JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 5, May 2008

Robust Optimal Autopilot Design for a Tail-Sitter Mini Unmanned Aerial Vehicle

Daisuke Kubo* and Shinji Suzuki†
The University of Tokyo, Tokyo 113-8656, Japan

DOI: 10.2514/1.33514

The transitional flight of tail-sitter vertical takeoff and landing mini unmanned aerial vehicles is a challenging problem because the flight includes wide operating ranges with nonlinearity and approaches stall in low-speed flights. Additionally, the sensor and computation performance of mini unmanned aerial vehicles is limited because of their size, weight, and cost. Simple controllers for a nonlinear system are therefore required, with robustness being a very important parameter. In order to solve the problem of nonlinearity, a gain-scheduled linear quadric regulator was used as a pitch attitude controller and an offline trained neural network was used as a transition reference command generator. A variable environment genetic algorithm was used to improve the robustness of the neural network. Here, "environment" implies the model errors and disturbance settings considered in the fitness evaluation simulations. The environment is changed repeatedly after several generations. In this manner, genes go through many types of environments over generations and they achieve moderate fitness for all environments. Using this algorithm, robust controllers for the transitional flight were obtained. The robustness was verified using Monte Carlo simulations.

I. Introduction

UNMANNED aerial vehicles (UAVs) have been playing important roles in military missions [1]; further, their use in civilian missions has been increasing [2, 3]. Their typical tasks include the reconnaissance of hazardous areas, rescue activities, meteorological monitoring, and so on. Mini UAVs are promising candidates for civilian use because they have advantages with regard to operational costs and human resources. Furthermore, they can be used in special missions such as high-resolution aerial photography from low altitudes, which larger UAVs cannot accomplish. In fact, "one-person portable" mini UAVs such as the Raven, BirdEye-500 [2], Carolo [4], and OBK-SkyEye [3] have demonstrated their potential in civilian missions. Although these mini UAVs are relatively small, they still require a considerable amount of space for takeoff or landing. The selection of areas for takeoff and landing is sometimes difficult because of buildings, trees, and hills that obstruct the takeoff or landing paths. The Raven uses a deep-stall vertical descent technique while the BirdEye-500 uses parachutes and air-cushions to improve the landing performance. However, these methods also have difficulties: although the deep-stall technique can help vehicles land in small areas, the descent rate is not sufficiently slow for some types of precision payloads. Parachutes have the disadvantage that accurate landings are not possible in strong wind conditions.

Vertical takeoff and landing (VTOL) aircraft have many operational advantages. Although the operation of conventional aircrafts is limited by the requirement of long runways, VTOL aircrafts can be operated from relatively small areas. Although helicopters have an inherent VTOL capability, their cruise performances characteristics such

Received 17 July 2007; accepted for publication 20 March 2008.

* Graduate Student, Department of Aeronautics and Astronautics, student member AIAA, kubo.daisuke@jaxa.jp
† Professor, Department of Aeronautics and Astronautics, AIAA member.

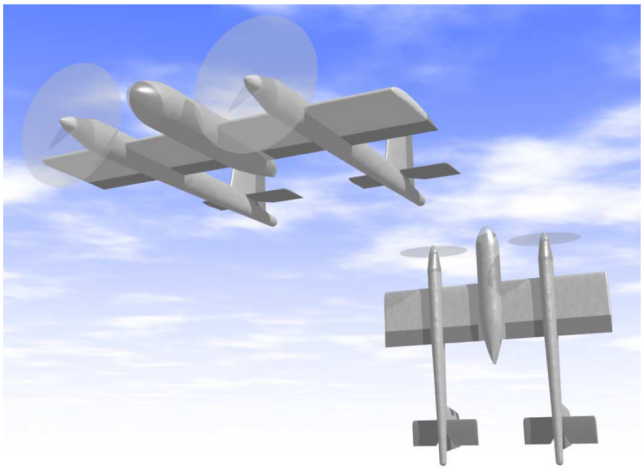


Fig. 1 The proposed tail-sitter VTOL mini UAV.

as flight speed, duration, and endurance are inferior to those of fixed-wing aircrafts. Aircraft with a VTOL capability similar to that of helicopters and a high-performance cruising ability similar to that of fixed-wing airplanes are therefore desired. The SkyTote [5], GoldenEye [6], Heliwing [7], and T-wing [8] are tail-sitter VTOLs. The tail-sitter has the advantage of not requiring variable mechanisms such as a tilt-rotor and tilt-wing. However, the abovementioned tail-sitter aircraft are still relatively complicated machines, and a simpler design for such aircraft is possible.

A new design for a tail-sitter VTOL mini UAV (shown in Fig. 1) was proposed in a previous research by the authors [9]. In that research, the flight characteristics were revealed through a trim analysis and an optimized transitional flight path analysis by using the mathematical model. These analyses revealed the necessity of leading-edge slats for the proposed aircraft. The controllers obtained from the optimized transitional flight analysis were the optimal feed-forward controllers. Therefore, the aircraft that use feed-forward controllers are ideal only in ideal conditions: no model errors and no gusts. As the feed-forward controller is very sensitive to variations in the system, that is its robustness is extremely low, it is difficult to implement it on real hardware and fly actual vehicles.

Research and experiments have been conducted on the transitional flight of tail-sitter-type vehicles. One such research describes the transitional flight of a fixed-wing UAV, which has a configuration similar to that of an aerobatic airplane [10]. Adaptive neural network (NN) control was used in the flight controller in reference [10]. Another is a T-wing UAV [8], which used a model predictive control technique for the transitional flight [11]. As the control methods of these UAVs require onboard computers with relatively high calculation power, they are not suitable for a "mini" tail-sitter that has limited onboard calculation power.

Therefore, the objective of the current paper is the design of feasible, simple, and robust controllers for the transitional flight of tail-sitter mini UAVs. A gain-scheduled linear quadric regulator (GS-LQR) was used for controlling the pitch altitude. This controller is suitable for a plant having a wide operating range and nonlinearity, such as the current problem. An offline trained NN was used for the outer loop controller, that is the reference command generator. The NN parameters are trained using a variable "environment" genetic algorithm (VE-GA) to improve the robustness of the system. Here, "environment" implies the model errors and disturbance settings assumed in the offline simulations for calculating the fitness function for the GA. The environment was changed repeatedly after several generations. In this manner, the population becomes robust. Using this algorithm, robust controllers for the transitional flight were obtained. The robustness was verified using Monte Carlo (MC) simulations.

II. Tail-Sitter VTOL Mini UAV

A. Design Features of the Vehicle

The proposed tail-sitter mini UAV is shown in Fig. 1. The vehicle has a 1 m wingspan. It is designed to have a takeoff weight of 2 kg. Although experimental tail-sitter VTOL UAVs such as GoldenEye, SkyTote, Heliwing, and

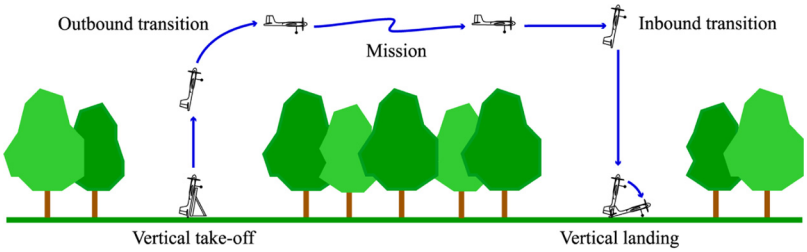


Fig. 2 Operation scenario from vertical takeoff to vertical landing.

T-wing already exist, the proposed design has the following special features:

1) A twin counter-rotating propeller configuration, which is advantageous because the mechanism is significantly simpler than other candidates such as coaxial counter-rotating propellers/rotors. Another advantage is the wide angular range of the forward view from the payload sensors positioned in the main fuselage.
2) The ailerons, rudders, and elevators immersed in the slipstream of the propellers are sufficient to enable attitude control even in low-speed and hovering flight. No complicated control devices such as the cyclic pitch control system of rotors are required for attitude control in low-speed flight.
3) With the exception of the control surfaces, there are no other variable mechanisms. The tilt mechanisms of tilt-rotors, tilt-wings, and tilt-ducts would make the system complicated. Such mechanisms are generally not suitable for mini UAVs.

B. Operation Scenario

An assumed operation scenario for the tail-sitter mini UAV proposed in reference [8] is illustrated in Fig. 2. In the takeoff phase, the vehicle uses a hand launch or a support gear and climbs vertically to a certain altitude that is determined from the location of the operation area. The vehicle then increases its flight speed and transitions to forward wing-borne flight; this is called an outbound transition. After the completion of the mission phase, the vehicle approaches the landing point. It decreases its flight speed and transitions to the hovering mode; this is called an inbound transition. In the final landing phase, the aircraft descends vertically and touches down with the tail gears; it then drops forward to touch down with the main gears and is thus supported by both the tail and main gears.

III. Mathematical Model

A. Nonlinear Model

The dynamics of a nonlinear plant can be represented as follows

$$\begin{aligned} \dot{x}(t) &= f\left(x(t),\ u(t)\right) \\ z(t) &= g\left(x(t),\ u(t)\right) \end{aligned} \tag{1}$$

where x is a state vector, u is a control input, and z is an output of a system. A nonlinear simulation model was constructed to investigate the flight characteristics of the tail-sitter VTOL mini UAV in a previous research by the present authors [9]. Considering the nature of this preliminary study, complex aerodynamic phenomena had not been considered in detail. However, the stall characteristics and aerodynamic forces caused by propeller slipstreams had to be evaluated because these are the most important characteristics of the tail-sitter. Therefore, a combination of simple estimation methods was used. See reference [8] for details about the construction of the nonlinear model.

B. Linear Parameter Varying (LPV) Model

The robust controller design method proposed in this paper is a model-based approach; therefore, mathematical models of the plant are required. However, it is unrealistic to assume that accurate nonlinear models of the plant are available in practical controller design applications. In this paper, it is assumed that the linear time invariant (LTI) models of the vehicle at r trimmed points are available. This assumption is reasonable because the LTI models can

be obtained using experiments such as wind tunnel testing. The r LTI models of the plant are represented as follows

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\end{bmatrix} = \begin{bmatrix}A_i & B_i\\ C_i & D_i\end{bmatrix}\begin{bmatrix}x(t)-x_i^d\\ u(t)-u_i^d\end{bmatrix} + \begin{bmatrix}0\\ z_i^d\end{bmatrix}, \quad \text{for } i = 1,\ldots,r \tag{2}$$

where

$$A_i \equiv \frac{\partial f(x_i^d, u_i^d)}{\partial x^T}, \quad B_i \equiv \frac{\partial f(x_i^d, u_i^d)}{\partial u^T}, \quad C_i \equiv \frac{\partial g(x_i^d, u_i^d)}{\partial x^T}, \quad D_i \equiv \frac{\partial g(x_i^d, u_i^d)}{\partial u^T} \tag{3}$$

A linear parameter varying (LPV) model can be derived from these LTI models using a varying parameter ρ that identifies the r LTI models. An LPV model is constructed from linearly interpolated LTI models using the parameter ρ

$$\begin{bmatrix}\dot{x}(t)\\ z(t)\end{bmatrix} = \begin{bmatrix}A(\rho) & B(\rho)\\ C(\rho) & D(\rho)\end{bmatrix}\begin{bmatrix}x(t)-x^d(\rho)\\ u(t)-u^d(\rho)\end{bmatrix} + \begin{bmatrix}0\\ z^d(\rho)\end{bmatrix}, \quad \text{for } \rho \in [\rho_i, \rho_{i+1}] \tag{4}$$

Here

$$\mu_i(\rho) \equiv \frac{\rho_{i+1}-\rho}{\rho_{i+1}-\rho_i}, \quad \mu_{i+1}(\rho) \equiv \frac{\rho-\rho_i}{\rho_{i+1}-\rho_i} \tag{5}$$

$$\begin{bmatrix}A(\rho) & B(\rho)\\ C(\rho) & D(\rho)\end{bmatrix} = \mu_i(\rho)\begin{bmatrix}A_i & B_i\\ C_i & D_i\end{bmatrix} + \mu_{i+1}(\rho)\begin{bmatrix}A_{i+1} & B_{i+1}\\ C_{i+1} & D_{i+1}\end{bmatrix} \tag{6}$$

$$\begin{bmatrix}x^d(\rho)\\ u^d(\rho)\end{bmatrix} = \mu_i(\rho)\begin{bmatrix}x_i^d\\ u_i^d\end{bmatrix} + \mu_{i+1}(\rho)\begin{bmatrix}x_{i+1}^d\\ u_{i+1}^d\end{bmatrix} \tag{7}$$

The matrices A, B, C, and D, and variables x_d, u_d, and z_d, represented using ρ are given linearly to interpolate the intermediate region with the varying parameter ρ.

IV. Control System Architecture

The objective of the current paper is the design of feasible robust controllers for the transitional flight of the tail-sitter mini UAV. Only the longitudinal dynamics of the vehicle were considered. An elevator deflection and throttle setting are the control inputs to the longitudinal dynamics. The major challenges faced in this problem are as follows:

1) Wide operating range of the vehicle from hovering to cruising with nonlinearity.
2) Robustness of the control system for practical applications.

Additionally, the following limitations exist:

3) Available information about the vehicle dynamics is limited.
4) Computational power of the control hardware is limited.

Considering these issues, the control system architecture shown in Fig. 3 was proposed. A gain-scheduled LQR was used for the pitch attitude control. This controller is suitable for a plant with a wide operating range with nonlinearity, such as the transitional flight of the tail-sitter. The propeller rotation speed controller was designed using a conventional fixed-gain LQR because the dynamics of the propeller rotation vary only marginally.

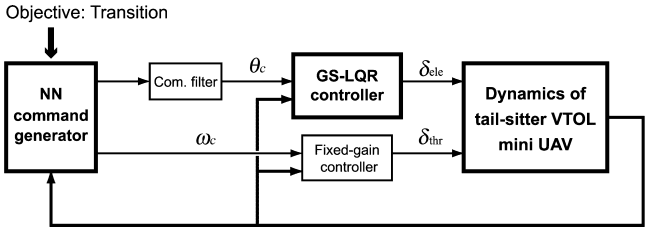


Fig. 3 Transitional flight control system architecture.

A three-layered NN is used as the outer loop controller, that is a reference command generator for the inner loop closed system. It outputs a pitch attitude command θ_c and a propeller rotation speed command ω_c. To improve the robustness of the control system, the network parameters are trained using a VE-GA, as proposed in Sec. VI. The details will be described later.

To reduce the impulsive elevator input at the initial rise owing to the feed-forward term of the pitch attitude controller, the pitch command signal is filtered using a first-order filter.

V. Gain-Scheduled LQR Controller

The dynamics of hovering flight and cruising flight of the tail-sitter differ significantly. Therefore, the dynamics cannot be controlled well by using only a single fixed-gain feedback controller. On the other hand, gain-scheduled controllers are widely used to control such nonlinear plants, particularly for practical applications. Gain-scheduled controllers occasionally have problems with their global stability [12]. However, whether or not such problems may arise for a particular implementation can be verified through careful simulations before the practical implementation. Therefore, these controllers are still useful for some practical applications.

A. Formulation

The gain-scheduled state feedback law is represented as

$$u_{\mathrm{fb}}(t) = -F(\rho)(x(t) - x^d(\rho)) \tag{8}$$

where

$$F(\rho) = \mu_i(\rho)F_i + \mu_{i+1}(\rho)F_{i+1}, \quad \text{for } \rho \in [\rho_i, \rho_{i+1}] \tag{9}$$

The feedback gains F_i ($i = 1, 2, \ldots, r$) are derived via the LQR theorem for each LTI model. It is assumed that the gains are designed appropriately and the closed-loop system is stable. In fact, the problems described in the present paper can be stabilized using this method.

As the objective of the controller is to track a reference command signal, a feed-forward input to track the command is required [13]. The total control input is then represented as follows

$$u(t) = u_{\mathrm{fb}}(t) + u^d(\rho) + u_{\mathrm{ff}}(\rho) \tag{10}$$

Then, the closed-loop system is represented as

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} A_F(\rho) & B(\rho) \\ C_F(\rho) & D(\rho) \end{bmatrix} \begin{bmatrix} x(t) \\ u_{\mathrm{ff}}(\rho) \end{bmatrix} + \begin{bmatrix} -A_F(\rho)x^d(\rho) \\ z^d(\rho) - C_F(\rho)x^d(\rho) \end{bmatrix}, \quad \text{for } \rho \in [\rho_i, \rho_{i+1}] \tag{11}$$

where z_c is a reference command signal. A_F and C_F are defined as follows

$$\begin{aligned} A_F(\rho) &\equiv A(\rho) - B(\rho)\ F(\rho) \\ C_F(\rho) &\equiv C(\rho) - D(\rho)\ F(\rho) \end{aligned} \tag{12}$$

If the closed loop is stable, the following steady state values x_∞ and z_∞ exist.

$$x_\infty = -A_F^{-1} B\, u_{\mathrm{ff}} + x_d \tag{13}$$

$$z_\infty = C_F x_\infty + D\, u_{\mathrm{ff}} - C_F x^d + z^d \tag{14}$$

To conform the steady state output z_∞ to the reference command signal z_c, the tracking feed-forward input should be represented as

$$u_{\mathrm{ff}}(\rho) = \{D(\rho) - C_F(\rho)\ A_F^{-1}(\rho)\ B(\rho)\}^{-1}\{z_c - z^d(\rho)\} \tag{15}$$

Finally, the total control input to the plant is given as

$$u(t) = -F(\rho)(x(t) - x^d(\rho)) + u^d(\rho) + \{D(\rho) - C_F(\rho)\ A_F^{-1}(\rho)\ B(\rho)\}^{-1}\{z_c - z^d(\rho)\} \tag{16}$$

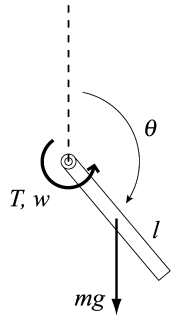


Fig. 4 Swing-up of inverted pendulum.

B. Example: Inverted Pendulum

The gain-scheduled LQR controller was applied to an example problem, that isangle tracking control of an inverted pendulum (shown in Fig. 4), to demonstrate its effectiveness. The problem is formulated as follows. The dynamics of the pendulum is written as

$$I_p\ddot{\theta} = \frac{1}{2}lmg\sin\theta - T \tag{17}$$

The model parameters are $l = 1$ m, $m = 1$ kg, and $g = 9.8$ m/s^2. The control objective is tracking an angle command signal θ_c in a range $[0, \pi]$. The dynamics are nonlinear with respect to the angle θ; which implies that it is difficult to control them using a fixed gain controller. Therefore, gain-scheduled controllers are preferred. The angle θ is used as a scheduling parameter for this problem. A linearized model around $\theta = \theta_0$ is obtained analytically as follows

$$\frac{d}{\mathrm{d}t}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ lmg\cos\theta_0/2I_p & 0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}0\\-1/I_p\end{bmatrix}u \tag{18}$$

Here

$$x_1 = \theta - \theta_0, \quad x_2 = \dot{\theta}, \quad u = T - \frac{1}{2}lmg\sin\theta_0 \tag{19}$$

Three linearization points, $\theta_0 = 0, \pi/2$, and π were selected and the LPV model was constructed. The feedback gains F_i $(i = 1, 2, 3)$ were designed using LQR for each LTI model. The following weight factors were used to derive the gains.

$$Q = \begin{bmatrix}1 & 0\\ 0 & 0.1\end{bmatrix}, \quad R = 0.001 \tag{20}$$

In this manner, the following gains are derived

$$F_1 = \begin{bmatrix}16.04 & 4.55\end{bmatrix}, \quad F_2 = \begin{bmatrix}10.00 & 4.08\end{bmatrix}, \quad F_3 = \begin{bmatrix}6.24 & 3.76\end{bmatrix} \tag{21}$$

After the gain-scheduled LQR controller was obtained, the controller was evaluated in a nonlinear simulation. The step response of the closed nonlinear system is shown in Fig. 5. The green dashed line indicates the step command while the red one indicates the filtered command, which is an actual reference for the system. The blue line indicates the system response. Although there is no steady state error at the linearized points $\theta = \pi, \pi/2$, and 0, there is some steady state error at the interpolated points $\theta = \pi/4$ and $3\pi/4$. This occurs because there is a modeling error in the LPV model used for the controller design. Such results are generally not desired. However, the purpose of this paper is not the achievement of accurate tracking control but the demonstration of the robustness of the total control system. Therefore, this is not a serious problem in the scope of the current paper.

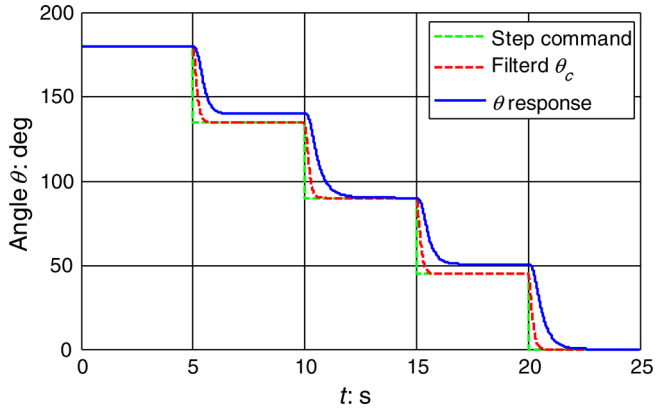


Fig. 5 Step response of inverted pendulum system with GS-LQR controller.

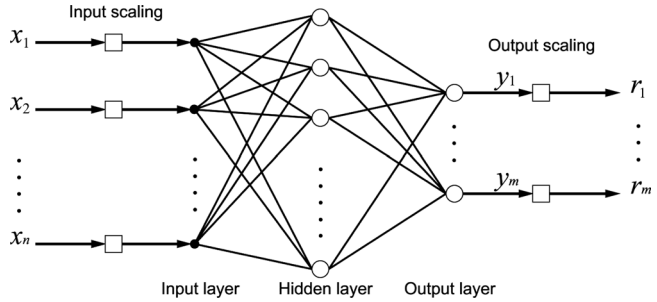


Fig. 6 Three-layered N with I/O scaling as a reference command generator.

VI. Neural Network Reference Command Generator

To achieve control objectives such as transitions, appropriate reference command signal sequences must be provided to the inner loop system. An offline trained NN is used as a generator for reference command signals. It is a type of nonlinear state feedback controller. The inputs of the NN are the state variables and the outputs are the reference command signals for the inner loop system. In this paper, a standard three-layered NN with input/output scaling (shown in Fig. 6) is used.

A sigmoid function, defined as follows, is used as a transfer function for the hidden layer and the output layer neurons

$$f(s) = \frac{1}{1 + \exp(-s)} \tag{22}$$

The actual reference command signal $r_i (i = 1, 2, \ldots, m)$ is obtained by scaling the output layer values $y_i (i = 1, 2, \ldots, m)$

$$r_i = (r_i^{\max} - r_i^{\min}) y_i + r_i^{\min} \quad (i = 1,\ 2, \ldots, m) \tag{23}$$

Here, $r^{\max}$ and $r^{\min}$ are the upper and lower limits of the reference command signal, respectively.

VII. Variable Environment Genetic Algorithm (VE-GA)

A real-coded GA is used to optimize the NN reference command generator (see Fig. 6). To improve the robustness of the NN, the environment in the evaluation simulations is randomly changed every E generations of the GA [14]. Here, environment implies the model uncertainty parameters and disturbance settings considered in the evaluation simulations. The authors of reference [14] called the algorithm "gradual evolutions." In this paper, we term the parameter E "era," that is, a period of generations that have the same environment. Using this algorithm, genes go

through many types of environments over generations; in this manner, they not only converge to an optimal solution for a certain special environment but also have moderate fitness for all environments. In other words, the population becomes more robust.

In reference [12], a constant E was used. If the value of E is given correctly, the algorithm occasionally works well. However, this algorithm may occasionally not work well if the setting of E is inadequate. Moreover, even if E is set adequately, the algorithm does not work well for certain types of problems. If information about the environment, such as gust speeds, is available to the NN controller as inputs, the NN may be trained appropriately using the constant era algorithm. This is similar to curve fitting through sequence learning. However, if the information is not available to the NN, it can only approximately fit the average environment. Therefore, the controller cannot obtain sufficient robustness. As the information about the environment is not available in the case of the tail-sitter, the GA with constant era is inadequate.

In the current paper, a new algorithm using a variable era is proposed. A variable parameter E is used instead of the constant parameter E and the value is updated in GA iterations. By applying this algorithm, the stability of the GA was improved and a robust solution can be obtained stably. The parameter E is updated using the following equation

$$E = [\min(E_{\max}, \zeta_1 E_1 + \zeta_2 E_2)] \tag{24}$$

Here, the symbol $[\cdot]$ denotes the Gauss symbol; $[x]$ is the greatest integer that does not exceed x. Parameters $E_{\max}$, E_1, and E_2 are constant parameters. The coefficients ζ_1 and ζ_2 are defined as follows

$$\zeta_1 \equiv \frac{m_{\max} - m}{m_{\max} - m_{\min}}, \quad \zeta_2 \equiv \frac{m - m_{\min}}{m_{\max} - m_{\min}} \tag{25}$$

Here, m is the average fitness of the current generation and $m_{\max}$ and $m_{\min}$ are the recorded maximum and minimum values of the averages in the current and past generations, respectively. For the first era, as $m = m_{\min} = m_{\max}$, Eq. (25) cannot be applied. Therefore, $E = E_{\max}$ is used only for the first era. These update laws are illustrated in Fig. 7.

Therefore, if the environment of a new era is unsuitable for the current population, the parameter E is updated to a larger value. This implies that a difficult era continues for a longer duration than an easy era. In this manner, the population cannot only fit for major easy environments but also for minor difficult environments.

In the present paper, the value J_{rob} (lower values are preferable) is used as a measure of the robust optimum

$$J_{\text{rob}} = J_{\text{ave}} + \sigma_J \tag{26}$$

It comprises the average fitness J_{ave} and the standard deviation σ_J. In this paper, the population size is set to 100.

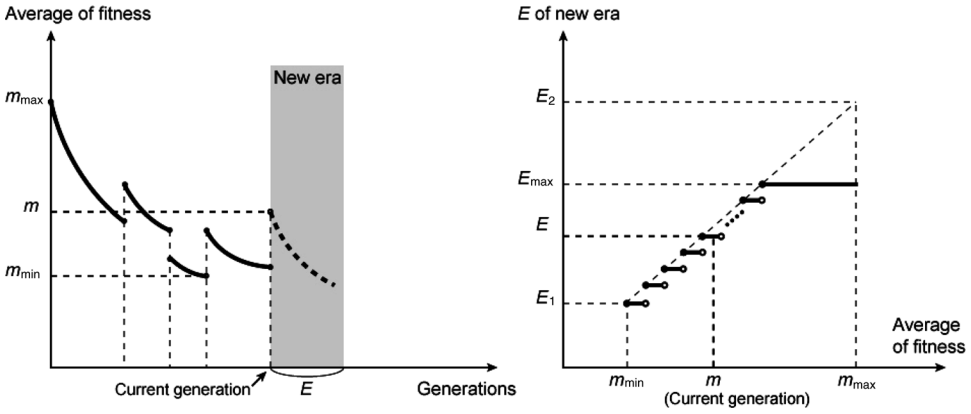


Fig. 7 Update law for variable era parameter E.

A. Guidelines of Parameter Settings

The parameters E_1, E_2, and $E_{\max}$ are used for the calculations of an era setting E. They are strongly related to the stability of the algorithm. The guidelines for the parameter settings were obtained through numerical experiments as follows:

1) E_1 should be set to 1 or another such small number. This implies that if a new environment is easy for the current population, further learning of the new environment is hardly required. If E_1 is set to zero, the algorithm occasionally work better; however, the learning may occasionally plateau, which is not appropriate.
2) E_2 should be set depending on the problems. If E_2 is set to a small value, the stability of the VE-GA increases. However, the computational cost also increases significantly because the number of recalculations required for the fitnesses of new environments increases. On the contrary, if E_2 is set to a large number, the stability as well as the computational cost decrease. This is a trade-off problem.
3) $E_{\max}$ should be set to a number equal or near to the population size. This also depends on the complexity of a problem; in complicated problems, $E_{\max}$ may be set to a slightly large number, while in simple problems, it may be set to a slightly small number.

B. Procedure of the Proposed VE-GA

The flowchart of the proposed VE-GA is shown in Fig. 8.

Step 1 An initial population is created randomly. The counters i, j, and k are initialized. The era parameter E is initialized as E_0. In this paper, $E_0 = E_{\max}$ is used. The environment is set to the initial condition and the fitnesses of all individuals in the population are evaluated.

Step 2 Next, crossover operations are conducted. In the current paper, the unimodal normal distribution crossover (UNDX) [15] is used as the crossover method, and the distance dependent alternative (DDA) model [16] is used as the alternation model for the crossover. After these crossover operations, the counters i and j are incremented. In this paper, five crossovers per generation are used. Mutation operations are not used.

Step 3 If generation i has reached a given maximum number of generations $i_{\max}$, go to end. If i has not reached $i_{\max}$, go to the next step.

Step 4 If the counter $j$has reached E (the end of the era), the environment setting is updated randomly and all individuals in the population are reevaluated for the new environment. Then, the counter j is reset and k is incremented. If j has not reached E, go back to step 2.

Step 5 The era parameter E is updated using the update equation Eq. (24). Then, go back to step 2.

After an interval of a certain number of generations, an MC simulation is conducted to select a robust optimal gene from the current population. The simulation involves 1000 evaluations of a fitness function J for different environment settings. The environments used in this MC are designed using the Latin hypercube sampling (LHS) [17] method to improve the precision of the estimation of J_{rob} with a small number of sampling points and evaluations.

Note that the LPV model is used in the evaluation simulations in the VE-GA because the nonlinear model is supposed to be unavailable in the controller design process.

C. Example: Minimization of a Two-Dimensional Function with Uncertainty

The proposed VE-GA was evaluated by applying it to the robust optimization of a simple test function with uncertainty. The function to be minimized is as follows

$$J(x, y) = (1 + p)(x - p)^2 + y^2 + p \tag{27}$$

Here, p is an uncertainty parameter defined as follows

$$p = w^6 \tag{28}$$

Here, w is a random value having a uniform distribution on an interval [0, 1].

The histories of the robustness evaluation values are shown in Fig. 9, and three optimization cases a) conventional GA, b) VE-GA with constant era, and c) VE-GA with variable era, and the final distributions of the population are shown in Fig. 10.

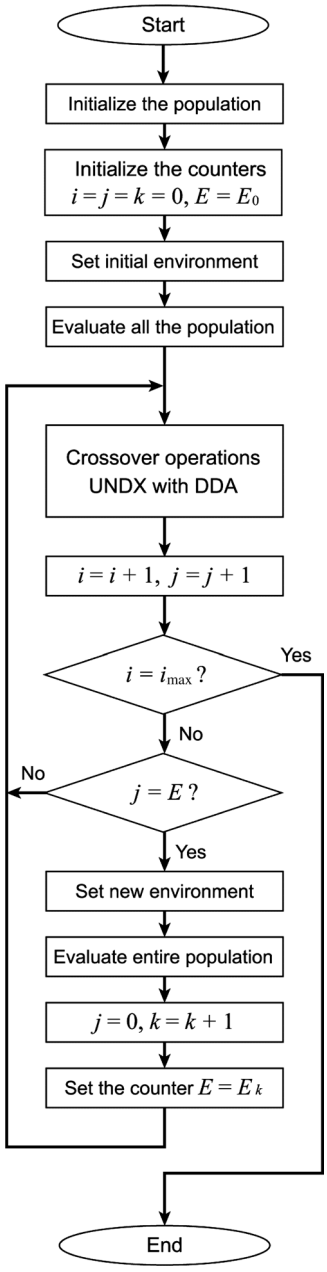


Fig. 8 Flowchart of the variable environment genetic algorithm (VE-GA) with a variable era.

In case a, immediately after beginning the iteration, a robustness evaluation J_{rob} decreased rapidly, following which it increased again. This implies that some individuals accidentally approached a robust optimum point in the early stages. However, the population finally converged to a nominal optimum, as shown in Fig. 10(a). In case b, J_{rob} also increased again after some generations. Although it did not increase to the same extent as that in case a, it is not desired. On the other hand, in case c, J_{rob} did not increase again after it had reduced. It stably maintained small values of J_{rob}.

With regard to Fig. 10, the blue dots denote the positions of population existing at the end of each GA. A green cross denotes a nominal optimum point $(x, y) = (0, 0)$ and a red cross denotes a robust optimum point

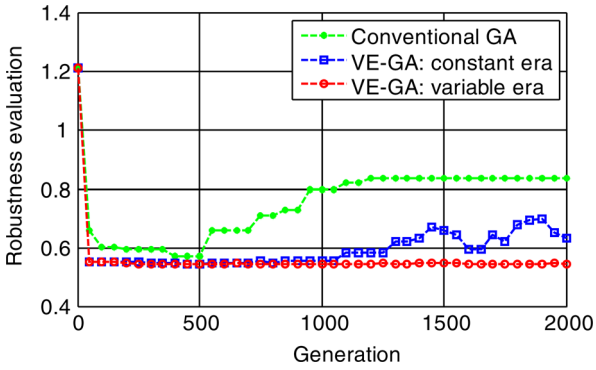
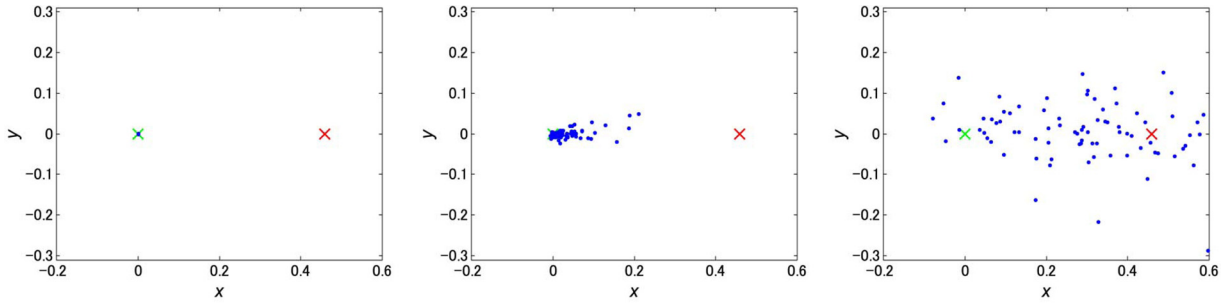


Fig. 9 Convergence history of GAs.



a) Conventional GA

b) VE-GA with constant era ($E = 10$)

c) VE-GA with variable era ($E_1 = 1$, $E_2 = 1000$, $E_{\max} = 20$)

Fig. 10 Distribution of population after GA iteration.

$(x, y) = (0.459, 0)$—an optimum of the robustness evaluation J_{rob}. The entire population in a) converged to the nominal optimum point. This implies that this conventional method cannot find a robust solution. Although method b did not converge to the nominal optimal point completely, it cannot approach a robust solution either. However, method c can stably stay close to the robust optimum point. Using the proposed VE-GA, the population gets scattered around the robust optimum point and stays close to it.

D. Example: Swing-up Control of Inverted Pendulum with Torque Constraint

The example problem described in Sec. V was used again to evaluate the proposed robust controller design method. Here, a disturbance torque w in the system was considered.

$$I_p\ddot{\theta} = \frac{1}{2}lmg\sin\theta - T - w \tag{29}$$

The disturbance torque w is constructed using a uniform distribution signal and a second-order low-pass filter. It has a zero average and a standard deviation $\sigma = 0.5\,\text{Nm}$. The existence of model errors is also considered in this problem. It was assumed that the parameters l and m have uncertainties of $\pm 20\%$.

The control objective is to move the pendulum rapidly from $\theta = \pi$ to $\theta = 0$ under the uncertainties and the disturbance, while satisfying the limited torque, given as

$$|T| \leq T_d \tag{30}$$

where $T_d = 6\,\text{Nm}$ is used. The objective function to be minimized is represented as

$$J = \int_0^{t_f} \{|\theta| + 3.0 \times 10^2 \times (|T| - T_d)^2\}\mathrm{d}t + 10^3 \times P_{\text{NN}} \tag{31}$$

where P_{NN} is a penalty function to constrain the gene vectors to stay inside a hypersphere with radius $r_p = 50$.

$$P_{\mathrm{NN}} = \max\left(0, \sqrt{\Sigma\, x_i^2} - r_p\right) \tag{32}$$

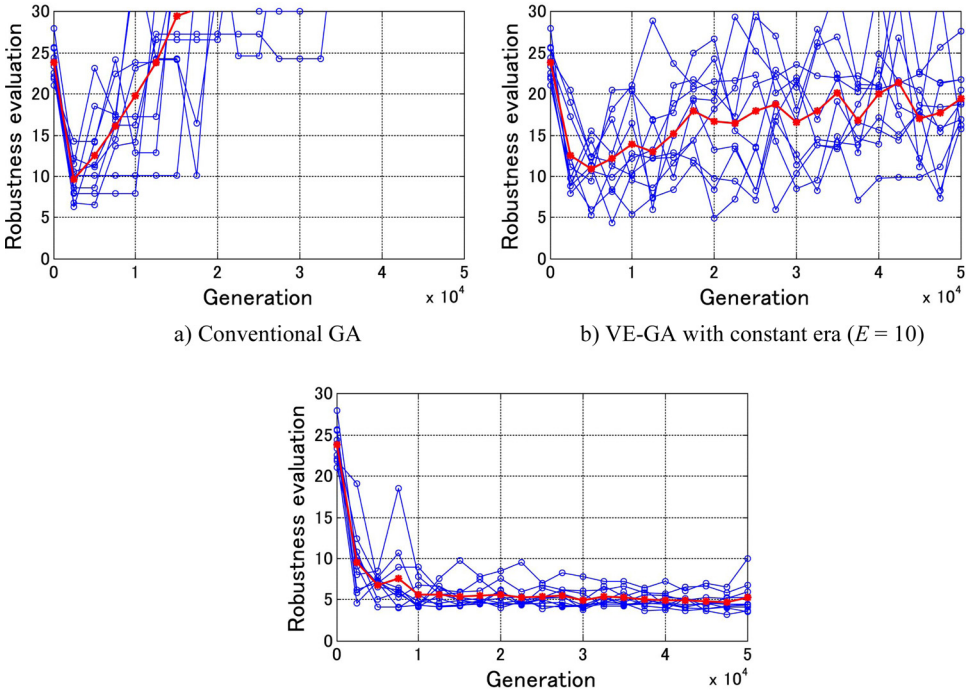
where x_i is the ith component of a gene. It is used to avoid unfavorable initial convergence to a local optimum existing at a long distance from the origin. NNs having large parameter values have high sensitivity, which is not a favorable characteristic for their robustness. Therefore, this penalty does not have any adverse effect.

Here, an NN used as an angle reference command generator has two, four, and one neurons for the input, hidden, and output layers, respectively; 17 parameters are to be optimized. The inputs x and outputs r are as follows

$$x^T = \begin{bmatrix} \dot{\theta} & \theta \end{bmatrix}, \quad r = [\theta_c] \tag{33}$$

The following three types of GA were tested: a) conventional GA, b) VE-GA with constant era ($E = 10$), and c) VE-GA with variable era ($E_1 = 1$, $E_2 = 1000$, $E_{\max} = 100$). The robustness evaluation histories of ten trials and the average for each GA are plotted in Fig. 11. The blue lines and circles denote the histories of each of the ten trials. The red lines and asterisks denote their averages. The evaluation value of case a increased with the number of iterations because the population converged to a nominal optimum. In case b, it did not increase to the same extent as that in a), although it did not constantly decrease either. On the other hand, in case c, it stably decreased.

The results of MC using the nonlinear model and the obtained NN controllers are compared in Fig. 12, and a quantitative comparison is shown in Table 1. Here, the obtained typical controllers were used; they are not the best cases of each of the ten trials but the average ones. The values in Table 1 are normalized such that the value J (nominal case) of the conventional GA is 1. In the case of training using conventional GA, J for the nominal case (without model errors and disturbances) was better than that in case b but worse than that in case c as the optimal solution in the conventional GA approaches the constraint such that J constantly improves. However, as there is a difference between the LPV simulation and the nonlinear simulation, the result violates the constraint in the nonlinear

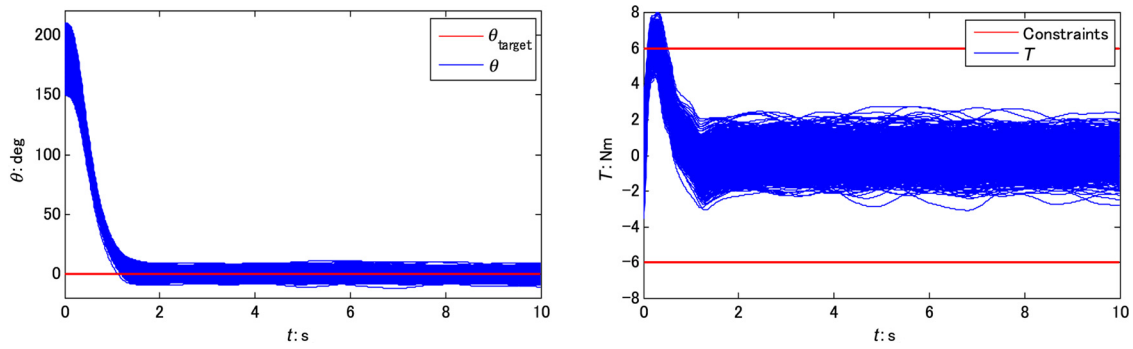


a) Conventional GA

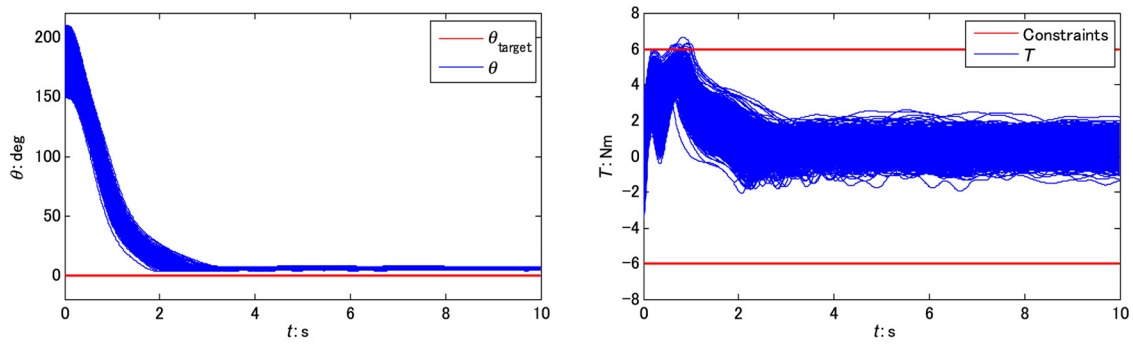
b) VE-GA with constant era ($E = 10$)

c) VE-GA with variable era ($E_1 = 1$, $E_2 = 1000$, $E_{\max} = 100$)

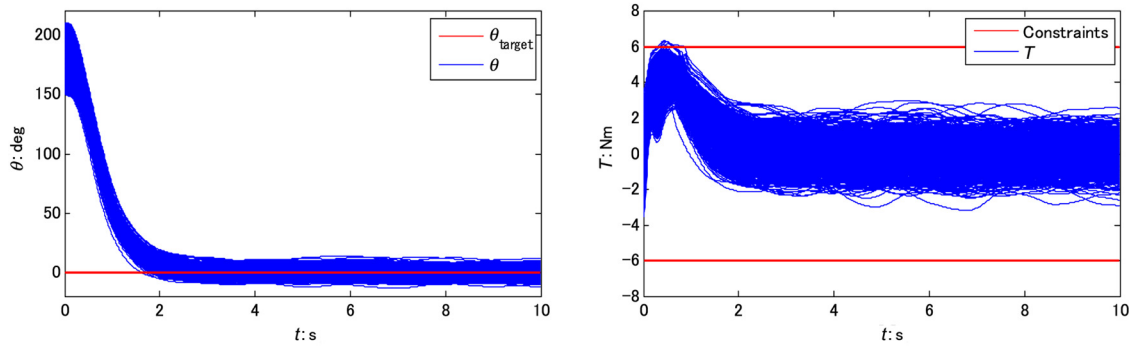
Fig. 11 Comparison of convergence history of GAs.



a) Conventional GA



b) VE-GA with constant era



c) VE-GA with variable era

Fig. 12 MC results: comparison of training NNs of gas.

simulation; the fitness J for the nominal case worsens. This implies that the optimum solution of the conventional GA is not robust. The torque often violates the constraint for both the nominal case as well as for other environment settings. Therefore, the standard deviation was worse and the robust evaluation value J_{rob} worsened. Case b yielded a more robust evaluation value, although the angle θ did not converge to zero. The objective of the control was not achieved. In contrast, in case c, the control objective was achieved. Although the torque was occasionally only slightly greater than the constraint, it can be resolved by modifying the settings of the penalty function.

Table 1 Comparison of MC results: swing-up of inverted pendulum.

GAs	J (nominal case)	Average J_{ave}	Variance σ_J^2	Robustness evaluation J_{rob}
Conventional	1.000	7.567	497.706	20.460
Constant era	1.246	1.255	0.312	1.578
Variable era	0.924	1.029	0.054	1.163

VIII. Transitional Flights of Tail-Sitter VTOL Mini UAV

A. Preparation

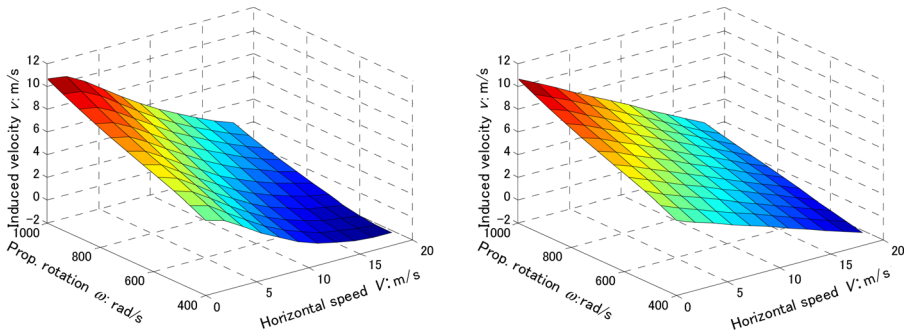
The flight speed V was selected as the scheduling parameter. Three trimmed "level" flights were selected as sampling points for the LTI models. They are as follows: hovering ($V = 0\,\text{m/s}$), low-speed forward flight ($V = 4\,\text{m/s}$), and high-speed forward flight ($V = 18\,\text{m/s}$). An LPV model of the tail-sitter was constructed by interpolating these LTI models.

In the case of the tail-sitter dynamics, the induced velocity of the propeller is very important for evaluating the effective angles of attack of the immersed wing. However, this value is not a state variable of the LPV model. Therefore, an additional induced velocity model has to be constructed to evaluate the value in the LPV simulations. In the current paper, the induced velocity is modeled using linear interpolation with propeller rotation speed ω and flight speed V. The maps of the induced velocity model are shown in Fig. 13. The induced velocities at the four corner points were used as sample data. For example, at the corner point $V = 20\,\text{m/s}$ and $\omega = 400\,\text{rad/s}$, the induced velocity was calculated under the following condition: the vehicle attitude was set to 20 m/s trimmed level flight and then the propeller rotation speed was set to $\omega = 400\,\text{rad/s}$.

The NN used as the reference command generator has four, four, and two neurons for the input, hidden, and output layers, respectively; 30 parameters are to be optimized. The inputs x and outputs r are as follows

$$x^T = \begin{bmatrix} \dot{H} & V & \theta & \omega \end{bmatrix}, \quad r^T = \begin{bmatrix} \theta_c & \omega_c \end{bmatrix} \tag{34}$$

In this problem, model uncertainties, initial state errors from the trim conditions, and a random gust were considered in the calculation as the "environment." The model uncertainties are variations of the system matrices and initial conditions settings. It is assumed that every element of the system matrix has a variation range of ±30%. The initial condition settings have variation ranges as follows: q_i, $\pm 0.2\,\text{rad/s}$; θ_i, $\pm 0.2\,\text{rad}$; and ω_i, $\pm 10\,\text{rad/s}$. With regard to the gusts, the horizontal component has zero average and 0.8 m/s standard deviation, and the vertical component has zero average and 0.56 m/s standard deviation, which is a reasonable gust condition for mini UAV flights. The time series data of the gust were constructed from white noise through a second-order low pass filter with a cut-off frequency of $\omega_f = 1\,\text{rad/s}$.



a) Original nonlinear model b) Interpolated model

Fig. 13 Induced velocity model.

B. Outbound Transitions

The outbound transition is the transitional flight from hovering to cruising. The fitness function to be minimized is given as

$$J = w_1 \int_0^{t_f} |V_{\text{tar}} - V|^{\frac{1}{2}} \, \mathrm{d}t + w_2 \int_0^{t_f} |\dot{H}| \, \mathrm{d}t + \int_0^{t_f} (w_3 \dot{\theta}_c^2 + w_4 \dot{\omega}_c^2) \mathrm{d}t + w_5 P_{\alpha_s} + w_6 P_{\text{NN}} \tag{35}$$

The first term imposes the vehicle to change the flight speed to the target flight speed V_{tar}. With regard to the outbound transition, this is a cruising speed of $V_{\text{tar}} = 16\,\text{m/s}$. The second term is for reducing the variations in the altitude. The third term is for avoiding rapid changes in the output of the NN. The fourth tem $P_{\alpha s}$ is a penalty function to constrain the angle of attack to an interval $[\alpha_{s\,\min}, \alpha_{s\,\max}]$, which is represented as

$$P_{\alpha_s} = |\max(0, \quad \alpha_s - \alpha_{s\,\max})| + |\min(0, \quad \alpha_s - \alpha_{s\,\min})| \tag{36}$$

Here, $[\alpha_{s\,\min}, \alpha_{s\,\max}] = [-15, +15]$ in degrees. It is very important to maintain the flight conditions within the controllable region. If the angle of attack becomes too high, the lateral control margins decrease because the aileron deflections cause main wing stall, that is, an uncontrollable flight condition. In this problem, the weight factors of Eq. (35) are set as follows

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 10^{-4} & 10^{-4} & 10^3 & 10^3 \end{bmatrix} \tag{37}$$

Histories of the robustness evaluation J_{rob} in the iterations are shown in Fig. 14 for the following three optimization cases: a) conventional GA, b) VE-GA with constant era ($E = 10$), and c) VE-GA with variable era ($E_1 = 1$, $E_2 = 10000$, $E_{\max} = 100$). The MC (nonlinear simulation) results using the obtained NN command generator are shown in Figs. 15 and 16. Figure 15 shows the results using the conventional GA, and Fig. 16 shows those of the proposed GA. Additionally, a quantitative comparison is shown in Table 2.

The vehicle can transition from hovering to cruising with small altitude variations and there are large stall margins for α_s. Outbound transitions are relatively easy, that is, the control results are only slightly affected by uncertainties or disturbances. As outbound transitions are accelerative flights, the throttle settings are higher than those in trimmed flight and decelerating flight. Therefore, the propeller slipstreams strengthen, and the effective angles of attack α_s decrease. In this manner, the main wing does not stall even in low-speed flight in gusty wind conditions. Therefore, there are very small differences in the results obtained using the conventional GA and the proposed one.

C. Inbound Transitions

The inbound transition is the transitional flight from cruising to hovering. Here, the following fitness function was used

$$J = w_1 |V_{\text{tar}} - V(t_f)| + w_2 \int_0^{t_f} \dot{H}^2 \, \mathrm{d}t + \int_0^{t_f} (w_3 \dot{\theta}_c^2 + w_4 \dot{\omega}_c^2) \mathrm{d}t + w_5 P_{\alpha_s} + w_6 P_{\text{NN}} \tag{38}$$

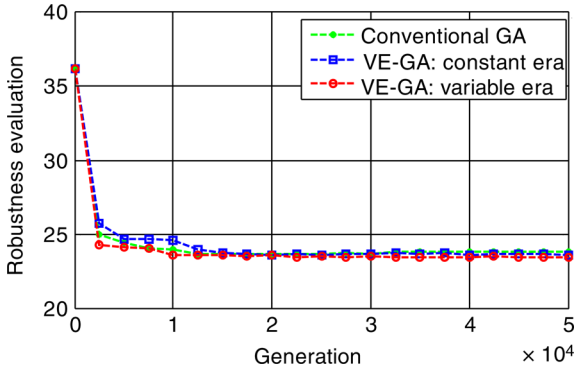


Fig. 14 Outbound transitions: convergence history of GAs.

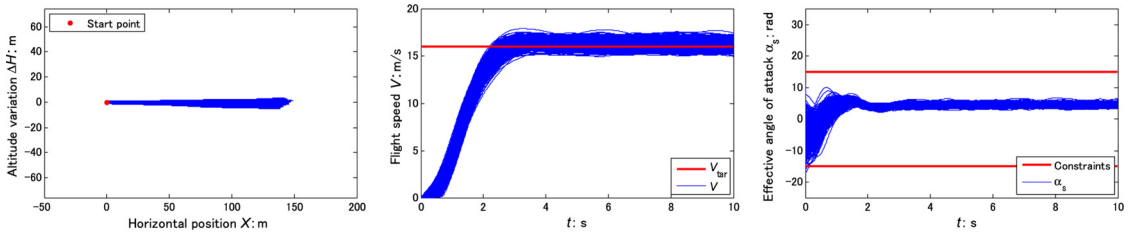


Fig. 15 Outbound transitions: results of MC using NN trained with conventional GA.

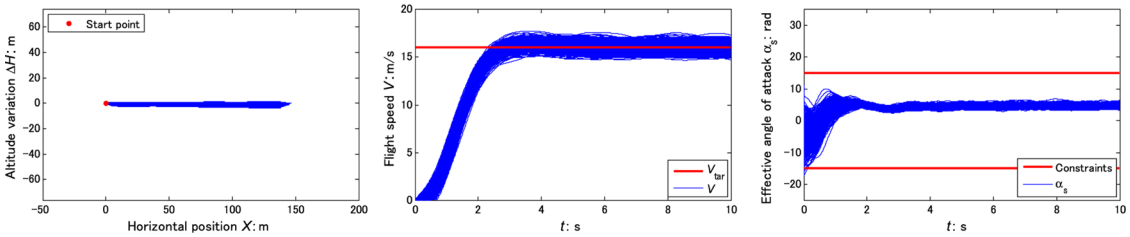


Fig. 16 Outbound transitions: results of MC using NN trained with proposed VE-GA.

Table 2 Comparison of MC results: outbound transition.

GAs	J (nominal case)	Average J_{ave}	Variance σ_J^2	Robustness evaluation J_{rob}
Conventional	1.000	1.164	0.147	1.272
Constant era	0.974	1.123	0.086	1.205
Variable era	0.871	1.092	0.076	1.169

where the target flight speed is $V_{\text{tar}} = 0\,\text{m/s}$, the angle of attack constraints are set as $[\alpha_{s\,\min}, \alpha_{s\,\max}] = [-15, +15]$ in degrees, and the weight factors are set as follows

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} = \begin{bmatrix} 1 & 10^{-3} & 10^{-6} & 10^{-6} & 10^{3} & 10^{3} \end{bmatrix} \tag{39}$$

The histories of the robustness evaluation J_{rob} over the iterations are shown in Fig. 17 for the following three optimization cases: a) conventional GA, b) VE-GA with constant era ($E = 10$), and c) VE-GA with variable era ($E_1 = 1$, $E_2 = 30000$, $E_{\max} = 100$). The MC results are shown in Figs. 18 and 19 and a quantitative comparison with the conventional one is shown in Table 3. Inbound transitions are relatively difficult. As such transitions are decelerative flights, the throttle settings become lower than those in trimmed and accelerative flight. The propeller slipstreams weaken, and the effective angles of attack increase. Hence, stall margins decrease. Therefore, the flights easily violate the stall constraint caused by the model uncertainties and disturbances. Therefore, the flights controlled by an NN trained using the conventional GA often violate the stall constraint. However, flights controlled by an NN trained using the proposed VE-GA only marginally violate the constraint. Additionally, even for the nominal condition, the fitness function improves more than the conventional one; this is caused by the difference between the LPV model used in the GAs and the nonlinear model used in the evaluation MC simulations.

D. Inbound Transition using Leading Edge Slats

The abovementioned inbound transitions are difficult flights. A climbing flight was required to avoid stalls. An inbound transition is usually conducted immediately before a landing. Therefore, large climbing maneuvers are not preferable. Leading edge slats are used to solve this problem. The angle of attack constraints are now set to $[\alpha_{s\,\min}, \alpha_{s\,\max}] = [-15, +25]$ in degrees, and the weight factors are set as follows

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} = \begin{bmatrix} 1 & 5 \times 10^{-2} & 10^{-6} & 10^{-6} & 10^{4} & 10^{3} \end{bmatrix} \tag{40}$$

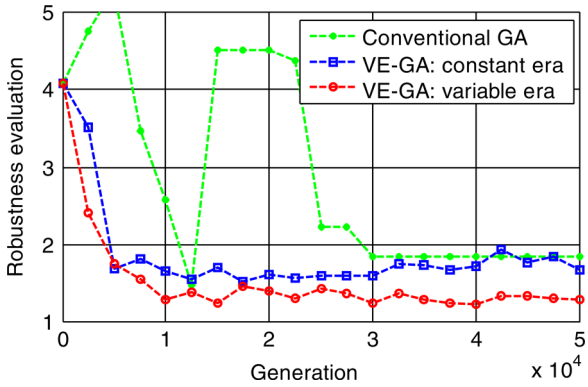


Fig. 17 Inbound transitions: convergence history of GAs.

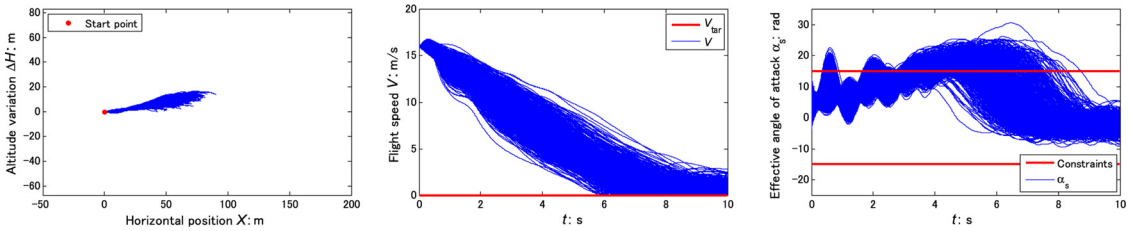


Fig. 18 Inbound transitions: results of MC using NN trained with conventional GA.

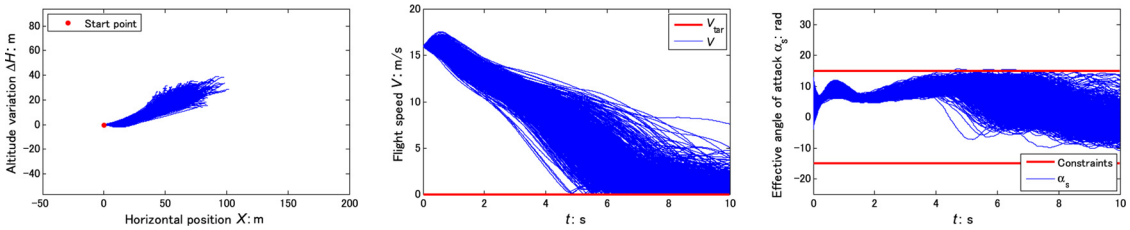


Fig. 19 Inbound transitions: results of MC using NN trained with proposed VE-GA.

Table 3 Comparison of MC results: inbound transition.

GAs	J (nominal case)	Average J_{ave}	Variance σ_J^2	Robustness evaluation J_{rob}
Conventional	1.000	2.206	29.826	4.805
Constant era	0.201	0.253	0.055	0.365
Variable era	0.108	0.206	0.031	0.289

The histories of the robustness evaluation J_{rob} over the iterations are shown in Fig. 20 for the following three optimization cases: a) conventional GA, b) VE-GA with constant era ($E = 10$), and c) VE-GA with variable era ($E_1 = 1$, $E_2 = 1000$, $E_{\max} = 100$). The MC results obtained using a robust NN controller are shown in Figs. 21 and 22, and a comparison with the conventional one are shown in Table 4. Although the altitude variations were sometimes approximately 40 m in the case without slats, the variations are reduced to less than 20 m in the case with slats.

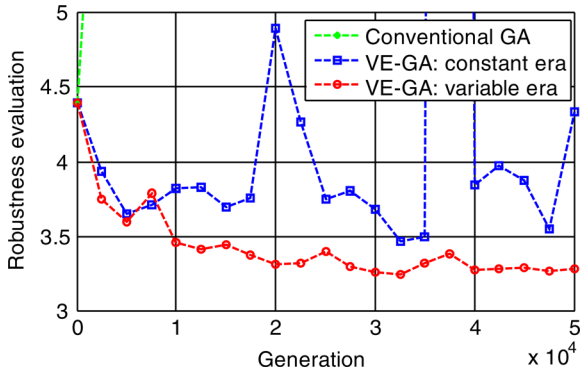


Fig. 20 Inbound transitions: convergence history of GAs.

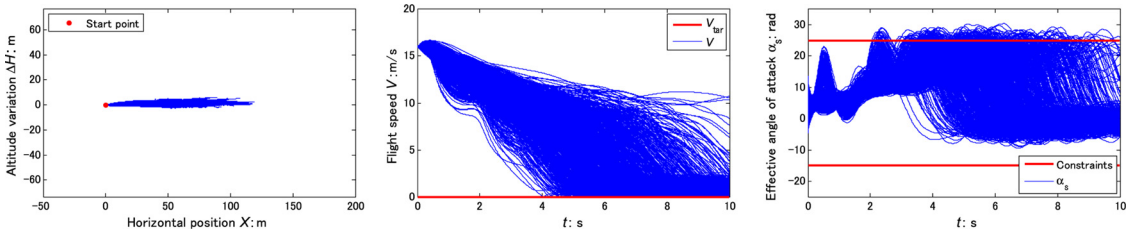


Fig. 21 Inbound transitions with slats: results of MC using NN trained with conventional GA.

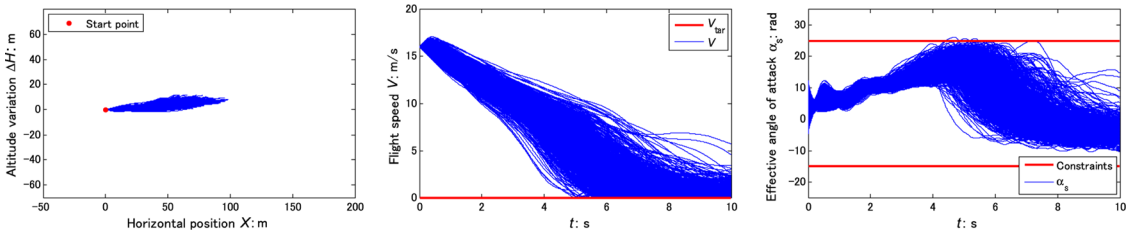


Fig. 22 Inbound transitions with slats: results of MC using NN trained with proposed VE-GA.

Table 4 Comparison of MC results: inbound transition without slats.

GAs	J (nominal case)	Average J_{ave}	Variance σ_J^2	Robustness evaluation J_{rob}
Conventional	1.000	2.627	18.828	6.982
Constant era	1.798	1.968	0.052	2.197
Variable era	0.969	1.306	0.124	1.660

IX. Conclusion and Future Work

A design method for robust controllers for a tail-sitter VTOL mini UAV was presented. To solve the problem of wide operating range and nonlinearity, a GS-LQR was used as a pitch attitude controller. An offline trained NN was used as a reference command generator to achieve control objectives such as transitional flight. A VE-GA was used to improve the robustness of the NN. Here, the word "environment" implies the model errors and disturbance settings considered in the evaluation simulations. The environment was changed repeatedly after several generations.

Although a constant era was used in an existing study, the introduction of a variable era and the update law was proposed in this paper. The method was applied to the example problem of the swing-up control of an inverted pendulum, and its effectiveness was verified.

Finally, the method was applied to transitional flight control problems. In the case of outbound transition, there was no improvement in the results because proper control was easily achieved by the conventional method. However, the new algorithm was found to be superior for inbound transitions and robust controllers for the transition were obtained. The robust optimization method proposed in this paper is a general methodology and therefore it can be applied to a variety of robust optimization problems such as the optimization of a function with uncertainty, as described in this paper.

Although the effectiveness of the method was verified, certain problem continue to exist. The method can create genes close to a robust optimal solution, but it is not asymptotic in convergence. Therefore, only chance decides the quality and accuracy of a solution. This problem may be solved by combining the proposed system with a local optimization method. Although only numerical simulations were used in this paper, it is necessary to conduct real flight tests for the evaluation.

Acknowledgments

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

References

[1] United States Department of Defense, "*Unmanned Aircraft Systems Roadmap 2005–2030*" , [online] document, http://www.acq.osd.mil/usd/Roadmap%20Final2.pdf [retrieved August 2005].

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

[3] Kubo, D., Suzuki, S., and Kagami, T., "Development of Experimental Small UAV Equipped with Cellular Phone Data Link System," *Proceedings of 25th Congress of International Council of Aeronautical Sciences*, International Council of the Aeronautical Sciences, Hamburg, Germany, 2006, ICAS-2006-1.7.3.

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

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

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

[7] Howard, B. M. and Kaminer, I., "Survey of Unmanned Air Vehicles," *Proceedings of the American Control Conference*, Seattle, Washington, 1995, pp. 2950–2953, FA1-9:35

[8] Stone, H., and Clarke, G., "Optimization of Transition Maneuvers for a Tail-Sitter Unmanned Air Vehicle (UAV)," *Australian International Aerospace Congress*, Canberra, 2001, Paper 105.

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

[10] Wu, A. D., Turbe, M. A., Kannan, S. K., Neidhoefer, J. C., and Johnson, E. N., "Flight Results of Autonomous Airplane Transitions to and from Vertical Hover," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2006, AIAA Paper 2006-6775.

[11] Anderson, P., and Stone, H., "Predictive Guidance and Control for a Tail-Sitting Unmanned Aerial Vehicle," *Proceedings of Information, Decision and Control 2007*, IEEE, Adelaide, Australia, 2007, pp. 148–153.

[12] Shamma, J. S., and Athans, M., "Gain Scheduling: Possible Hazards and Potential Remedies," *IEEE Control Systems Magazine*, Vol. 12, No. 3, 1992, pp. 101–107.
doi: 10.1109/37.165527

[13] 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, 2005, pp. 96–105.

[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*, JSASS (Japan Society for Aeronautical and Space Sciences), Yokohama, Japan, 2004, pp. 538–542 (in Japanese).

[15] 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*, Morgan Kaufmann, 1997, pp. 246–253.

[16] Kita, H., Ono, I., and Kobayashi, S., "Theoretical Analysis of the Unimodal Normal Distribution Crossover for Real-coded Genetic Algorithms," *Proceeding of International Conference on Evolutionary Computation*, IEEE, 1998, pp. 529–534.

[17] Mckay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output of a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245. doi: 10.2307/1268522

Ella Atkins
Associate Editor