ROBUST OPTIMAL DESIGN OF AN AUTOPILOT FOR TRANSITIONAL FLIGHTS OF A TAIL-SITTER MINI UAV

Daisuke Kubo*, Shinji Suzuki**
***Japan Aerospace Exploration Agency, **The University of Tokyo**

Keywords: *robust optimization, genetic algorithm, tail-sitter, unmanned aerial vehicle*

Abstract

The transitional flight of tail-sitter aircraft is a challenging problem because the flight includes a wide operating range with nonlinearity and at low speed approaches the stall. In order to solve this problem, a robust autopilot design method using a variable environment genetic algorithm (VE-GA) is proposed. VE-GA is a new robust optimization method based on a real coded genetic algorithm (RCGA). Here, the word "environment" refers to the uncertainties considered in the evaluation functions. In a VE-GA, the environment is changed repeatedly after several generations. In this manner, genes go through many types of environments over generations, and obtain robustness against uncertainties. In order to improve the efficiency and accuracy of the optimization, we introduce a local optimization method—Powell's direction set method (PDSM), and term the combined robust optimization procedure VE-GA/PDSM. Finally, our proposed method is applied to the offline-based parameter optimization of a neural network (NN) which is part of a tail-sitter mini unmanned aerial vehicle's (UAV) autopilot architecture.

1 Introduction

For a model-based controller design approach, it is essential to prepare an accurate model of the actual plant. If the model is sufficiently accurate, many kinds of controller design methods can be applied, but an accurate model is often unavailable in real implementations and so it is important to design controllers with high robustness against model uncertainties. However, designing such robust nonlinear controllers is not an easy problem, because robust optimization usually requires much greater calculation than conventional deterministic optimization. In previous research, the authors proposed robust controller design using a new robust optimization method—a variable environment genetic algorithm (VE-GA) [1]. Here, "environment" refers to the model errors and disturbance settings assumed in the offline simulations used to optimize the controller parameters. During the evolutionary iteration of VE-GA, the environment is systematically changed after a number of generations. This way, genes go through many types of environments and achieve moderate fitness for all environments; the population becomes more robust.

However, the VE-GA has the following two problems: low efficiency of the final phase because it is based only on a probabilistic approach, and a lack of accuracy of constraint evaluations because the weight factor of the penalty functions cannot be increased in order to maintain the algorithm's stability. In this paper, we will reformulate the VE-GA, especially the evaluation (fitness) functions, and introduce a local optimization method—Powell's direction set method (PDSM)—to solve the problems of the previous VE-GA. The new robust optimization procedure (VE-GA/PDSM) is applied to the design problem of an autopilot for the transitional flight of a tail-sitter mini unmanned aerial vehicle (UAV).

2 Tail-Sitter Mini UAV

Mini UAVs are very small, single-person portable unmanned aircraft which are useful in many applications in a variety of fields, such as environmental observation, law enforcement and disaster mitigation [2–4]. However, in spite of the potential of mini UAVs, there are a number of problems in their operation. One problem is takeoff and landing. Although mini UAVs do not require runways and can be operated from relatively compact areas such as athletic grounds or football fields, it can still be difficult to find such locations for takeoff and landing in practical operations. In fact, the authors experienced having to abandon a forest observation mission using mini UAVs because no suitable takeoff and landing point could be found in the deep forest.

Common methods to improve the landing performance of mini UAVs are parachutes [2] and deep-stall decent technique. However, these methods also have disadvantages, such as low accuracy or impact shock at touchdown. Another option that also improves takeoff performance is the vertical takeoff and landing (VTOL) approach [5]. One of the simplest VTOL mechanisms is tail-sitter. A tail-sitter will takeoff and land on its tail section with its fuselage and nose pointing upward. Tail-sitters have the advantage of not requiring variable mechanisms to transition between hover and cruise, and this configuration is therefore particularly appropriate for "mini" UAVs that have a strict weight constraint because of their small size.

The authors proposed a new design for such a tail-sitter mini UAV in previous research [6,7] (Fig. 1). The vehicle can cruise efficiently like a conventional fixed wing airplane (Fig. 1 left) and hover like a helicopter (Fig. 1 right).

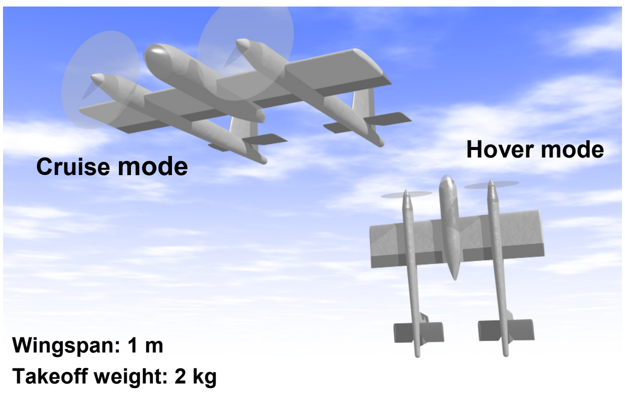

Fig.1 The proposed tail-sitter mini UAV

Transition flight between cruise and hover is a challenging problem for the tail-sitter because the transition covers a wide operating ranges with nonlinearity and tends to approach the stall, an uncontrollable condition in low speed flight. Additionally, the sensor and on-board processing performance of mini UAVs is limited because of their size, weight, and cost constraints. Therefore, simple controllers for a nonlinear system are required, with robustness being very important.

2.1 Design Features of the Vehicle

Although other experimental tail-sitter VTOL UAVs exist [8–10], our proposed design has the following distinguishing features:

1. Twin contra-rotating propellers. This has the advantage of being much simpler than other configurations that eliminate engine torque effects such as coaxial contra-rotating propellers/rotors.
2. The ailerons, rudders, and elevators are immersed in the propeller slipstream and so are effective for attitude control even in low-speed flight. No other complex control devices are required.
3. Apart from the control surfaces, there are no variable mechanisms such as the tilt mechanisms of tilt-rotors, tilt-wings, or tilt-ducts that would complicate the system.

2.2 Operation Scenario

An assumed operation scenario for the tail-sitter mini UAV is illustrated in Fig. 2. In the takeoff phase, the vehicle is launched by hand or by support equipment and climbs vertically to a certain altitude. The vehicle then increases its flight speed and transitions to forward wing-borne flight; this is called an outbound transition. After completing its mission, the vehicle approaches the landing point. It decreases its flight speed and transitions to the hovering mode; this is called an inbound transition. During the final landing phase, the vehicle descends vertically and touches down on its tail gear, then drops forward to touch down on its main gear, and comes to rest supported by both the tail and main gear.

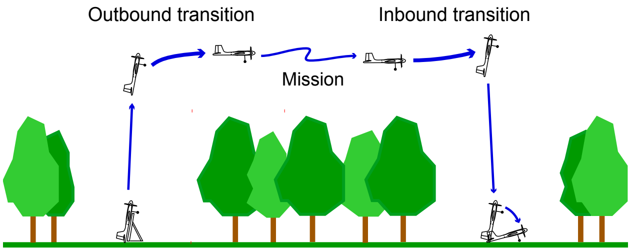

Fig.2 Operation scenario over forest area from vertical takeoff to vertical landing

3 Control System Architecture

Only the longitudinal dynamics of the vehicle were considered in this paper. The control inputs to the longitudinal dynamics are elevator deflection δ_{ele} and throttle setting δ_{thr}. The major challenges faced in this problem are as follows:

1. Wide operating range of the vehicle from hover to cruise with nonlinearity.
2. Control system robustness required for practical application.

Additionally, the following limitations exist:

3. Only limited information available about the vehicle dynamics.
4. Limited computational power of the control hardware.

Considering these issues, the control system architecture shown in Fig. 3 was proposed.

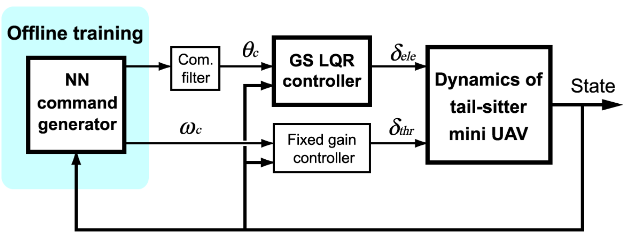

Fig.3 Control system architecture

3.1 Gain Scheduled Linear Quadratic Regulator

A gain-scheduled (GS) linear quadric regulator (LQR) was used for the pitch attitude control [1,11]. This controller is suitable for a plant with a wide operating range and nonlinearity, such as the transitional flight of a tail-sitter. The pitch command signal is filtered by a first-order filter in order to reduce the impulsive elevator input at the initial rise due to the feed forward term of the pitch attitude GS controller. The propeller rotation speed controller was designed using a conventional fixed-gain LQR since the propeller rotation dynamics vary only marginally.

3.2 Reference Command Generator (Neural Network)

In order to achieve the control objectives, appropriate reference command signal sequences must be provided to the inner loop system. An improper reference command, such as a very rapid pitch up command in high-speed flight, may cause the vehicle to stall. Therefore, the reference command generator has to be designed considering such constraints, and at the same time the existence of uncertainties must also be considered. In this study, a neural network (NN) optimized by offline simulations is used as a reference command signal generator. This outputs the pitch attitude command θ_{c} and propeller rotation speed command ω_{c}. In order to improve robustness, the network parameters are optimized using VE-GA/PDSM considering constraints and uncertainties. The details will be described in the next section.

4 Variable Environment Genetic Algorithm

The VE-GA which the authors proposed in previous papers [1] it has a problem of a lack of accuracy of constraint evaluations because the weight factor of the penalty functions cannot be increased to maintain the algorithm's stability. Here, we reformulate the evaluation (fitness) functions and handling of constraints.

4.1 Formulation of a Robust Optimization

A general deterministic optimization problem without uncertainty can be formulated as follows:

$$\begin{aligned} &\text{Minimize:} \quad f(\boldsymbol{x}) \\ &\text{Subject to:} \quad g_i(\boldsymbol{x}) \le 0 \ \ (i = 1, 2, \cdots, m) \end{aligned} \tag{1}$$

where x is the parameter vector to be optimized, function $f(x)$ is an evaluation function to be minimized, and functions $g_i(x)$ indicate constraints. However, if these functions contain uncertainties δ, that is $f(x,\delta)$ and $g(x,\delta)$, they cannot be optimized using only such deterministic formulations because their values depend on δ.

In such a case, we need a robust optimization approach, which is optimization that considers variation of the evaluation function $f(x,\delta)$ and constraints $g(x,\delta)$. The following equations using statistical values of the functions are one way to formulate such a robust optimization problem:

$$\begin{aligned}&\text{Minimize:} \quad \mu_f(\boldsymbol{x}) + k_f\sigma_f(\boldsymbol{x})\\ &\text{Subject to:} \quad \mu_{g,i}(\boldsymbol{x}) + k_{g,i}\sigma_{g,i}(\boldsymbol{x}) \le 0 \\ &\qquad\qquad\qquad\qquad (i = 1, 2, \cdots, m)\end{aligned} \tag{2}$$

where μ and σ are respectively the mean and standard deviation values of the functions. Using a penalty function method, the robust optimization problem including constraints can be written as the following single evaluation function J_{rob}:

$$\begin{aligned}\text{Minimize:} \quad & J_{\mathrm{rob}}(\boldsymbol{x}) = \mu_f(\boldsymbol{x}) + k_f\sigma_f(\boldsymbol{x})\\ & + \sum_{i=1}^{m} w_{g,i} \max\left(0,\ \mu_{g,i}(\boldsymbol{x}) + k_{g,i}\sigma_{g,i}(\boldsymbol{x})\right)\end{aligned} \tag{3}$$

where $w_{g,i}$ (i = 1, 2, ..., m) are penalty weighting factors. If this Eq. (3) can be optimized, a robust optimal solution will be obtained. However, this “direct” optimization is inappropriate because of the high calculation cost of evaluating the statistical terms.

We previously proposed a new robust optimization method, VE-GA [1], based on a real coded genetic algorithm (RCGA) [12,13]. Here, we use a non-statistical fitness function $F(x)$ defined as follows:

$$F(\boldsymbol{x},\boldsymbol{\delta}) = f(\boldsymbol{x},\boldsymbol{\delta}) + \sum_{i=1}^{m} v_{g,i} \max\left(0,\ g_i(\boldsymbol{x},\boldsymbol{\delta})\right) \tag{4}$$

This function considers only one sampling point δ in the uncertainty space. For further discussions below, the following index $F_{\mathrm{rob}}(x)$ is defined:

$$F_{\mathrm{rob}}(\boldsymbol{x}) = \mu_F(\boldsymbol{x}) + k_F\sigma_F(\boldsymbol{x}) \tag{5}$$

where $\mu_F(x)$ and $\sigma_F(x)$ are respectively the mean and standard deviation of the function $F(x,\delta)$ against uncertainties δ.

4.2 Variable Environment Concept

A VE-GA is a probabilistic robust optimization method based on an RCGA. An algorithm flow for a conventional RCGA is illustrated in Fig. 4. An initial population is created randomly, and this population is then optimized through evolutionary operations such as crossover, alternation, and mutation. In this research, we use unimodal normal distribution crossover (UNDX) [12] as the crossover method and the distance dependent alternative (DDA) model [13] as the alternation model for the crossover. Mutation operations are not used. The evolutionary process is continued until a given maximum number of generations (iterations).

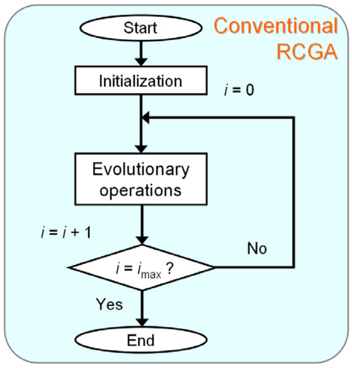

Fig.4 Flow chart of conventional real coded genetic algorithm (RCGA)

For our proposed VE-GA, a variable environment algorithm is added to the conventional RCGA as shown in Fig. 5. Here, the word “environment” implies the uncertainties δ considered in the evaluation of the fitness function $F(x,\delta)$. If a counter j reaches a value E, the environment setting is updated randomly, the parameter E is updated, and the counter j is reset. Therefore, a variation of the uncertainties δ is considered through the iteration of the optimization process. If we focus only on single generation step, the fitness function $F(x,\delta)$ for the particular value of δ given for that generation is evaluated not statistically but deterministically. However, over a number of generations, various $F(x,\delta)$

based on different values of δ will be evaluated, and this can be considered as virtually equivalent to a statistical evaluation.

We therefore term this algorithm the variable environment genetic algorithm, VE-GA. Using this algorithm, genes go through many types of environments over generations; in this manner, they not only converge to an optimal solution for one particular environment but also have moderate fitness for all environments.

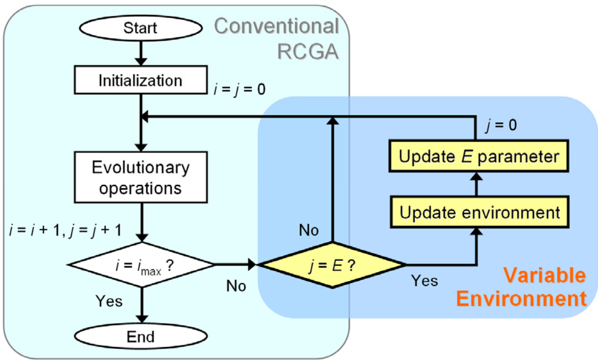

Fig.5 Flow chart of proposed variable environment genetic algorithm (VE-GA)

4.3 Parameter *E* Update Rule

Partially similar concepts can be found in other researches [14,15], but these researches use a "fixed value *E*". If a fixed *E* is used, only "average" optimization results, that is the genes can be optimized in a manner of average $\mu_F(x)$ (see Eq. (5)), and so the controller cannot obtain sufficient robustness. In this study, we propose a new algorithm to sequentially update *E* so that it is not fixed but variable, and thereby obtain a robust population.

The *E* update rule concept is very simple: ***"if the new environment is 'difficult' for the current population, the parameter E is updated to a large value, and if the new environment is 'easy', the parameter E is updated to a small value."***
This concept can be formulated as:

$$E = \lfloor f_E(m) \rfloor \tag{6}$$

where the symbol $\lfloor \cdot \rfloor$ denotes the Gauss symbol (i.e. $\lfloor x \rfloor$ is the greatest integer that does not exceed x) and m is the average fitness of the current population to the new environment. The function $f_E(m)$ is a monotonically increasing function of m. At every environment change, E is updated using this equation depending on the value of m. There is an indefinitely large number of such monotonically increasing function options that we can design. In this paper, we will use a simple linear function with saturation defined as follows:

$$f_E(m) = \min\left(E_{\max}, \zeta_1 E_1 + \zeta_1 E_1\right) \tag{7}$$

where E_1, E_2, and E_{max} are constant parameters. The coefficients ζ_1 and ζ_2 are defined as follows:

$$\zeta_1 = \frac{m - m_{\min}}{m_{\max} - m_{\min}}, \quad \zeta_2 = \frac{m_{\max} - m}{m_{\max} - m_{\min}} \tag{8}$$

where m_{max} and m_{min} are the recorded maximum and minimum values over the history of the average fitness m. $E = E_{max}$ is used only for the first environment in order to avoid dividing by zero.

If a new environment is difficult for a current population (i.e. m is large), the parameter E is updated to a larger value. This implies that a difficult environment continues for more iterations than an easy environment. In this manner, the population becomes fit not only for major easy environments but also for minor difficult ones.

4.4 Guidelines for Setting Parameters

Three parameters (E_1, E_2, and E_{max}) are used for updating parameter E, and are strongly related to the stability of the algorithm. The following guidelines for the parameter settings were determined through numerical experiments.

1. E_1 should be set to 0 or another small number. This implies that if a new environment is sufficiently easy for the current population, further learning of the new environment is not required. $E_1 = 0$ is best for the most cases; however, sometimes the iteration may plateau depending on problems. In such cases, E_1 should be set to non-zero value such as 1.

2. E_2 should be set depending on the problem. If E_2 is set to a small value, the stability of the VE-GA increases. However, the computational cost also increases because the number of recalculations required for

the fitness functions of new environments increases. On the other hand, if E_2 is set to a large number, the computational cost decreases but so does stability. This is a trade-off problem.

3. E_{max} determines the upper limit of E. For some problems, E_2 should be set to a much greater value than the population size N. Then, E will be large particularly during initial iterations because m is still large for some environment at the initial phase of the optimization. If E is too large, though, the population may converge to a local optimum for the current environment before changing to a next environment, thus losing robustness. Therefore, E_{max} should be set to a moderate number. We recommend setting the value of E_{max} to similar to the order of the population size N. This also depends on the complexity of the problem; E_{max} may be set to a larger number for a complex problem and a smaller number for a simple problem.

5 Powell's Direction Set Method

The VE-GA proposed above has two problems. One is low efficiency in the latter phase of iteration. Since VE-GA does not continuously monitor a history of the true robust evaluation function J_{rob} to save calculation cost, it cannot keep the best elite individual at each generation, and so a good individual based on robust criterion J_{rob} can sometimes be lost in the evolutionary operations. The other problem is a lack of accuracy of constraint evaluations, because the weight factor of the penalty functions cannot be increased in order to maintain the stability of the algorithm. If the penalty weight factor is set too large, the population has difficulty moving close to the constraint lines because of the characteristics of RCGA.

To solve these problems, we use Powell's direction set method (PDSM) [16] (Fig. 6). The best individual selected from output groups generated through the VE-GA process is fed to PDSM as an initial solution for it. Although this direct robust optimization based on the evaluation function J_{rob} incurs high calculation cost, fewer iterations are required because of pre-convergence, so the total required computational cost is not unfeasibly high.

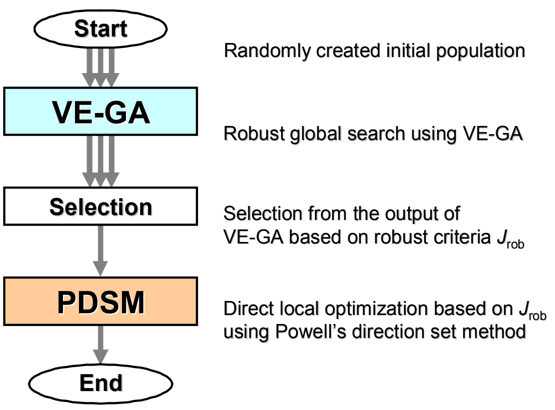

Fig.6 VE-GA/PDSM flow overview

6 Transitional Flights of Tail-Sitter VTOL Mini UAV

In this section, we apply the new robust optimization method proposed above to the problem of autopilot design for transitional flight of a tail-sitter mini unmanned aerial vehicle (UAV).

6.1 Preparation

Flight speed V was selected as the scheduling parameter, and three trimmed level flights (V = 0, 4, and 18 m/s) were selected as sampling points for the linear time invariant (LTI) models. A linear parameter varying (LPV) model of the tail-sitter was constructed by interpolating these LTI models [1] for the offline optimization simulations.

The NN used as the reference command generator has four, four, and two neurons for the input, hidden, and output layers, respectively. Thirty parameters are to be optimized in this problem. The inputs x and outputs r of the network are as follows:

$$\boldsymbol{x}^T = \begin{bmatrix} \dot{H} & V & \theta & q \end{bmatrix}^{\mathrm{T}}, \quad \boldsymbol{r}^T = \begin{bmatrix} \theta_c & \omega_c \end{bmatrix}^{\mathrm{T}} \qquad (9)$$

In this problem, model uncertainties, initial state errors from the trim conditions and a random gust were considered in the evaluation simulations for robust optimization. The model uncertainties are variations of the system matrices and initial conditions. It is assumed that every element of the system matrix has a variation range of ±30%. The initial conditions have variation ranges as follows: U_i, ±1 m/s; W_i, ±0.5 m/s; q_i, ±0.1 rad/s; θ_i, ±0.1 rad; ω_i, ±10 rad/s. With regard to gusts, the horizontal component has zero average and 1 m/s standard deviation, and the vertical component has zero average and 0.5 m/s standard deviation, which is a reasonable gust condition for mini UAV flights. Time series gust data were constructed from white noise through a second-order low pass filter with a cut-off frequency $\omega_f = 1$ rad/s.

6.2 Outbound Transition

Outbound transitions are relatively easy because the control results are only slightly affected by uncertainties or disturbances. Since outbound transitions are accelerating flight, the throttle settings are higher than in trimmed and decelerating flight, and so the propeller slipstream over the wing is strong and the effective angle of attack becomes smaller. As a result, the main wing does not stall even in low-speed flight in gusty wind conditions. Because of this, there are only very small differences between the results obtained using the conventional RCGA and those obtained using the proposed VE-GA/PDSM.

6.3 Inbound Transition

Inbound transitions on the other hand are relatively difficult. There are remarkably large differences between the results obtained using conventional RCGA and the proposed VE-GA/PDSM.

The objective function to be minimized is given as:

$$\text{Minimize: } f(\boldsymbol{x}) = w_{f,1}\left|V_{\text{tar}} - V_{tf}\right| + w_{f,2}\int_0^{t_f} \dot{H}^2 dt + w_{f,3}P_{\text{NN}}(\boldsymbol{x}) \tag{10}$$

The first term causes the vehicle to change its flight speed to the target flight speed V_{tar}. For inbound transition, this is the hovering speed $V_{\text{tar}} = 0$ m/s. The second term is to reduce variations in altitude. The third term, P_{NN}(x) is a regularization term to constrain the size of the parameter vector x. If the parameters of the network are large, the network will have high sensitivity, which means its outputs change sensitively against its inputs. This term is defined as:

$$P_{NN}(\boldsymbol{x}) = \sum_i x_i^2 \tag{11}$$

Constraints are given as:

$$\begin{aligned} g_1(\boldsymbol{x}) &= \alpha_s - \alpha_{s,\text{stall},u} \le 0 \\ g_2(\boldsymbol{x}) &= \alpha_{s,\text{stall},l} - \alpha_s \le 0 \end{aligned} \tag{12}$$

It is very important to maintain flight conditions to within the controllable region. If the angle of attack becomes too high, lateral control margin decreases because large aileron deflection can cause the main wing to stall, which is an uncontrollable condition. The parameter settings are listed in Table 1.

Table 1 Calculation settings

$f(\boldsymbol{x}, \boldsymbol{\delta})$	Weight factor	$\boldsymbol{w}_f = (1, 0.15, 10^{-3})$
	Stall angle of attack	$(\alpha_{s,\text{stall},l}, \alpha_{s,\text{stall},u}) = (-15, +25)$ deg
$F(\boldsymbol{x})$	Penalty weight factor	$\boldsymbol{v}_g = (1, 1)$
$F_{\text{rob}}(\boldsymbol{x})$	Standard deviation weight factor	$k_F = 1$
$J_{\text{rob}}(\boldsymbol{x})$	Penalty weight factor	$\boldsymbol{w}_g = (20, 20)$
	Standard deviation weight factor	$k_f = 1, k_{g,1} = k_{g,2} = 3$
VE-GA	E update parameters	$(E_1, E_2, E_{\max}) = (0, 500, 200)$
	Population size	$N = 100$
	Maximum number of generation	$G = 5\times10^4$
PDSM	Maximum number of iteration	$I = 3$
	Number of sampling points for statistical estimation of μ and σ	$M = 1000$

6.4 Comparison of Evaluation Value Histories and Discussion

The following four types of optimizations were tested: a) conventional GA without considering uncertainties, b) PDSM only with a randomly selected initial solution, c) VE-GA only, and d) the proposed VE-GA/PDSM. The histories of F_{rob} (Eq. (5)) and J_{rob} (Eq. (3)) are plotted in Fig. 7.

In case a), the conventional GA, robustness was lost rapidly just after iteration started because it is based only on the nominal case without considering uncertainties. In case b), using only PDSM, a large number of iterations were required and what is worse, an appropriate solution was not obtained because of convergence to a local optimum.

In case c), using only VE-GA, a relatively better value of index F_{rob} was achieved and the value F_{rob} was kept below 8; this steady state is a "convergence" of VE-GA. However, the convergence of index J_{rob} was not steady. From this, it can be seen that VE-GA is a method to optimize based on the index F_{rob}. It is natural to understand that VE-GA optimizes based on the index F_{rob}, because $F(x,\delta)$ is used in each generation step of VE-GA.

However, the primary objective is to optimize J_{rob}, and J_{rob} is better for handling penalty terms accurately because it contains explicit penalties considering variations caused by uncertainties. In contrast, F_{rob} cannot express penalties accurately. Therefore, actual robust evaluation J_{rob} is not stable after convergence of VE-GA, which means the population, even the optimum one in respect to J_{rob}, sometimes slightly violates the constraints. However, if PDSM is used additionally, case d), the proposed VE-GA/PDSM algorithm is able to reach a better solution based on index J_{rob}.

What it comes down to is as follows: The ultimate objective is optimization based on J_{rob}, but VE-GA is an optimization method based on F_{rob}. However, the VE-GA based on F_{rob} can yield a sufficient initial solution for PDSM, and PDSM can improve the solution based on J_{rob} directly so that finally, we obtain a significantly better solution than by other methods.

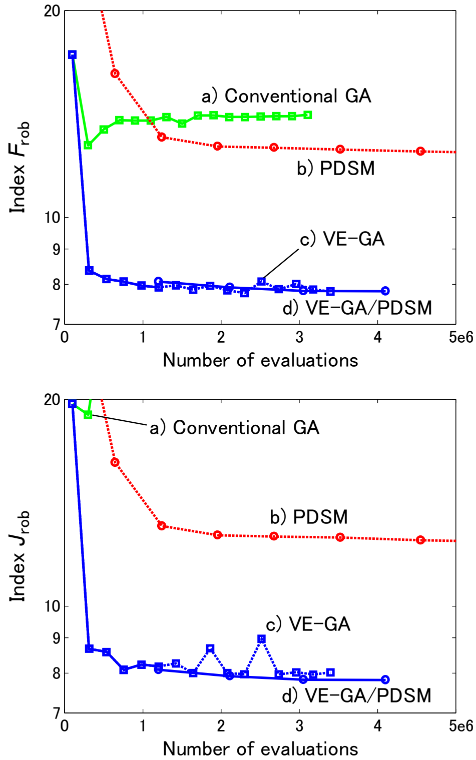

Fig.7 Histories of robust evaluation index F_{rob} (upper) and J_{rob} (bottom)

6.5 Comparison of Control Results

The NNs optimized with the above-mentioned algorithms were evaluated using Monte Carlo simulations (MCSs) based on a nonlinear flight simulation of the tail-sitter mini UAV. The results are shown in Figs. 8–11.

Since inbound transitions are decelerative flight, throttle settings are lower than in trimmed and accelerative flight, the propeller slipstream over the wing is weaker and the effective angle of attack becomes larger. Hence, stall margin decreases and flight can easily violate the stall constraint due to model uncertainties and disturbances. Flights controlled by an NN trained using the conventional GA therefore often violated the stall constraint (Fig. 8).

The results using b) PDSM (Fig. 9) show that the control objective is not achieved because of the local optimum.

In the results using c) VE-GA only (Fig. 10), the stall constraint is still violated in a few case. On the other hand, for case d) using proposed the VE-GA/PDSM (Fig. 11), the stall

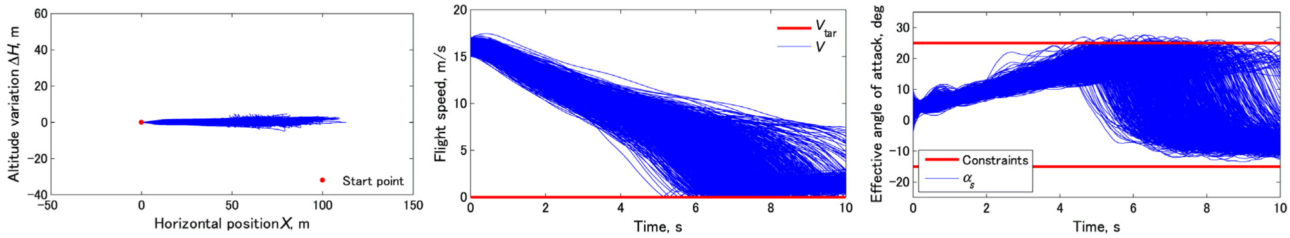

Fig.8 Monte Carlo simulation results of inbound transition control using NN optimized with conventional GA without considering uncertainties

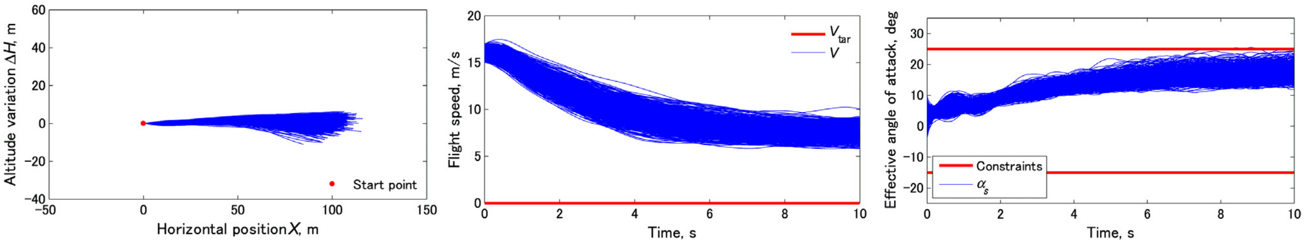

Fig.9 Monte Carlo simulation results of inbound transition control using NN optimized with PDSM with a randomly selected initial solution

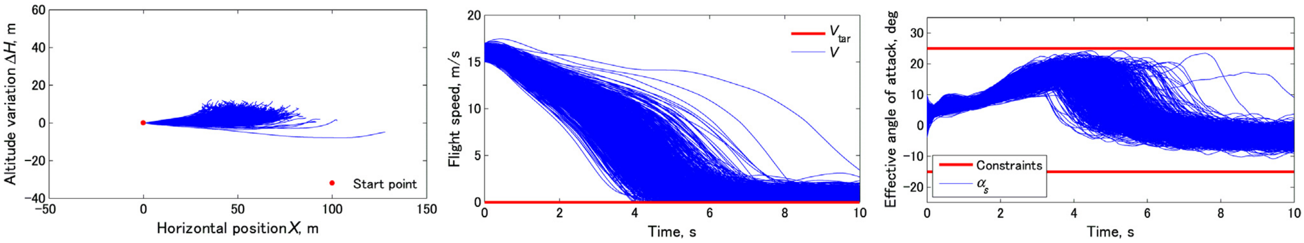

Fig.10 Monte Carlo simulation results of inbound transition control using NN optimized with only VE-GA

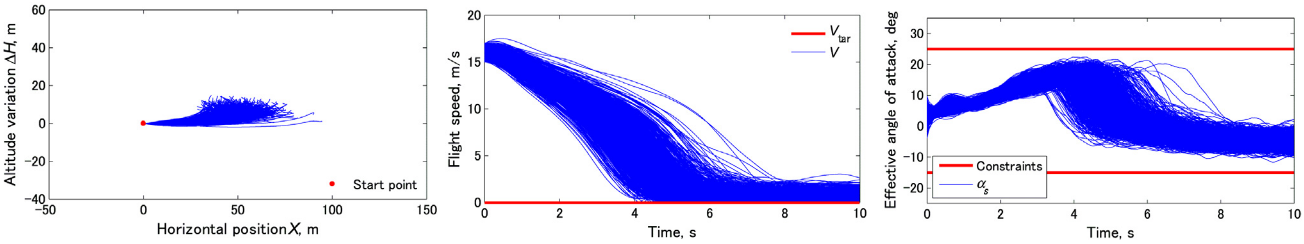

Fig.11 Monte Carlo simulation results of inbound transition control using NN optimized with proposed VE-GA/PDSM

constraint is not violated and the terminal altitude deviations are smaller.

7 Summary and Future Work

A variable environment genetic algorithm (VE-GA) and Powell's direction set method (PDSM) were used to optimize the parameters of a neural network (NN) that was used as a reference command generator to achieve the control objectives of transitional flight with small altitude changes without violating the stall constraint. VE-GA is a new robust optimization method based on the real coded genetic algorithm (RCGA) in which the environment (i.e. uncertainty settings considered in evaluation of the fitness function) is changed repeatedly after several generations. In this manner, genes go through many types of environments over generations and obtain robustness. PDSM was used to improve the efficiency and accuracy of the optimization process. We term the combined method VE-GA/PDSM. A notable characteristic of this method is that it is a “global” robust optimization method.

Finally, the method was applied to the design of an inbound transitional flight controller for a tail-sitter mini unmanned aerial vehicle (UAV). In the case of inbound transition, the new algorithm was found to be superior, and robust controllers for the transition were obtained.

While numerical simulations were used in this paper, it is necessary to conduct real flight tests for further evaluation.

The robust optimization method proposed in this paper is a general methodology and therefore can be applied to a variety of robust optimization problems.

Acknowledgments

The research was partially supported through the 21st Century Center of Excellence Program "Mechanical Systems Innovation" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] Kubo, D. and Suzuki, S., Robust Optimal Autopilot Design for a Tail-Sitter Mini Unmanned Aerial Vehicle, *Journal of Aerospace Computing, Information, and Communication*, Vol. 5, 2008, pp. 135–154.

[2] Hirokawa, R., Kubo, D., Suzuki, S., Meguro, J., and Suzuki, T., A Small UAV for Immediate Hazard Map Generation, *Proceedings of Infotech@Aerospace 2007*, Rohnert Park, California, AIAA Paper 2007-2725, 2007.

[3] Abershitz, A., Penn, D., Levy, A., Shapira, A., Shavit, Z., and Tsach, S., IAI's Micro/Mini UAV Systems-Development Approach, *Proceedings of Infotech@Aerospace*, Arlington, VA, AIAA Paper 2005-7034, 2005.

[4] Schulz, H. W., Buschmann, M., Kordes, T., Kruger, L., Winkler, S., and Vorsmann, P., The Autonomous Micro and Mini UAVs of the CAROLO-Family, *Proceedings of Infotech@Aerospace*, Arlington, Virginia, AIAA 2005- 7092, 2005.

[5] Rob Ransone, An Overview of VATOL Aircraft and Their Contributions, *Proceedings of the AIAA biennial International Powered Lift Conference and Exhibit*, Williamsburg, Virginia, 2002.

[6] Kubo, D., Study on design and transitional flight of tail-sitting VTOL UAV, *Proceedings of 25th Congress of ICAS*, Hamburg, Germany, 2006.

[7] Kubo, D. and Suzuki, S., Tail-Sitter Vertical Takeoff and Landing Unmanned Aerial Vehicle: Transitional Flight Analysis, *Journal of Aircraft*, Vol.45, No.1, pp. 292–297, 2008.

[8] Schaefer, C. G., Jr. and Baskett, L. J., GoldenEye: The Clandestine UAV, *Proceedings of 2nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations*, San Diego, California, AIAA 2003-6634, 2003.

[9] Taylor, D. J., Ol, M., and Cord, T., SkyTote: An Unmanned Precision Cargo Delivery System, *Proceedings of AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Yr*, AIAA 2003-2753, 2003.

[10] Stone, R. H., The T-Wing Tail-Sitter Research UAV, *Proceedings of Biennial International Powered Lift Conference and Exhibit*, Williamsburg, Virginia, AIAA Paper 2002-5970, 2002.

[11] Fujumori, A., Terui, F. and Nikiforuk, P. N., Flight Control Design of an Unmanned Space Vehicle Using Gain Scheduling, *Journal of Guidance, Control, and Dynamics*, Vol.28, No.1, pp. 96–105, 2005.

[12] Ono, I. and Kobayashi, S., A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, *Proceeding of 7th International Conference on Genetic Algorithms*, 1997, pp. 246–253.

[13] Takahashi, O., Kita, H., and Kobayashi, S., A Real-Coded Genetic Algorithm using Distance Dependent Alternation Model for Complex Function Optimization, *Proceeding of Genetic and Evolutionary Computation Conference 2000*, 2000, pp. 219–226.

[14] Tezuka, S., Torii, T., and Kouno, M., Structure and Learning Method of Neural Network that Controls Landing of Unmanned Helicopter, *Proceeding of the 42nd Aircraft Symposium*, Yokohama, Japan, 2004 (in Japanese).

[15] Imae, J., Kudo, S., Mori, K., and Torisu, R., A Practical Design Method for Nonlinear Robust Control Systems using Genetic Algorithms, *Transactions of the Japan Society of Mechanical Engineers. C*, Vol. 66, No. 651, 2000, pp. 3647–3654 (in Japanese).

[16] Powell, M. J. D., An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives, *The Computer Journal*, Vol. 7, No. 2, 1964, pp. 155–162.

Copyright Statement