JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 25, No. 2, March–April 2002

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 unpiloted 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, τ	=	uncertain system parameters
$\boldsymbol{d}$	=	design parameter vector
$\boldsymbol{d}^*$	=	optimized design parameter vector
$\boldsymbol{e}_i$	=	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_{cur}	=	one of the current cost functions
$J_{\mathrm{high}}, J_{\mathrm{shigh}}, J_{\mathrm{low}}$	=	highest, second highest, and lowest value among current cost functions
$J_{\mathrm{new}}, J_{\mathrm{new2}}$	=	generated new cost functions
$K_D, K_f, K_I, K_P, K_{Pf}, \Delta T$	=	design parameters
$K_{\max}, L, \alpha$	=	constants for simulated annealing cooling schedule
N_{fail}	=	number of failures
N_{MCE}	=	number of simulations for Monte Carlo evaluation
n	=	number of design parameters
P_{nofail}	=	probability that no failure occurs
p	=	actual probability of failure
T	=	temperature (parameter for metropolis algorithm)
$V_G, \dot{Z}$	=	ground speed and sink rate, m/s
β_G	=	side-slip angle of inertial velocity vector, deg
$\|\Delta \boldsymbol{d}\|_{\max}$	=	maximum Euclidean norm of two current design vectors
$\Delta J_{\max}$	=	maximum difference of two current cost functions
$\delta_1, \gamma_1, \gamma_2$	=	constants for stopping condition
λ	=	constant for initial design vector
Φ, Θ, Ψ	=	vehicle attitude, deg

Subscript

0	=	initial value

I. Introduction

In 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 systems[1–3] 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 unpiloted 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

Presented as Paper 2000-4557 at the Guidance, Navigation, and Control Conference, Denver, CO, 14–17 August 2000; received 15 September 2000; revision received 5 November 2001; accepted for publication 9 November 2001.

*Associate Senior Engineer, Office of Space Transportation Systems. Member AIAA.

†Professor, Department of Mechanical and Aerospace Engineering. Fellow AIAA.

‡Group Leader, Flight Systems Research Center. Member AIAA.

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 compared to the downhill–simplex method alone. In the remainder 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.[12,13] 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 influences 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,

$$z_i = 0 \tag{1a}$$

Otherwise,

$$z_i = 1 \tag{1b}$$

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

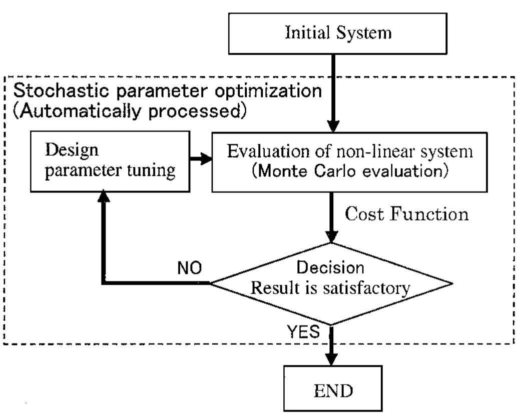


Fig. 1 Procedure of stochastic parameter optimization.

$$J = \frac{\sum_{i=1}^{N_{\mathrm{MCE}}} z_i}{N_{\mathrm{MCE}}} \tag{2}$$

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.[14]

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, $(J_{\mathrm{cur}} - J_{\mathrm{new}})$, 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:

$$\begin{aligned}
&\text{If} \quad J_{\mathrm{new}} \le J_{\mathrm{cur}} \\
&\qquad J_{\mathrm{cur}} = J_{\mathrm{new}} \\
&\text{Else if } \exp[(J_{\mathrm{cur}} - J_{\mathrm{new}})/T] > U(0, 1) \\
&\qquad J_{\mathrm{cur}} = J_{\mathrm{new}}
\end{aligned} \tag{3}$$

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

When a new cost is worse ($J_{\mathrm{cur}} < J_{\mathrm{new}}$), it is not likely to be accepted as $(J_{\mathrm{cur}} - J_{\mathrm{new}})$ and T decrease because the acceptance probability, $\exp\{(J_{\mathrm{cur}} - J_{\mathrm{new}})/T\}$, becomes small. This comparison is called the Metropolis algorithm.[14]

A. Cooling Schedule

In simulated annealing, the numerical analog temperature T reduces as the search proceeds. The probability of accepting $J_{\mathrm{new}}(> J_{\mathrm{cur}})$ 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[11]:

$$T(k) = T_0 \cdot (1 - k/K_{\max})^{\alpha} \tag{4}$$

where k is the iteration index and $K_{\max}$, α, 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 α 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.

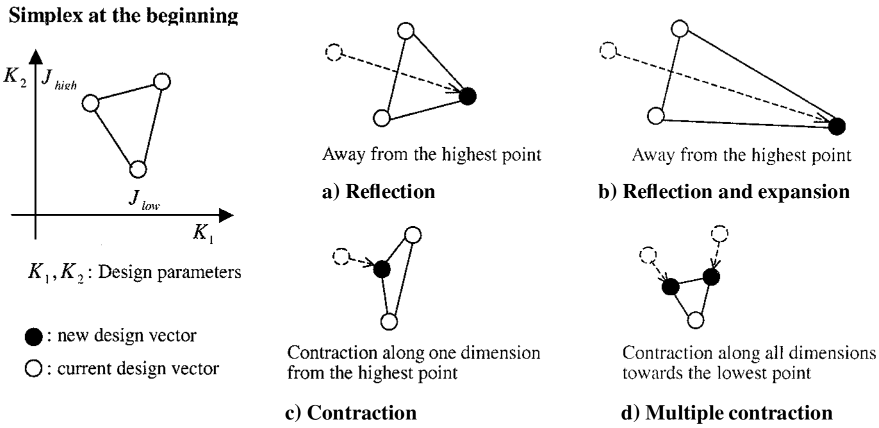


Fig. 2 Downhill-simplex method (generation of new design vector).

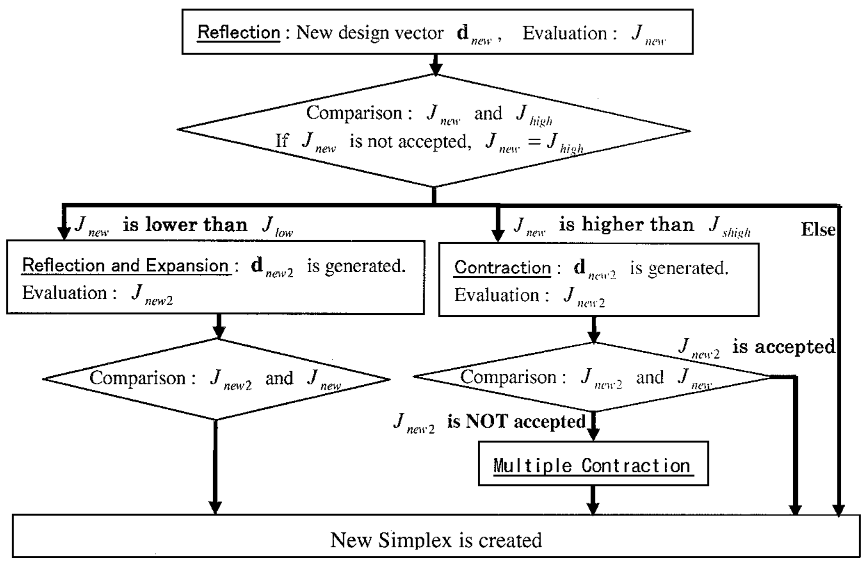


Fig. 3 Procedure of generation/evaluation/comparison.

B. Generation, Evaluation, and Comparison

The downhill–simplex method finds the minimum of a function of n variables without calculating gradients.[11,15] 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 $(n+2)$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 $(n+2)$nd design vector is generated by reflection, and the corresponding cost J_{new} is evaluated by Monte Carlo simulation. Then, J_{new} and J_{high} are compared by the Metropolis algorithm. When J_{new} is accepted, the design vector corresponding to J_{high} is replaced in the simplex by the new one. If J_{new} is not accepted, J_{high} is treated as J_{new} for one of the remaining procedures. J_{new} is compared with the current lowest cost J_{low}. If J_{new} is better than J_{low}, the next design vector is generated by reflection and expansion. The corresponding new cost function J_{new2} is evaluated and compared with J_{new} by the Metropolis algorithm. One of the two cost functions is accepted, and the corresponding design vector is retained. If J_{new} is worse than the second highest cost J_{shigh} the new design vector is generated by contraction. The corresponding new cost function J_{new2} is compared with J_{new}. If J_{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_{MCE} should be increased as the optimization proceeds because the N_{MCE} required to estimate 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_{MCE} is obtained from the binomial distribution.[16] When N_{MCE} simulations are performed, the probability of detecting no failure P_{nofail} is obtained by

$$P_{nofail} = {}_{N_{MCE}}C_0 \cdot p^0 \cdot (1-p)^{N_{MCE}-0} = (1-p)^{N_{MCE}} \tag{5}$$

Therefore,

$$N_{MCE} = \frac{\log P_{nofail}}{\log(1-p)} \tag{6}$$

Where p is an actual probability of failure. The value of N_{MCE} obtained by Eq. (6) is rounded up to the nearest integer. When the N_{MCE} derived by Eq. (6) is applied to Monte Carlo evaluation, the probability of occurrence of at least one failure is $(1 - P_{nofail})$.

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

$$N_{MCE}(k+1) = \frac{\log P_{nofail}}{\log(1 - J_{min})} \tag{7}$$

P_{nofail} is a constant determined by the designer. When $J_{\min}$ is zero, the minimum positive cost obtained by the current value of $N_{\mathrm{MCE}}(k)$ is used for $J_{\min}$:

$$J_{\min} = 1/N_{\mathrm{MCE}}(k) \tag{8}$$

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

Equation (7) shows that N_{MCE} becomes large if a very small value of $J_{\min}$ 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,

$$\Delta J_{\max} \le \gamma_1 \qquad \|\Delta \boldsymbol{d}\|_{\max} \le \delta_1 \cdot \|\Delta \boldsymbol{d}\|_{\max 0} \tag{9}$$

where γ_1 and δ_1 are constants determined by the designer.

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

$$\Delta J_{\max} = 0 \qquad J \le \gamma_2 \tag{10}$$

where γ_2 is a constant determined by the designer.

Even when $J > \gamma_2$ and $\|\Delta \boldsymbol{d}\|_{\max} > \delta_1 \cdot \|\Delta \boldsymbol{d}\|_{\max 0}$, condition $\Delta J_{\max} = 0$ sometimes occurs if N_{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_{MCE} is increased by a small amount, and additional Monte Carlo evaluations are performed. First, the number of failures that gives the current value $N_{\mathrm{MCE}}(k)$ is derived. In Eq. (5), $N_{\mathrm{MCE}}(k)$ and $N_{\mathrm{fail}}/N_{\mathrm{MCE}}(k)$ are replaced by N_{MCE} and p, respectively:

$$N_{\mathrm{MCE}}(k) = \frac{\log P_{\mathrm{nofail}}}{\log\{1 - N_{\mathrm{fail}}/N_{\mathrm{MCE}}(k)\}} \tag{11}$$

$$\therefore N_{\mathrm{fail}} = N_{\mathrm{MCE}}(k) \cdot \left\{1 - P_{\mathrm{nofail}}{}^{(1/N_{\mathrm{MCE}}(k))}\right\} \tag{12}$$

When $J_{\min} < N_{\mathrm{fail}}/N_{\mathrm{MCE}}(k)$ in Eq. (7), N_{MCE} increases. To obtain the increased number of Monte Carlo evaluations, N_{fail} is rounded down to the nearest integer, and $J_{\min} = N_{\mathrm{fail}}/N_{\mathrm{MCE}}(k)$ 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 n dimensional, $(n+1)$ design vectors are necessary to start the downhill–simplex procedure. In the real development of flight vehicles, a relatively good initial design vector $\boldsymbol{d}_0$ often can be based on conventional engineering metrics and earlier experience. The remaining n initial design vectors, $\boldsymbol{d}_i$, $i = 1, \ldots, n$, are determined as follows:

$$\boldsymbol{d}_i = \boldsymbol{d}_0 + \{\lambda \cdot \boldsymbol{d}_0(i)\} \cdot \boldsymbol{e}_i \qquad i = 1, \ldots, n \tag{13}$$

where $\boldsymbol{d}_0(i)$ is an ith element of the vector $\boldsymbol{d}_0$ and λ is a constant determined by the designer.

F. Summary

The simulated annealing method is summarized in Fig. 4. First, $\boldsymbol{d}$, T, and N_{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_{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 to 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.

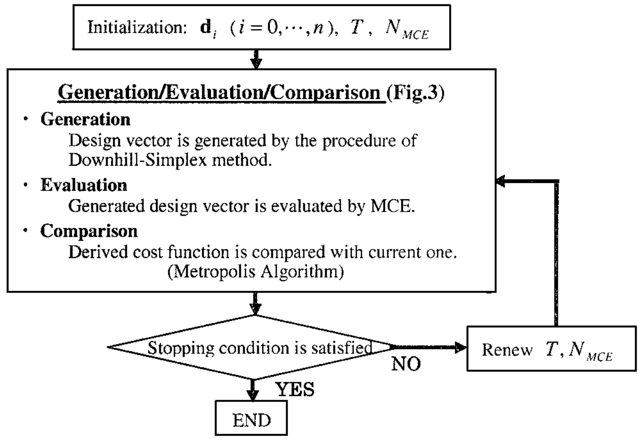


Fig. 4 Optimization procedure by simulated annealing.

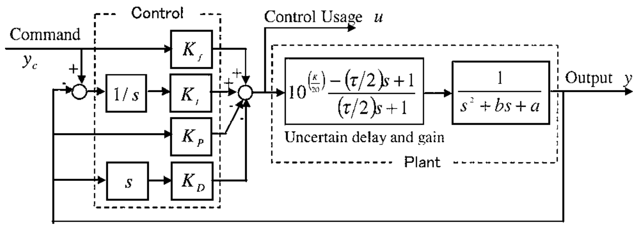


Fig. 5 Simplified model of pitch attitude control.

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 $\boldsymbol{x} = [K \;\; \tau \;\; a \;\; b]^T$, 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:

$$K : N(0, 3^2) \qquad \tau : N(0.2, 0.05^2)$$
$$a : N(0, 0.5^2) \qquad b : N(0, 0.3^2) \tag{14}$$

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, $\boldsymbol{d} = [K_f \;\; K_P \;\; K_I \;\; K_D]^T$. In this example, the initial design vector is chosen as

$$\boldsymbol{d}_0 = [3.0 \quad 3.0 \quad 3.0 \quad 3.0]^T \tag{15}$$

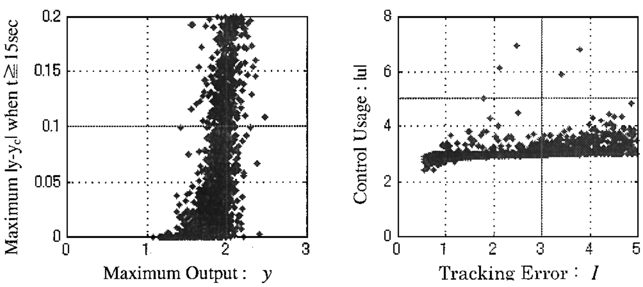
Requirements on the response to a unit step command are as follows: 1) overshoot $\Delta y \le 100\%$ of the input; 2) settling time $|y - 1.0| < 0.1$, when $t \ge 15$ s; 3) tracking error

$$\sum_{t=0}^{20} \left(\frac{y - y_c}{y_c}\right)^2 \cdot \Delta t < 3.0 \qquad (\Delta t : \text{sampling time})$$

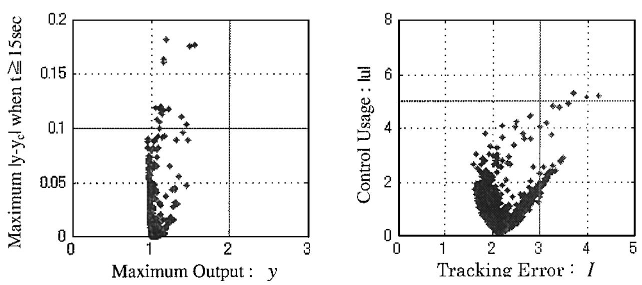
and 4) control usage, $|u| < 5.0$. The cost function J to be minimized is determined by Eq. (2) and is derived by Monte Carlo evaluation. Parameters for the optimization algorithm are shown in Table 1.

Table 1 Parameters for the optimization algorithm

Parameter	Value
Cooling schedule	
T_0	10
$K_{\max}$	100
L	5
α	2
Renewal of N_{MCE}	
N_{MCE0}	100
P_{nofail}	0.01
Stopping condition	
δ_1	0.1
γ_1	0.01
γ_2	0.01
Initial design vector	
λ	1.0



a) Results using initial design vector d_0



b) Results using optimized design vector d^*

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

B. Result of Stochastic Optimization

The design vector was optimized by simulated annealing. Figure 6 shows 10,000 Monte Carlo evaluations both for $\boldsymbol{d}_0$ and for $\boldsymbol{d}^*$. Solid lines in the graphs are the requirements just given in Sec. IV.A. The number of cases exceeding each requirement is much larger in Fig. 6a than that in Fig. 6b. Because the cost function for $\boldsymbol{d}_0$ is 0.14, whereas the cost function for $\boldsymbol{d}^*$ is 0.0135, the probability of unsatisfactory performance is much reduced.

The results of simulated annealing are compared with those obtained both by the downhill–simplex method and by the genetic algorithm. There are many genetic algorithm implementations[17]; the real-number genetic algorithm described in Ref. 5 is used for the comparison. It is important to compare not only the optimized cost but also the computational time.

The optimized design vector, cumulative number of simulations and number of evaluated design vectors in each optimization method are shown in Table 2. Each design vector is assessed by 10,000 Monte Carlo evaluations, and the cost functions and their confidence intervals[18] are compared in Table 3. The costs obtained by simulated annealing and the genetic algorithm are better than that obtained by the downhill–simplex method. The differences are significant because the confidence intervals do not overlap. However, the difference in the results between the simulated annealing and the genetic algorithm is quite small. Actually, the design vectors obtained by simulated annealing and by the genetic algorithm shown in Table 2 are quite similar, whereas that obtained by the downhill–simplex method is very different.

Table 2 Comparison of three optimization methods

Parameter	Simulated annealing	Downhill–simplex	Genetic algorithm
Best design vector $\boldsymbol{d}^*$			
K_f	0.866	2.95	0.423
K_P	3.88	4.33	4.11
K_I	1.04	2.24	1.08
K_D	3.05	3.31	3.18
Total simulation number	31,998	13,604	121,552
Number of evaluated design vectors	66	51	745

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

Method	Cost function J	[Confidence interval]
Simulated annealing	0.0135	[0.012, 0.016]
Downhill–simplex	0.0278	[0.025, 0.031]
Genetic algorithm	0.0133	[0.012, 0.015]

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 (ΔT) in the open-loop command generation, and one sink-rate feedback gain K_{Pf} 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

Category	Number of parameters
Mass parameters	5
Aerodynamics	27
Actuator dynamics	9
Sensor dynamics and error	38
Atmospheric condition	6
Initial condition and error at release	18

Table 5 Requirements of ALFLEX touchdown performance

Touchdown states	Requirement
Position,[a] m	$X > 0$, $\lvert Y\rvert < 18$
Velocity, m/s	$V_G < 62$, $\dot{Z} < 3$
Attitude, deg	$\Theta < 23$, $\lvert\Phi\rvert < 10$, $\lvert\Psi\rvert < 8$
Side slip, deg	$\lvert\beta_G\rvert < 8$

[a]Runway coordination; 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

Parameter	K_P	K_I	K_D	K_{Pf}	ΔT
d_0	0.0106	0.000700	0.0661	0.110	1.00
d^*	0.0206	−0.000211	0.0780	0.107	1.02

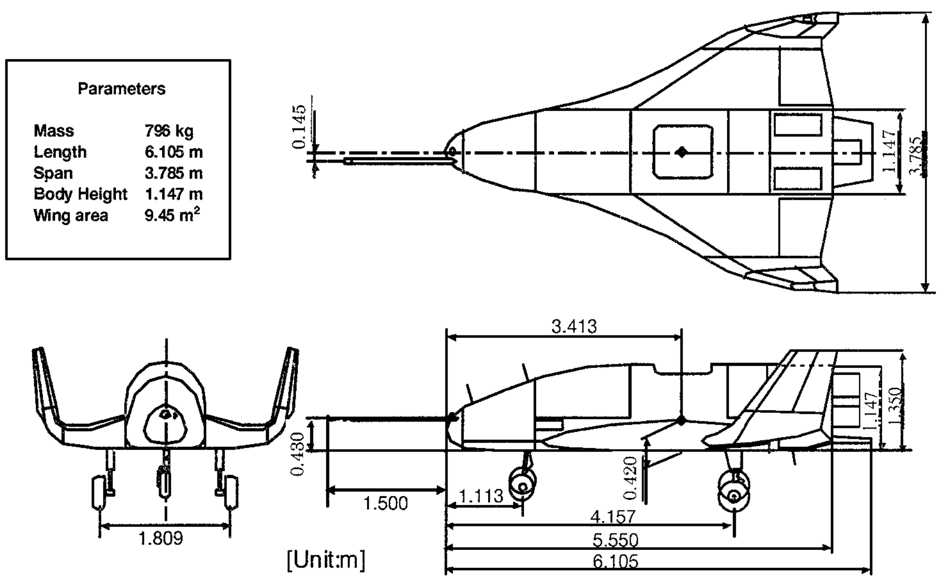


Fig. 7 Three views of ALFLEX vehicle.

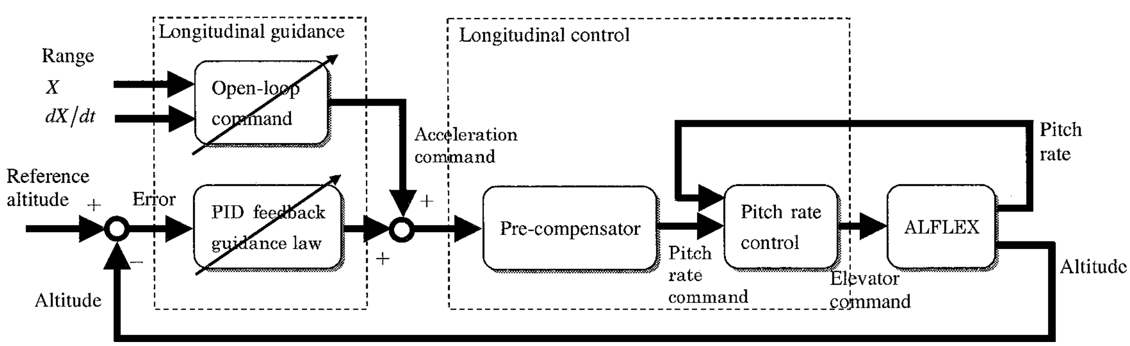
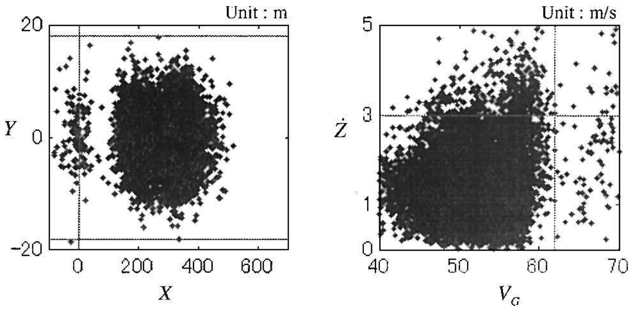


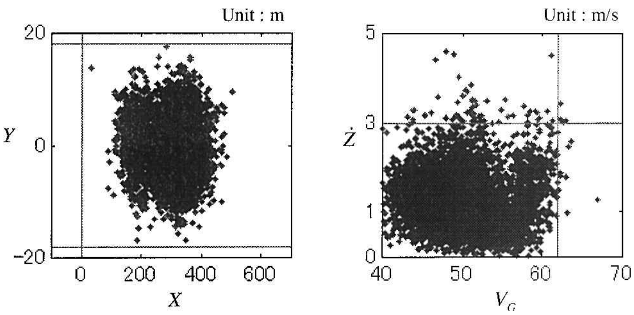
Fig. 8 Simple block diagram of ALFLEX longitudinal guidance and control.

Table 7 Number of cases exceeding each requirement

Parameter	X	Y	$\dot{Z}$	V_G	Θ	Φ	Ψ	β_G	Unstable	Total
$\boldsymbol{d}_0$	77	2	453	120	0	0	9	0	33	568
$\boldsymbol{d}^*$	0	0	64	12	1	0	4	0	33	108



a) Results using initial control parameter d_0



b) Results using optimized control parameter d^*

Fig. 9 Touchdown position and velocity of 10,000 Monte Carlo evaluations.

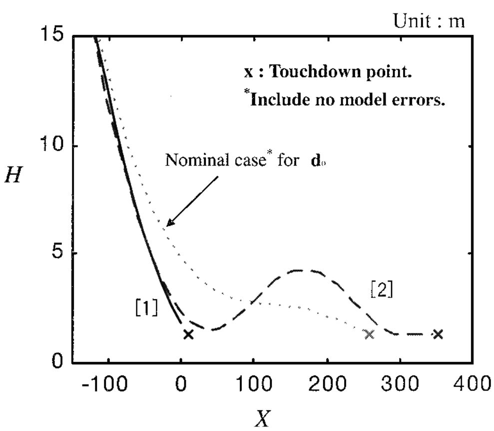


Fig. 10 Example of flight path near runway.

Monte Carlo results of 10,000 simulations are shown in Fig. 9 both for the initial design vector and for the optimized one. Solid lines in the graphs show the requirements. The touchdown distributions of the position, X and Y, and the velocity, sink rate $\dot{Z}$, and ground speed V_G are shown. In Fig. 9a two clusters appear, and there are a few cases around $X = 100$ m. The touchdown position depends on the guidance error between the vehicle state and the reference path calculated onboard using navigation data. When the guidance error is significant in the beginning of the final-flare phase, the vehicle lands around the runway threshold, shown by the solid line 1, in Fig. 10, and the touchdown ground speed becomes large. On the other hand, when the guidance error is large but not significant, the vehicle tends to float because of the sink-rate feedback in the final-flare phase. The vehicle lands further down the runway, shown by the broken line 2 in Fig. 10.

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 can not 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 ALFLEX 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

[1] Stengel, R. F., and Ray, L. R., "Stochastic Robustness of Linear-Time-Invariant Control Systems," *IEEE Transactions on Automatic Control*, Vol. 36, No. 1, 1991, pp. 82–87.

[2] Ray, L. R., and Stengel, R. F., "Application of Stochastic Robustness to Aircraft Control Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 6, 1991, pp. 1251–1259.

[3] Ray, L. R., and Stengel, R. F., "A Monte Carlo Approach to the Analysis of Control System Robustness," *Automatica*, Vol. 29, No. 1, 1993, pp. 229–236.

[4] Motoda, T., Miyazawa, Y., Ishikawa, K., and Izumi, T., "Automatic Landing Flight Experiment Flight Simulation Analysis and Flight Testing," *Journal of Spacecraft and Rockets*, Vol. 36, No. 4, 1999, pp. 554–560.

[5] Marrison, C. I., and Stengel, R. F., "Robust Control System Design Using Random Search and Genetic Algorithms," *IEEE Transactions on Automatic Control*, Vol. 42, No. 6, 1997, pp. 835–839.

[6] Marrison, C. I., and Stengel, R. F., "Design of Robust Control Systems for Hypersonic Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 58–63.

[7] Stengel, R. F., and Wang, Q., "Searching for Robust Minimal-Order Compensators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 123, No. 2, 2001, pp. 233–236.

[8] Miyazawa, Y., Motoda, T., Izumi, T., and Hata, T., "Longitudinal Landing Control Law for an Autonomous Reentry Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 6, 1999, pp. 791–800.

[9] Miyazawa, Y., and Motoda, T., "Stochastic Parameter Tuning Applied to Space Vehicle Flight Control Design," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001, pp. 597–604.

[10] Schubert, W. M., and Stengel, R. F., "Parallel Synthesis of Robust Control Systems," *IEEE Transactions on Control Systems Technology*, Vol. 6, No. 6, 1998, pp. 701–706.

[11] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C*, 2nd ed., Cambridge Univ. Press, New York, 1993, pp. 408–412, 444–455.

[12]Miyazawa, Y., Ishikawa, K., Motoda, T., Izumi, T., Sagisaka, M., Hata, T., and Onuma, H., "Flight Control System for the Automatic Landing Flight Experiment," AIAA Paper 96-3782, July 1996.

[13]Mooji, E., *Linear Quadratic Regulator Design for an Unpowered, Winged Re-entry Vehicle*, Delft Univ. Press, Delft, The Netherlands, 1998, pp. 5–16.

[14]Aarts, E., and Korst, J., *Simulated Annealing and Boltzmann Machines*, Wiley, New York, 1989, pp. 13–31.

[15]Anderson, L. R., "Fine Tuning of Aircraft Control Laws Using Pro-MATLAB Software," AIAA Paper 91-2600-CP, Aug. 1991.

[16]Shakarian, A., "Application of Monte Carlo Techniques to the 757/767 Autoland Dispersion Analysis by Simulation," AIAA Paper 83-2193, Aug. 1983.

[17]Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991, pp. 1–70.

[18]Conover, W. J., *Practical Nonparametric Statistics*, 3rd ed., Wiley, New York, 1999, pp. 131–133.

[19]Miyazawa, Y., Nagayasu, M., and Nakayasu, H., "Flight Testing of ALFLEX Guidance, Navigation, and Control System," International Council of the Aeronautical Sciences Paper ICAS-98-1.1.3, Sept. 1998.