

# OPENVSP BASED AERODYNAMIC DESIGN OPTIMIZATION TOOL BUILDING METHOD AND ITS APPLICATION TO TAILLESS UAV

Ju-Hoe Kim<sup>1</sup> & Takeshi Tsuchiya<sup>1</sup>

<sup>1</sup>The University of Tokyo

## Abstract

This paper presents an aerodynamic design optimization tool for aerial vehicles at the conceptual design phase and its detailed building method. An application example of the presented tool is also given. The proposed tool consists of OpenVSP, open-source software for aircraft conceptual design, and an optimization algorithm. After an overview of OpenVSP is given, user-friendly integration methods of OpenVSP and the optimization algorithm are described. Evolutionary computation is applied as the optimization algorithm to achieve robust optimization. Induced drag minimization problem and tailless UAV design problem are solved by the proposed method to validate its performance. As a result of solving the induced drag minimization problem of the wing, it was confirmed that the optimized wing has a uniform local lift distribution in the span direction. The optimization results agree with the theory. After optimizing the main wing section of tailless UAV with 1.6 m span length, the optimized UAV has enough lift and good longitudinal static stability under its cruise conditions. An experimental tailless UAV has been manufactured, and a flight test has been conducted to demonstrate the effectiveness of the proposed method for real application.

**Keywords:** unmanned aerial vehicle, optimization, OpenVSP, tailless aircraft, evolutionary computation

## 1. Introduction

New aerial vehicles such as air taxis and delivery drones have continued to attract attention. It is essential to improve their aerodynamic performance to increase their payload and flight range. The importance of introducing the aerodynamic design into the conceptual design of new aerial vehicles has been pointed out [1]. The recent emergence of fixed-wing and rotorcraft hybrid delivery drones shows that some companies are beginning to focus on aerodynamic design [2]. For today's conventional aircraft aerodynamic design, CFD is widely used, and it is also possible to combine CFD with optimization algorithms to design new conceptual aircraft. However, high-fidelity CFD software's operational and building environment cost is generally expensive and they need sophisticated fine-tuning of mesh files. Without skilled CFD engineers, it is not easy to implement it in the new aerial vehicle conceptual design. Providing a simple and stable aerodynamic design tool is essential for engineering to support the new aerial vehicle industry mentioned above. As a CFD with low computational cost, potential flow software such as PanAir [3], Tornado [4], and Xflr5 [5] are sometimes used in the conceptual design phase. In this research, OpenVSP, a free aircraft conceptual design tool, is used. OpenVSP has the advantage over other potential flow software in that it is easy to connect with other software due to its strong API and the simplicity of parameter expression of the vehicle geometry.

This paper provides a simple, user-friendly, and small construction cost aerodynamic design tool that can be applied to the conceptual design of the new aerial vehicle and conventional aircraft. The proposed design tool comprises OpenVSP and an optimization algorithm.

## 2. About OpenVSP

OpenVSP is a suite of tools that provides a parametric aerial vehicle design environment. On January 10, 2012, OpenVSP was released as an open-source project under the NASA Open-Source Agreement (NOSA) version 1.3 [6]. OpenVSP is still being updated as an open-source project [7], and the latest version, OpenVSP 3.28.0 has been released on May 31, 2022 [6]. The main functions of OpenVSP are called Vehicle Sketch Pad (VSP), which defines the aerial vehicle geometry, and VSPAERO, which performs aerodynamic analysis of the defined geometry with a potential flow solver.

### 2.1 VSP

VSP is a powerful tool since it allows users to define the aerial vehicle geometry by specifying simple and intuitive parameters for engineers. VSP also includes a simple analysis tool for the defined aerial vehicle geometry, which enables rough analysis of volume, surface area, mass, parasite drag, etc. The aerial vehicle defined by VSP can be output in some data formats used by general CAD software such as STL. It can also export standard mesh forms for FEM or high-fidelity CFD analysis.

### 2.2 VSPAERO

VSPAERO provides two types of potential flow solvers, the panel method and the Vortex Lattice Method (VLM). However, only the VLM can analyze supersonic flow. On the other hand, the panel method can more accurately capture the fluid interaction between components and handle the effects of model thickness and volume. Only the panel method can separate the pressure acting on the top and bottom surfaces of the model, while the VLM can only obtain the difference between them. A common point to note in VSPAERO's aerodynamic analysis tools is that they use the component type information to determine the wake placement, and only wing and propeller receive wakes. This means that some components' lift and induced drag cannot be handled, but it is possible to handle moments. To verify the performance of VSPAERO, comparisons between the theoretical performance of simple airfoil shapes [8] and the NASA EET wind tunnel test results [9] have been reported. These results show that VSPAERO is accurate enough to estimate lift and moments generated by the aerial vehicle at the conceptual design stage. As for estimations of drag, since lift estimations are reasonable, the method of adding parasite drag based on the empirical formula [10] to the induced drag obtained by VSPAERO is effective. At the early-stage design, the VLM is recommended because it is considered more stable and faster than the panel method [11]. Therefore, aerodynamic analysis using the VLM is assumed for this proposed tool.

One of the features of OpenVSP is that it requires less effort from users to convert the aerial vehicle geometry to the aerodynamics analysis mesh file. This feature makes OpenVSP suitable for aerodynamics design optimization.

## 3. Optimization Method

Evolutionary Computation (EC) does not use gradient information for optimization so that its convergence is slow, but stable optimization can be performed. In addition, since a global search is performed, much data is generated in the optimization process, and design knowledge can be extracted through post-processing analysis. These features make it suitable for conceptual design tools. In general, ECs are used in conjunction with unconstrained optimization problems, so it is necessary to introduce constraint handling methods to solve typical engineering problems. This chapter describes the optimization algorithm and constraint handling method used in the proposed design tool.

### 3.1 Optimization Algorithm

Differential Evolution (DE) [13] is one of the most powerful ECs because it can be applied to many problems and shows high search performance. However, its search performance is often affected by the parameter settings. Recently developed DEs are self-adaptive algorithms, which automatically adjust their parameters during run time. One such algorithm is SHADE, proposed by Tanabe [14]. SHADE requires only three parameters from the user: the maximum number of iterations  $I_{\max}$ , the population size  $N_{\text{pop}}$ , and the memory for storing valid past internal parameters  $H$ . SHADE shows

better results than other self-adaptive DEs on benchmark problems. The implementation of SHADE has been used herein in Tanabe's doctoral thesis [14].

### 3.2 Constraints Handling Method

In the proposed aerodynamic design tool, the  $\varepsilon$  constrained method by Takahama et al. [15] is introduced. Consider the following general optimization problem.

$$\begin{cases} \text{find} & \mathbf{x} \\ \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & g_j(\mathbf{x}) \leq 0, j = 1, \dots, q \\ & h_j(\mathbf{x}) = 0, j = q + 1, \dots, m \\ & l_i \leq x_i \leq u_i, i = 1, \dots, n \end{cases} \quad (1)$$

In this method, constraint violation  $\phi(\mathbf{x})$  is defined as follows.

$$\phi(\mathbf{x}) = \sum_{j=1}^q |\max\{0, g_j(\mathbf{x})\}| + \sum_{j=q+1}^m |h_j(\mathbf{x})| \quad (2)$$

$\varepsilon$  constrained method compares two solution candidates  $\mathbf{x}_1, \mathbf{x}_2$  by the following  $\varepsilon$  level comparison.

$$\mathbf{x}_1 <_{\varepsilon} \mathbf{x}_2 \Leftrightarrow \begin{cases} f(\mathbf{x}_1) < f(\mathbf{x}_2), \text{ if } \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \leq \varepsilon \\ f(\mathbf{x}_1) < f(\mathbf{x}_2), \text{ if } \phi(\mathbf{x}_1) = \phi(\mathbf{x}_2) \\ \phi(\mathbf{x}_1) < \phi(\mathbf{x}_2), \text{ otherwise} \end{cases} \quad (3)$$

$$\mathbf{x}_1 \leq_{\varepsilon} \mathbf{x}_2 \Leftrightarrow \begin{cases} f(\mathbf{x}_1) \leq f(\mathbf{x}_2), \text{ if } \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \leq \varepsilon \\ f(\mathbf{x}_1) \leq f(\mathbf{x}_2), \text{ if } \phi(\mathbf{x}_1) = \phi(\mathbf{x}_2) \\ \phi(\mathbf{x}_1) < \phi(\mathbf{x}_2), \text{ otherwise} \end{cases} \quad (4)$$

The initial epsilon value  $\varepsilon(0)$  is selected from the constraint violation of individuals with a specific rank. This rank is based on ascending order of constraint violations in the initial population.

$$\varepsilon(t) = \begin{cases} \varepsilon(0) \left(1 - \frac{t}{T_c}\right)^{cp}, & 0 < t < T_c \\ 0, & t \geq T_c \end{cases} \quad (5)$$

$\varepsilon$  update parameter  $cp, T_c$ , and a specific rank for initial  $\varepsilon$  are the parameters of this method. The epsilon constrained method has a few setting parameters, but it can handle constrained problems with relatively good stability.

## 4. Integration of Optimization Method and OpenVSP

OpenVSP provides several APIs to automate processes using MATLAB, Python, and AngelScript. In using numerical calculation environments such as MATLAB or Python, environment and software version management is required. Unfortunately, the automation environment construction is not easy. AngelScript is a flexible cross-platform library[16]. A text file that describes the process to be executed by OpenVSP in AngelScript is called a vspscript and saved with a .vspscript filename extension. OpenVSP is usable via GUI and Command Line Interpreter (CLI) through vspscript. In this way, OpenVSP and the numerical computing environment are connected via CLI and vspscript. This method does not require version management or advanced knowledge of some implementation. For this reason, this method is considered straightforward and user-friendly.

### 4.1 OpenVSP control from MATLAB

This section provides an overview of OpenVSP and the optimization algorithm integration method. The optimization algorithm described in Chapter 3 is implemented in MATLAB 2020a. Also, cmd.exe is used as CLI and the operating system is Windows 10. The version of OpenVSP is 3.25.0.

**STEP 1** User creates .vspscript file to update aerodynamic analysis mesh file in VSP. For creating a vspscript, online documentation [18] and the sample collections included in OpenVSP in the “script” folder are useful.

**STEP 2** User creates three text files (.vsp3, .des, and .vspaero) with the OpenVSP GUI. The .vsp3 defines baseline aerial vehicle geometry, the .des defines design variables, and the .vspaero defines aerodynamic analysis solver settings. Passing the .des file to VSP from the CLI, the vehicle geometry can be updated to reflect the design variables. Therefore, optimization can be performed by rewriting the design variable parameters in the .des file from the numerical calculation environment. For more information on .des files, please refer to the documents published in the past OpenVSP workshops [17].

**STEP 3** Optimization algorithm rewrites the design variables in the .des file.

**STEP 4** Optimization algorithm operates the CLI to start vsp.exe with .des and .vspscript. When VSP is launched from the CLI with vspscript, it does not open the GUI window and exits itself after executing instructions written in vspscript. The following is an example command.

->vsp example.vsp3 -des example.des -script example.vspscript

The aforementioned command can be executed from MATLAB by using a “system function”.

**STEP 5** After VSP finishes running, it will create text files about the analysis results and an updated mesh file for VSPAERO. Analysis result files can be imported into the numerical calculation environment if necessary.

**STEP 6** Optimization algorithm rewrites the .vspaero file if solver parameters vary each iteration.

**STEP 7** Optimization algorithm enters the command in the CLI to run VSPAERO. VSPAERO exits itself after finishing the analysis. The following is an example command.

->vspaero example

**STEP 8** After VSPAERO finishes running, it will create text files containing data from the analysis, and these files will be imported into the numerical calculation environment.

**STEP 9** Using the imported results, the optimization algorithm performs additional analysis and evaluation and returns to step 3.

Figure 1 illustrates the workflow discussed above.

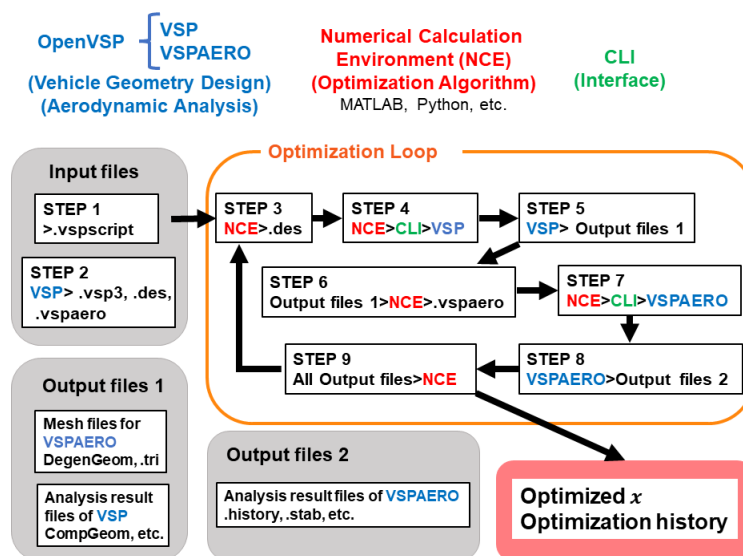


Figure 1 – Workflow overview.

## 5. Application to induced drag minimization

In this chapter, an aerodynamic optimization problem is solved to validate the proposed method. It is known that the induced drag associated with lift is smaller when the spanwise lift distribution of a wing is elliptical. For a wing with no twist and constant span length, this condition can be achieved by making the spanwise chord proportional to the elliptical distribution [19]. Therefore, this induced drag minimization problem is a useful benchmark problem for the validation of aerodynamic design tools.

### 5.1 Problem definition

The induced drag minimization problem is defined in following optimization problem form.

$$\left\{ \begin{array}{ll} \text{find} & \mathbf{x} = (C_{\text{root}}, C_{\text{tip}}, \beta_{\text{bz}}, M_{\text{bz}}) \\ \text{minimize} & C_{\text{Di}} \\ \text{subject to} & |C_L - C_{L,\text{ref}}| \leq 0.001 \\ & 0.05 \leq C_{\text{root}} \leq 2 \text{ m} \\ & 0.05 \leq C_{\text{tip}} \leq 2 \text{ m} \\ & 0 \leq \beta_{\text{bz}} \leq 60 \text{ deg} \\ & 0.5 \leq M_{\text{bz}} \leq 1.5 \end{array} \right. \quad (6)$$

The constraint about lift ensures that the designed wing will have the same amount of lift as the baseline wing. Since the perfect agreement is difficult to achieve due to numerical errors, inequality constraints are used. The baseline wing is rectangular shape with a NACA 0012 airfoil [20]. Its chord length is 1 m and span length is 10 m. The design point flow conditions, reference parameters, and the results of the VSPAERO analysis about baseline are shown in Table 1. These flow conditions and reference parameters are fixed through optimization.

Table 1 – Flow analysis settings and results about the baseline

$AoA$	$M$	$Re_{\text{cref}}$	$S_{\text{ref}}$	$B_{\text{ref}}$	$C_{\text{ref}}$	$C_{L,\text{ref}}$	$C_{\text{Di}}$
10	0.15	$6 \cdot 10^6$	10	10	1	0.8571	0.0238

VSP has a “blending function” for defining the leading edge and trailing edge with Bézier curves. The two design variables  $\beta_{\text{bz}}$  and  $M_{\text{bz}}$  control the Bézier curve. For the leading edge,  $M_{\text{bz}}$  is the magnitude of the tangent vector at the starting point of the leading edge of the wing tip, and  $\beta_{\text{bz}}$  is its angle with the y-axis. Figure 2 illustrates the meaning of the design variables. Another parameterization method is to discretize the spanwise chord distribution. This method increases the number of design variables. Also, VSP is difficult to automate when a component has too many design variables because it requires fine tuning of the mesh file.

### 5.2 Algorithm setting

Table 5 shows the parameter settings of the optimization algorithm. The optimization was run 10 times.

Table 2 – Algorithm parameter settings.

$I_{\text{max}}$	$N_{\text{pop}}$	$H$	$\text{ranking}$	$cp$	$T_c$
100	30	10	0.2	4	90

### 5.3 Results

Figure 3 shows the convergence of the optimization. It can be seen that each case converges to the same value with the constraint satisfied.

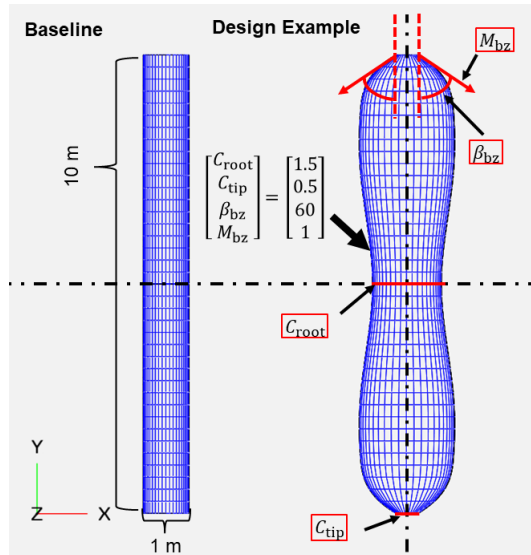


Figure 2 – Design variables overview.

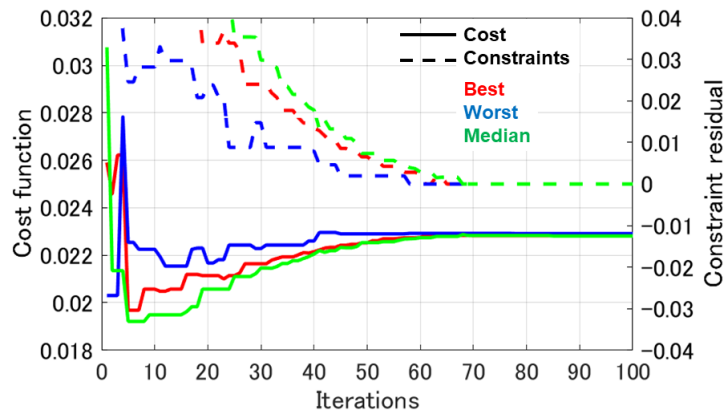


Figure 3 – Optimization history.

It is known that the section lift coefficient of the elliptical wing is uniform in spanwise [19]. Figure 4 shows that the optimized wings have spanwise section lift coefficients distribution close to the theory. Figure 5 shows the planforms of optimized wings. The black wing is manually adjusted elliptical wing created by controlling the discretized spanwise chord distribution. Optimized wings are close to the elliptical wing except for the wingtip. To obtain a more elliptical shape, changing the parameterization or tightening the constraints on lift can be considered.

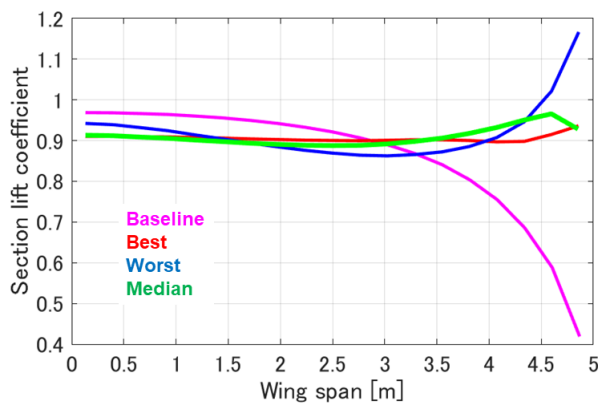


Figure 4 – Section lift coefficient distribution.

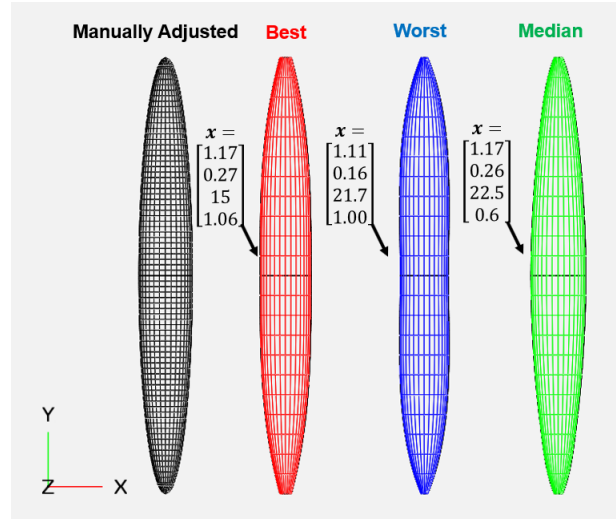


Figure 5 – Comparison of wing planforms.

The obtained statistics of the objective value,  $C_{Di}$ , are shown in Table 3. It can be seen that the optimized wings have smaller  $C_{Di}$  than the baseline. From these results, the proposed method can be applied to the aerodynamic optimization at the conceptual design phase.

 Table 3 – Statistics of obtained  $C_{Di}$ .

Baseline	Best	Worst	Median	SD
0.02380	0.02281	0.02294	0.02287	$4.6 \cdot 10^{-5}$

## 6. Application to tailless UAV

A tailless aircraft is one type of aircraft with the smallest components and is expected to have lightweight and high flight efficiency [10]. For UAV, the advantage of small components is the ease of mass production by reducing the manufacturing cost. There is a trade-off between flight efficiency and static stability of tailless aircraft, and computer-aided design is important because of the high degree of design difficulty. In this section, an example of a tailless aircraft design is presented as an application of the proposed aerodynamics optimization tool

### 6.1 Problem Definitions

As a design example, a radio-controlled tailless styrofoam UAV with 1.6 m span length and a 3-cell LiPo battery is considered. The cruise conditions are set to a visible altitude (less than 100 m) and a cruise speed of about 12 m/s. The angle of attack at the cruise point is set to 3 degrees. The design problem around the cruising point is defined as the following optimal design problem.

$$\left\{ \begin{array}{ll} \text{find} & \mathbf{x} = (\lambda, \beta, Twist) \\ \text{minimize} & -m_{\text{margin}} = \frac{mg - L_c}{g} \\ \text{subject to} & C_{m_{y,c}} \leq 0.0001 \\ & \frac{\partial C_{m_y}}{\partial \text{AoA}} < 0 \\ & 0.05 \leq SM_c \\ & mg - L_c \leq 0 \\ & 0.3 \leq \lambda \leq 0.6 \\ & 20 \leq \beta \leq 60 \\ & -10 \leq Twist \leq 0 \end{array} \right. \quad (7)$$

The design target is the main wing section, and the design variables are the taper ratio [-], sweep angle [deg], and twist angle [deg]. The airfoil at wing root is Eppler 328 [21] and the airfoil at wing tip is NACA 0012 [20]. The subscript c means cruise condition.

The objective function  $m_{margin}$  means mass margins for payloads and structure design. The constraints are related to the longitudinal static stability, which is important for manual operation.  $SM$  is static margin defined by following equation [10]

$$SM = -\frac{\partial C_{my}}{\partial A\alpha} / \frac{\partial C_L}{\partial A\alpha} \quad (8)$$

Figure 6 shows overview of this problem

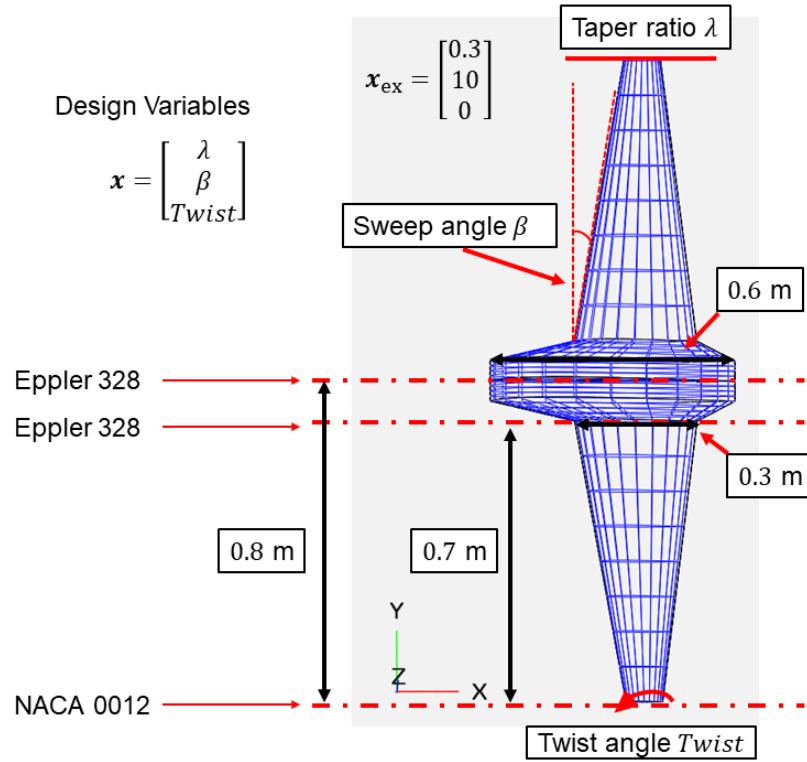


Figure 6 – Design problem overview.

The center of gravity (CG) is important when discussing the stability of the aircraft. The estimation function of VSP is used for wing's CG estimation. The CG of total system is estimated by Equation (9). Figure 7 shows the arrangement of electrical equipment.

$$\mathbf{CG} = \frac{\sum \mathbf{CG}_i \cdot m_i}{\sum m_i} \quad (9)$$



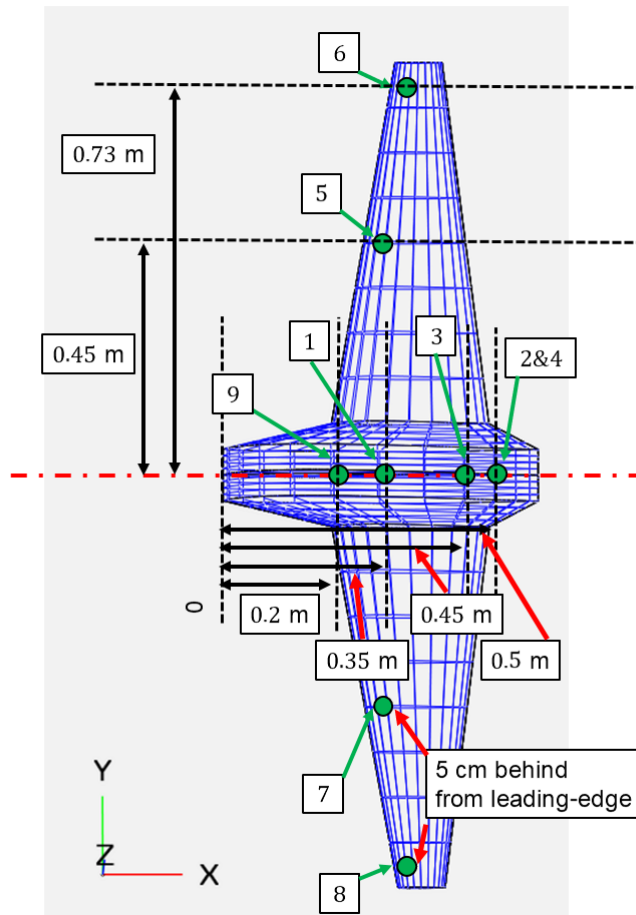


Figure 7 – Electric equipment arrangement

The mass of structure is estimated following equation (10)

$$m_{st} = \rho Vol (1 - ratio) \quad (10)$$

The styrofoam density  $\rho$  is  $30 \text{ kg/m}^3$ . The structure volume  $Vol$  is estimated by VSP. The lightening ratio  $ratio$  is 0.2.

Table 4 shows the mass of electrical equipment.

Table 4 – Mass list of electrical equipment

	Battery	Motor	ESC	Propeller	Servo	Receiver
Index at Figure 7	1	2	3	4	5,6,7,8	9
Mass [g]	200	50	20	30	10	10
Count	1	1	1	1	4	1

## 6.2 Algorithm Settings

Table 5 shows the parameter settings of the optimization algorithm.

Table 5 – Parameter settings.

$I_{max}$	$N_{pop}$	$H$	ranking	$cp$	$T_c$
100	20	10	0.2	4	95

### 6.3 Results

Figure 8 shows optimization history of objective function and sum of constraints.

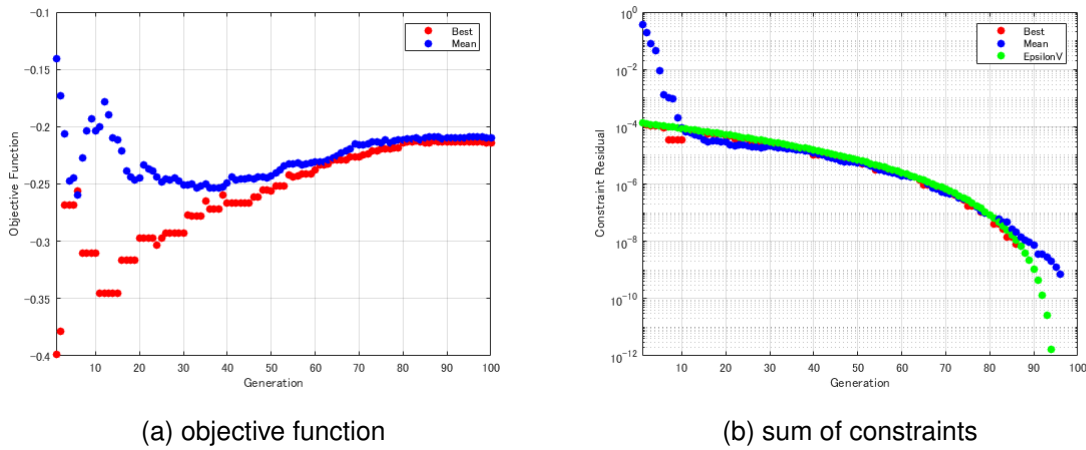


Figure 8 – Optimization history

From Figure 8, it can be seen that design variables with small objective function are selected as best in the early stages of optimization, and design variables with small constraint residuals are selected in the later stage. Figure 9 shows Optimized UAV.

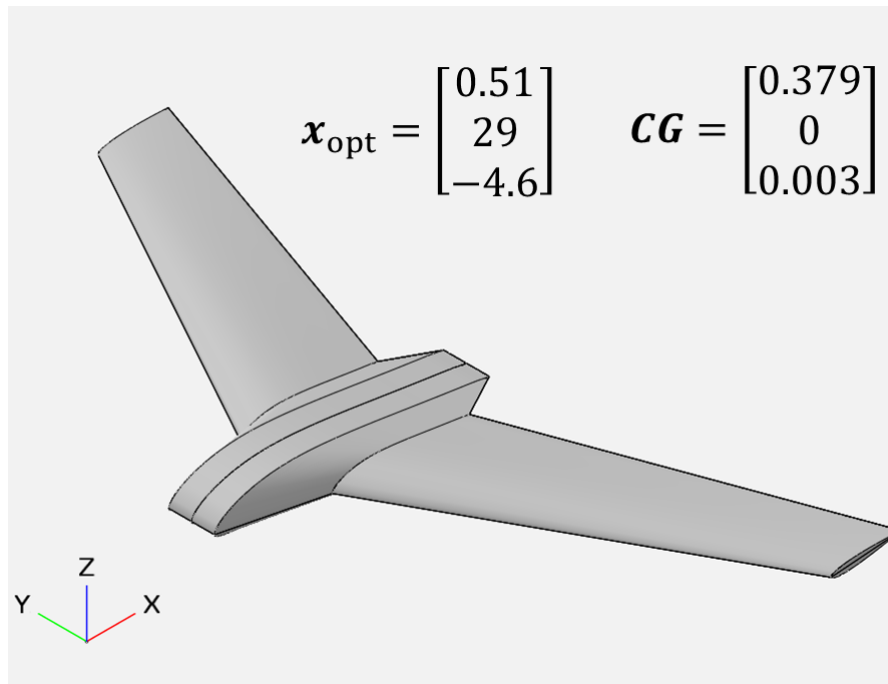


Figure 9 – Optimized UAV

Figure 10 shows aerodynamic coefficients of optimized UAV.

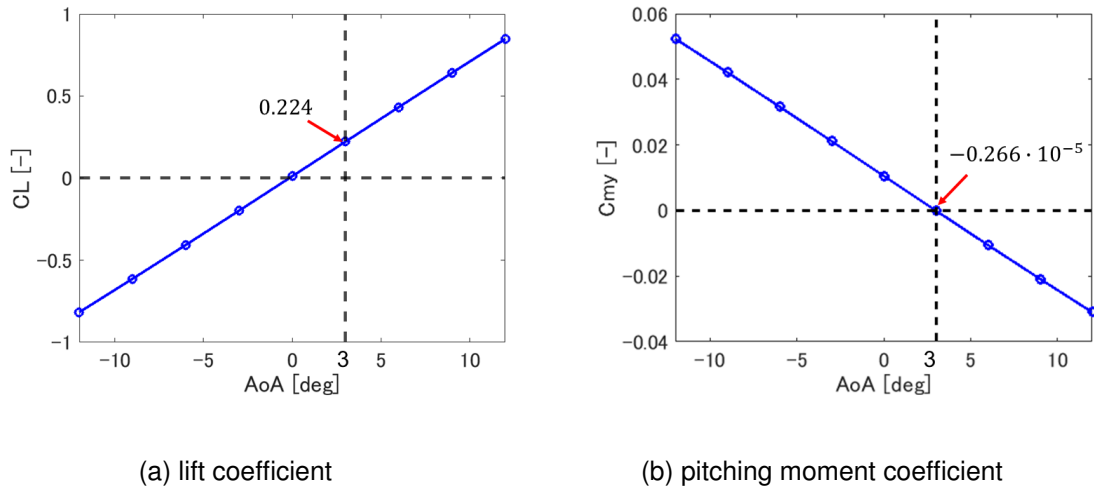


Figure 10 – Aerodynamic coefficients

Table 6 shows the results of optimization.

Table 6 – Results of optimization.

$\lambda$ [-]	$\beta$ [deg]	$Twist$ [deg]	$m_{margin,c}$ [kg]	$m$ [kg]	$C_{L,c}$ [-]	$C_{my,c}$ [-]	$SM_c$ [-]
0.51	29	-4.6	0.21	0.63	0.224	$-0.266 \cdot 10^{-5}$	5 %

The optimized UAV has 0.21 kg mass margins and its pitching moment at the cruise point is close to zero with 5 % static margin. These results indicate that the proposed aerodynamics design optimization tool is effective for this problem.

### 7. Optimized UAV manufacturing and first test flight

In building the experimental UAV based on optimization results obtained in section 6, it was not easy to reproduce the shape of the main wings by processing styrofoam because of their complex three-dimensional shape. Therefore, the center portion was manufactured with styrofoam, and the main wings were manufactured with balsa wood. Figure 11 shows the UAV in the process of manufacturing. Figure 11 shows the UAV in the process of manufacturing.



Figure 11 – Experimental UAV in the process of manufacturing

Materials of the UAV changed from the initial model, and its weight increased, so the electrical parts of the motor and battery were also changed to larger ones. The arrangement of electrical equipment

was changed from the initial model to match the UAV's CG with the optimization results. In addition, winglets were attached to the UAV for lateral-directional stability. Finally, the built UAV, including batteries, weighed 1.8 kg and its CG was 38 cm from the front nose tip. Due to the weight increase of the UAV, estimated cruising speed changed to about 17 m/s. As the CG position and wing shape are close to the optimization results, it is assumed that the performance of the experimental aircraft and the optimization results is close with regard to longitudinal static stability. Figure 12 shows manufactured UAV. Figure 12 shows manufactured UAV.



Figure 12 – Manufactured UAV

On the first test flight, the UAV took off by hand-throwing, and after gaining altitude, it circled the airfield and landed by belly landing. Figure 13 shows the first test flight.



Figure 13 – First test flight of experiment UAV

The test pilot evaluated that the UAV had longitudinal static stability and could be controllable by human. The future task is to quantitatively evaluate the effectiveness of the proposed method by acquiring flight data and comparing it with the aerodynamic data obtained by the design optimization.

## 8. Conclusions

A simple and user-friendly aerodynamic design optimization tool has been proposed. The proposed tool consists of OpenVSP and an optimization algorithm via text files and CLI. The building method of this tool has been explained, and two design optimization problems have been solved. The first problem is induced drag minimization. The optimized wing is close to a shape in which the span direction distribution of chord lengths is proportional to the elliptical distribution. The span direction distribution of local lift coefficients is nearly uniform. These results agree with theory. The second problem is tailless UAV design problem. Optimized tailless UAV has enough lift and good longitudinal static stability under its cruise conditions. The experiment UAV has been manufactured based on optimization results and its test flight has been conducted. The first flight of the UAV is successful. Both theory and experiment have shown the effectiveness of the proposed method.

## 9. Acknowledgment

This work was supported by JST SPRING, Grant Number JPMJSP2108. The first test flight was supported by the Great Flying Club, one Japanese amateur RC aircraft club.

## 10. Contact Author Email Address

For further communication, the author's email address is shown as follows.

Ju-Hoe Kim: kim-juhoe0380@g.ecc.u-tokyo.ac.jp

Takeshi Tsuchiya: tsuchiya@mail.ecc.u-tokyo.ac.jp

## 11. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

## References

- [1] Yamada K, Uchiyama T, Kato H, and Ueno M. Investigation Research on Potential Needs for Future Performance of Unmanned Fixed-Wing Aircraft, JAXA-RM-20-003, 2021 (in Japanese)
- [2] WINGCOPTER HP, <https://wingcopter.com/>, (accessed November 22, 2021).
- [3] Derbyshire T, and Sidwell K.W. PAN AIR Summary Document, (Version 1.0). NASA Contractor Report 3250, 1982.
- [4] Melin T. Tornado, a vortex lattice MATLAB implementation for Linear Aerodynamic Wing applications, Masters thesis, Royal Institute of Technology (KTH), Sweden, December 2000.
- [5] xflr5 website, André, D. <http://www.xflr5.tech/xflr5.htm>, (accessed January 10, 2022).
- [6] OpenVSP official website, <http://openvsp.org/>, (accessed May 31, 2022).
- [7] Source cord of OpenVSP at Github, <https://github.com/OpenVSP>, (accessed November 22, 2021).
- [8] Payne L. VSPAERO Verification Testing, [http://openvsp.org/wiki/lib/exe/fetch.php?media=workshop17:payne-vspero\\_verification.pdf](http://openvsp.org/wiki/lib/exe/fetch.php?media=workshop17:payne-vspero_verification.pdf), (accessed November 22, 2021).
- [9] Kinney D. VSPAERO. . . What's New?, [http://openvsp.org/wiki/lib/exe/fetch.php?media=workshop21:2021\\_vspero\\_-\\_what\\_is\\_new.pdf](http://openvsp.org/wiki/lib/exe/fetch.php?media=workshop21:2021_vspero_-_what_is_new.pdf), (accessed November 22, 2021).
- [10] Raymer D. P. *Aircraft Design*. 6th Edition, AIAA Education series, Virginia, 2018, pp. 416–442, pp. 592-593 pp. 834-839.
- [11] OpenVSP Google group discussion about VLM vs Panel method. <https://groups.google.com/g/openvsp/c/CKhkiLJ919k/m/VOEXIm7AAQAJ>, (accessed November 22, 2021).
- [12] OpenVSP Google Group, <https://groups.google.com/g/openvsp>, (accessed November 22, 2021).

- [13] Storn R, and Price K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Global Optimiz.*, **11(4)** (1997), pp. 341—359.
- [14] Tanabe R. Studies of Adaptive Differential Evolution for Function Optimization Problems, Ph.D. Thesis, The University of Tokyo, 2016 (in Japanese)
- [15] Takahama T, and Sakai S. Constrained Optimization by the  $\varepsilon$  Constrained Differential Evolution with an Archive Gradient-Based Mutation and Feasible elites., *2006 IEEE International Conference on Evolutionary Computation*, pp. 1–8, 2006.
- [16] AngelCode.com, <https://www.angelcode.com/angelscript/>. (accessed November 22, 2021).
- [17] About .des file, [http://openvsp.org/wiki/lib/exe/fetch.php?media=workshop15:vsp\\_xddm.pdf](http://openvsp.org/wiki/lib/exe/fetch.php?media=workshop15:vsp_xddm.pdf). (accessed November 22, 2021).
- [18] Documentation for the OpenVSP API, [http://openvsp.org/api\\_docs/latest/](http://openvsp.org/api_docs/latest/), (accessed November 22, 2021).
- [19] McCormick B. W. *Aerodynamics, Aeronautics & Flight Mechanics*, John Wiley & Sons, Inc. USA, 1979, pp. 130–139, 185–193.
- [20] Abbot I. A, and Von Doenhoff A. E. *Theory of Wing Sections*, Dover Publications, Inc, New York, 1959, p. 462.
- [21] UIUC Airfoil Coordinates Database, [https://m-selig.ae.illinois.edu/ads/coord\\_database.html](https://m-selig.ae.illinois.edu/ads/coord_database.html), (accessed November 22, 2021).