 0.7 s\(^{-1}\) of natural frequency, and 0.2 of damping ratio.

The control design vector to be optimized contains the PID gains, \( d = [K_r \ K_p \ K_i \ K_d]^T \). In this example, the initial design vector is chosen as

\[
d_0 = [3.0 \ 3.0 \ 3.0 \ 3.0]^T
\]  

(15)

Requirements on the response to a unit step command are as follows: 1) overshoot \( \Delta y \leq 100\% \) of the input; 2) settling time \( |y - y_c| < 0.1, \) when \( t \geq 15 \) s; 3) tracking error

\[
\sum_{i=0}^{20} \left( \frac{y_i - y_c}{y_c} \right)^2 \cdot \Delta t < 3.0 \quad (\Delta t : \text{sampling time})
\]

and 4) control usage, \( |u| < 5.0 \). The cost function to be minimized is determined by Eq. (2) and is derived by Monte Carlo evaluation. Parameters for the optimization algorithm are shown in Table 1.
Table 1 Parameters for the optimization algorithm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooling schedule</td>
<td></td>
</tr>
<tr>
<td>$T_0$</td>
<td>10</td>
</tr>
<tr>
<td>$K_{\text{max}}$</td>
<td>100</td>
</tr>
<tr>
<td>$L$</td>
<td>5</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>2</td>
</tr>
<tr>
<td>Renewal of $N_{\text{MCCE}}$</td>
<td>100</td>
</tr>
<tr>
<td>$P_{\text{ref}}$</td>
<td>0.01</td>
</tr>
<tr>
<td>Stopping condition</td>
<td></td>
</tr>
<tr>
<td>$\delta_1$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.01</td>
</tr>
<tr>
<td>$P_2$</td>
<td>0.01</td>
</tr>
<tr>
<td>Initial design vector $\lambda$</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Fig. 6 Results of 10,000 Monte Carlo evaluations.

a) Results using initial design vector $d_0$

b) Results using optimized design vector $d^*$

Table 2 Comparison of three optimization methods

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Simulated annealing</th>
<th>Downhill-simplex</th>
<th>Genetic algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best design vector $d^*$</td>
<td>0.866</td>
<td>2.95</td>
<td>0.423</td>
</tr>
<tr>
<td>$K_f$</td>
<td>3.88</td>
<td>4.33</td>
<td>4.11</td>
</tr>
<tr>
<td>$K_r$</td>
<td>1.04</td>
<td>2.24</td>
<td>1.08</td>
</tr>
<tr>
<td>$K_D$</td>
<td>3.05</td>
<td>3.31</td>
<td>3.18</td>
</tr>
<tr>
<td>Total simulation number</td>
<td>31,998</td>
<td>13,604</td>
<td>121,532</td>
</tr>
<tr>
<td>Number of evaluated design vectors</td>
<td>66</td>
<td>51</td>
<td>745</td>
</tr>
</tbody>
</table>

Table 3 Results of 10,000 Monte Carlo evaluations using optimized design parameters

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost function $J$ [Confidence interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated annealing</td>
<td>0.0135 [0.012, 0.016]</td>
</tr>
<tr>
<td>Downhill-simplex</td>
<td>0.0278 [0.025, 0.031]</td>
</tr>
<tr>
<td>Genetic algorithm</td>
<td>0.0133 [0.012, 0.015]</td>
</tr>
</tbody>
</table>

It is also important to compare the computational time because stochastic optimization is computationally intensive. The cumulative numbers of simulations shown in Table 2 correspond to the computational load. Though the evaluation number required by the downhill-simplex method is the smallest, the optimized cost function is the worst. The second best is the number required by simulated annealing. In simulated annealing, 31,998 simulations were performed, and the number is about one-fourth of that required by the genetic algorithm. The difference arises from the nature of the algorithms.

In the genetic algorithm, many design vectors are generated simultaneously and randomly by the operations called selection, crossover, or mutation. Though good design vectors may be included, most of them are not as good, yet all of them must be evaluated. As a result, a large number of simulations are required. On the other hand, the combination of simulated annealing and the downhill-simplex methods generates just a single design vector at a time. Because of this, the number of generated design vectors in the search can be much reduced. Actually, the numbers of evaluated design vectors are much different, as shown in Table 2; in the genetic algorithms, 745 design vectors were evaluated, whereas, in simulated annealing, 66 design vectors were evaluated.

It is concluded that new design vectors are effectively generated in simulated annealing by the application of the downhill-simplex procedure. It has been demonstrated that simulated annealing requires much less computational time than that of genetic algorithm to obtain the same level of cost function.

V. Application to Unpiloted Flight Control System

Though simulated annealing is shown to be an efficient algorithm for stochastic optimization, it is important to show that the algorithm can be used in a practical application because the disadvantage of stochastic optimization is computational time. In this section, simulated annealing is applied to a real control system, the longitudinal autoland logic for the ALFLEX unpiloted reentry vehicle.

A. Unpiloted Flight Control System

ALFLEX was a Japanese flight experiment program to establish automatic landing technology for a future unpiloted reentry vehicle: it was successfully conducted at Woomera, South Australia in 1996 (Ref. 19). The ALFLEX vehicle and its major parameters are shown in Fig. 7. In the experiment, the vehicle was suspended from a helicopter and released 2700 m from the runway threshold at a height of 1500 m. The vehicle was guided to follow the predetermined reference path by its onboard computer. The flight path consisted of four phases: path capture phase, equilibrium gliding phase, preflare phase, and final flare phase. The details of the ALFLEX system and experiment are described in Refs. 12 and 19.

In the preflare phase, the open-loop command was predominant compared to the feedback command because the predetermined reference path was curved to reduce path angle before landing. The
system was less robust against various uncertainties; for this reason, the stochastic optimization algorithm is mainly applied to the longitudinal guidance of the preflare phase to obtain a robust system.

B. Control Parameter Optimization

The longitudinal guidance parameters are optimized to improve the probability of mission achievement. The preflare guidance and control system is shown in Fig. 8. In this system, five parameters are tuned by the optimization algorithm, namely, three PID gains in the guidance feedback control ($K_p$, $K_i$, $K_d$), one delay compensation time ($\Delta T$) in the open-loop command generation, and one sink-rate feedback gain $K_{FF}$ of final-flare phase. The number of uncertainties is more than 100, as shown in Table 4. Among them, influential uncertainties are shown in Ref. 8. The requirements for touchdown state errors are shown in Table 5.

The control parameters were carefully tuned by engineers before the experiment, and these values are used as initial values for the current optimization. The vehicle motion from release to landing is simulated repeatedly in the Monte Carlo evaluation. The optimization is performed using a parallel processing system in which 16 CPUs (Sun Microsystems UltraSPARC-2) are connected to reduce computational time.

As a result of the numerical search, 80 design vectors are evaluated. The total number of simulations is 25,228, with a calculation time of about 9 h. The initial and optimized design vectors are shown in Table 6. The proportional gain $K_p$ is almost double the initial value, and this is the most effective parameter. The derivative gain $K_d$ is increased by 18%, and this is the second most effective parameter. System robustness is improved by increasing feedback gains. The values of integral gain $K_i$ are very small both for the initial and the optimized system, and this means that this gain is unnecessary in this system. The other parameters are not changed much, indicating that the initial values were appropriately chosen.

Table 4 Uncertain parameters for ALFLEX model

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass parameters</td>
<td>5</td>
</tr>
<tr>
<td>Aerodynamics</td>
<td>27</td>
</tr>
<tr>
<td>Actuator dynamics</td>
<td>9</td>
</tr>
<tr>
<td>Sensor dynamics and error</td>
<td>38</td>
</tr>
<tr>
<td>Atmospheric condition</td>
<td>6</td>
</tr>
<tr>
<td>Initial condition and error at release</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 5 Requirements of ALFLEX touchdown performance

<table>
<thead>
<tr>
<th>Touchdown states</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position, m</td>
<td>$X &gt; 0,</td>
</tr>
<tr>
<td>Velocity, m/s</td>
<td>$V_G &lt; 62, Z &lt; 3$</td>
</tr>
<tr>
<td>Attitude, deg</td>
<td>$\Theta &lt; 23, \Psi &lt; 10,</td>
</tr>
<tr>
<td>Side slip, deg</td>
<td>$</td>
</tr>
</tbody>
</table>

*Runway coordinate; the origin is at the runway threshold, the $X$ axis is directed along the runway centerline, and the $Z$ axis is directed downward.

Table 6 Design parameters for modified ALFLEX guidance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
<th>$K_{FF}$</th>
<th>$\Delta T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_0$</td>
<td>0.0106</td>
<td>0.000700</td>
<td>0.0661</td>
<td>0.110</td>
<td>1.00</td>
</tr>
<tr>
<td>$d^*$</td>
<td>0.0206</td>
<td>-0.000211</td>
<td>0.0780</td>
<td>0.107</td>
<td>1.02</td>
</tr>
</tbody>
</table>

Fig. 7 Three views of ALFLEX vehicle.

Fig. 8 Simple block diagram of ALFLEX longitudinal guidance and control.
Table 7  Number of cases exceeding each requirement

<table>
<thead>
<tr>
<th>Parameter</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Vg</th>
<th>Θ</th>
<th>Φ</th>
<th>ψ</th>
<th>βg</th>
<th>Unstable</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>d0</td>
<td>77</td>
<td>2</td>
<td>453</td>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>568</td>
</tr>
<tr>
<td>d*</td>
<td>0</td>
<td>0</td>
<td>64</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>33</td>
<td>108</td>
</tr>
</tbody>
</table>

In Table 7, the number of cases exceeding each requirement that corresponds to Fig. 9 is shown. Unstable in Table 7 means that the vehicle becomes unstable and cannot reach the runway. The total number of unsatisfactory cases is shown in the last column, and its value is different from a simple sum of requirement violation numbers shown in the other columns because some cases exceed more than one requirement. From the result, the probability of failure for the optimized system shown in Fig. 9b is 1.1%, whereas the one for the initial system shown in Fig. 9a is 5.7%. The system robustness is improved by the stochastic optimization, and it is demonstrated that simulated annealing is effective in a practical application.

VI. Conclusions

Design parameters of a nonlinear flight control system are optimized probabilistically using simulated annealing. Each design vector is assessed by Monte Carlo evaluation to derive a cost function, and the cost function is minimized numerically. For the simple example considered here, the computational time required for simulated annealing was much less than that required by the genetic algorithm to obtain the same level of cost function, and both methods produced better results than the downhill–simplex method. Application of simulated annealing to redesign of the AFLEX flight control reduced the likelihood of unsatisfactory stability and performance from 5.7 to 1.1%, a significant improvement. It is concluded that the combination of Monte Carlo evaluation and simulated annealing provides a powerful new method of designing robust flight control systems.

Acknowledgments

This research has been supported by the Federal Aviation Administration (FAA) and NASA under FAA Grant 95-G-0011 and by the Science and Technology Agency of Japan.

References


