OPTIMIZATION OF FLAPPING WING MOTION

Keiichi Ito
University of Tokyo

Keywords: *Vortex Lattice Method, Genetic Algorithms, Sequential Quadratic Programming, Neural Network, Unsteady Aerodynamics*

Dept. of Aeronautics and Astronautics, University of Tokyo, 7-3-1Hongo, Bunkyo-ku, Tokyo113-8656, Japan
E-mail: ito@izumo.t.u-tokyo.ac.jp

Abstract

This paper presents an application of Genetic Algorithm and Sequential Quadratic Programming in improving the performance of flapping wings. A Vortex Lattice Method was employed to model a finite-span flapping wing. The flapping wing model investigated was a rigid rectangular-planform plate with two degrees of freedom, i.e. pitching and heaving. A combination of Genetic Algorithms and Sequential Quadratic Programming was employed to find out the optimum combination of flapping frequency, flapping amplitude, pitch amplitude and phase difference between pitching and heaving that gave maximum efficiency for a given thrust. The hybrid optimization method was superior in solution search capability to that of GA or SQP alone. A function approximation of aerodynamic forces was made using Radial Basis Function to improve computational time cost. However, further refinement was necessary for a successful application of this method in high dimensional search space.

1 Nomenclature

A	=	heave amplitude
A_{ij}, B_{ij}	=	influence coefficient to the control points
$\tilde{A}_{ij}$, $\tilde{B}_{ij}$	=	influence coefficient to the quarter chord
AR	=	aspect ratio
c	=	wing chord
D	=	drag as in wind tunnel coordinate
ea	=	center of pitch location
h	=	panel vertical distance (heaving)
L	=	lift as in wind tunnel coordinate
m	=	number of wake vortex lattices
n	=	number bound vortex lattices
q	=	pitch rate
S	=	wing area
t	=	time
U	=	free-stream velocity
$\vec{V}$	=	velocity vector
x, y, z	=	wing coordinates with y along elastic axis
x_i	=	x-coordinates of control points
Γ	=	vortex or circulation
Γ_b	=	bound vortex
Γ_w	=	wake vortex
Θ	=	pitch amplitude
θ	=	pitch angle
ρ	=	air density
ϕ	=	potential function or phase angle
ω	=	circular flapping frequency

2 Introduction

Recently, flapping wing flight has attracted considerable attention partly due to the increase in interest towards Micro Air Vehicles (MAVs). In the past years, there has been a fairly good number of research concerning the mechanics and aerodynamics of flapping or oscillating wings [1, 2, 8]. However, few literatures have addressed the problem of designing an optimal flapping wing. Computational tools for designing a flapping wing with desired performance are needed. This paper will present advances towards such objectives. Here, Vortex Lattice Method was employed to model the aerodynamic force exerted to a rigid flapping wing. The coupling of heaving and pitching motion was optimized using Genetic Algorithms and Sequential Quadratic Programming codes [4]. Inertial forces due to the mass and geometry of the wing are not considered in this paper.

Although Vortex Lattice Method is not appropriate for modeling low-Reynolds number aerodynamics, it incorporates unsteady-aerodynamic effects at modest computational cost, and it serves the purpose of this research, in which the prime objective is to design an optimization procedure for flapping wings.

The challenge of optimizing flapping wings is two fold. One is the computational cost caused by the unsteady aerodynamics. The aerodynamic force calculations are computationally intensive. To obtain average thrust and power, time integration of the aerodynamic force is required, which makes it more time consuming than the steady-state case. The other is the number of parameter to be optimized is bound to be large (especially if we want to find an optimum planform geometry), and as result, search for an optimum solution has to be conducted in a high-dimensional space which are likely to have highly multimodal (i.e. many local minima or maxima) response surfaces.

3 Vortex Lattice Method

The VLM in this paper represented the wing as a planar surface on which a grid of vortex lattices was superimposed. The velocities induced by each vortex lattice at specified control points (3/4 of the chord on the mid-span of each panels) were calculated using the law of Biot-Savart. A summation was performed for all control points on the wing to produce a set of linear algebraic equations for the vortex strengths that satisfied the condition of no flow through the wing, i.e. the tangential condition (at the control points). A few governing equations are in order, and we follow closely the presentation given in [5], [6], and [9]. For an irrotational flow, the velocity may be expressed in terms of a potential function

$$\vec{V} = \nabla \phi \quad (1)$$

In incompressible flow, the continuity equation is

$$\nabla \cdot \vec{V} = 0 \quad (2)$$

Thus,

$$\nabla^2 \phi = 0 \quad (3)$$

Another condition that has to be satisfied by the solution is Kevin's theorem, namely, that there is no net change in the circulation in the field at any time step, or

$$\frac{d\Gamma}{dt} = 0 \quad (4)$$

In this research, the model, which satisfied these four equations, consisted of panels having a constant bound vortex in each quarter chord of a rectangular cell. A control point was located in each cell at the three-quarter point of centerline. As stated previously, if the incident air velocity to the control points are known, a linear algebraic system of equation can be set up to solve for the unknown bound vortex strengths at every time step. The vortex strengths Γ s determined from the past time steps were shed behind the wing to form the wake. In the present model, one panel was used chordwise to represent the bound vortices and the wake was planar, and constrained in the same plane as that of the wing as shown in Fig. 1.

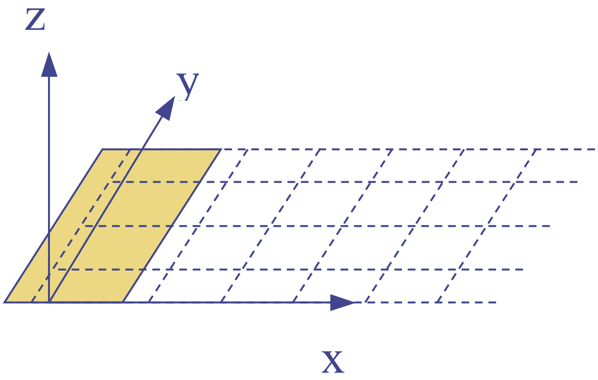

Fig. 1 The wing and the vortex lattices (dashed lines)

The wing was a rigid rectangular-planform plate with two degrees of freedom, i.e. pitching with respect to y-axis and heaving along z-axis. The simplicity of the model works in favor of the optimization process discussed in the next section. The bound vortex strengths Γ_b s were determined from the following equation. Γ_w s are known from previous time steps.

$$-U\sin\theta - qx_i + \frac{\partial h_i}{\partial t}\cos\theta = \left[A_{ij}\right]\begin{bmatrix}\Gamma_{b_1}\\ \vdots \\ \vdots \\ \Gamma_{b_n}\end{bmatrix} + \left[B_{ij}\right]\begin{bmatrix}\Gamma_{w_1}\\ \vdots \\ \vdots \\ \Gamma_{w_m}\end{bmatrix} \quad (5)$$

Once Γ s were determined, the pressure difference could be determined using unsteady Bernoulli equation as derived by Katz. Then, lift and induced drag can be calculated from the following equations.

$$L = \rho U \sum_{k=1}^{n} \Gamma_{b_k} \Delta y + c\frac{d}{dt}\left(\sum_{k=1}^{n} \Gamma_{b_k}\right)\Delta y \quad (6)$$

$$D = L\tan\theta + \frac{\rho\left(\sum_{k=1}^{n} W_k \Gamma_{b_k}\right)\Delta y}{\cos\theta} \quad (7)$$

where

$$\begin{bmatrix}W_1\\ \vdots \\ \vdots \\ W_n\end{bmatrix} = \left[\tilde{A}_{ij}\right]\begin{bmatrix}\Gamma_{b_1}\\ \vdots \\ \vdots \\ \Gamma_{b_n}\end{bmatrix} + \left[\tilde{B}_{ij}\right]\begin{bmatrix}\Gamma_{w_1}\\ \vdots \\ \vdots \\ \Gamma_{w_m}\end{bmatrix} + U\sin\theta + qx_i - \frac{\partial h_i}{\partial t}\cos\theta \quad (8)$$

When D is negative, it means the flapping is producing thrust. Power was computed from the following equation

$$P = -(L\dot{h} + M_p\dot{\theta}) \quad (9)$$

where M_p is pitching moment about ea.

Heaving and pitching are assumed to follow sinusoidal motions, i.e.

$$h = A\cos\omega t \quad (10)$$

$$\theta = \Theta\cos(\omega t + \phi) \quad (11)$$

Fig. 2 shows different flapping modes with respect to the phase angle ϕ.

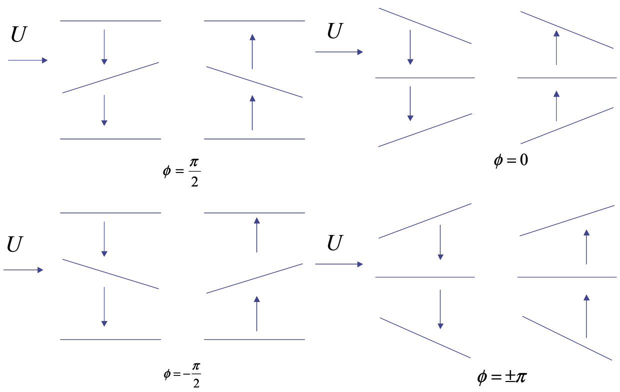

Fig. 2 Side view of the flapping motion with respect to phase angles

4 Optimization Procedure

In this section, optimization procedures employed in this research are briefly described.

4.1 Genetic Algorithms

Floating Point Genetic Algorithm was employed to find maximum efficiency (i.e.

CT/CP) with respect to design variables such as flapping frequency, phase angle between heaving and pitching, and pitching axis location. By floating point, it is meant that the genes are not expressed in terms of binary codes but as decimal base real valued variables. GAs search the solution space of a function through the use of simulated evolution, i.e. the survival of the fittest strategy [4]. The basic idea is as follows. First, populate the solution space with individuals with different values assigned to their genes (or variables), which are elements of chromosomes (or vectors). Thus, for example, if we want to maximize y as in the following equation

$$y = f(x_1, x_2, x_3) \qquad (12)$$

x1, x2, and x3 can be thought of as genes whereas a row vector [x1, x2, x3] as a chromosome or individual. A population consists of numerous instances of this vector. These vectors are evaluated with fitness function (12) Then, better chance of survival is given to the more fit individual (i.e. larger y) from the population and pass the characteristics of the fit individual to the next generation (or search the state-space neighborhood of these individuals) through an operation called cross over (usually, two new individuals are created out of two parents, but not necessarily). Mutation is employed to avoid premature convergence. Successive application of the process enables us to find the optimal solution. In the current study, decimal base floating-point representation of genes was used

In this research, Normalized Geometric Ranking method [4] was employed in the process of selecting mates, and Arithmetic Cross Over method [4] was employed in the reproduction process.

4.2 Hybrid GA-SQP

A hybrid of GA and SQP was tried out. SQP is a very standard gradient-based optimization method that closely mimics Newton's method in constrained optimization [7]. In the hybrid algorithms, SQP is nested inside GA. Each chromosome in the population is optimized using SQP and then mated and reproduced using GA. The SQP iterations were not necessarily carried out until convergence. Rather, it was stopped after fixed number of iteration, which was set to 40 function evaluations in this research. This helped in maintaining the diversity of the population. Fig. 3 shows the flow of the optimization procedure. The SQP worked as a subroutine to the fitness function evaluation, i.e. SQP took the chromosome as the initial condition and locally improved to return a modified chromosome. Thus, the population was shifted along its local gradient before the selection of mates took place.

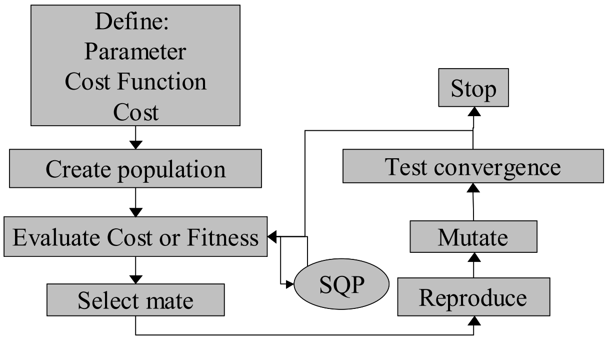

Fig. 3 Flow chart for Hybrid GA-SQP

4.3 Function Approximation (Interpolation)

In order to save computational time, Radial Basis Function was employed to model the mapping between the design variables and the flapping efficiency. Radial Basis Function is a form of neural network, which employs gauss distribution function in the hidden layer and linear transfer function in the output layer. The weight matrix in the hidden layer corresponds to the location of center of the gauss distribution. The weight matrix in the output layer corresponds to the magnitude (height) of the gauss distribution. The spread or sigma is usually predetermined as a constant. A significant advantage of RBF modeling over other neural network is that its weight matrices are determined very quickly.

Instead of computing aerodynamic forces for every change in the design variables, RBF was trained with a set of data obtained by calculating the aerodynamic forces with respect to the design variables. Once the RBF was trained, it was used in the optimization procedure to obtain a solution.

5 Results

Numerical computation of thrust and efficiency was performed based on the simple model described previously. The following values were used in the calculation unless specified otherwise.

ea	=	0.25
h	=	Chord length: 0.05m
S	=	0.075 m^2
AR	=	3
U	=	1 m/s
ρ	=	1 kg/m^3

A desktop computer with 866 MHz Intel® Pentium® III processor and Matlab® was used to produce the numerical results.

Fig. 4 and Fig.5 show the average thrust and average efficiency with respect to two design variables. Fig. 4 shows the thrust at various flapping frequencies and phase differences (here, denoted as phi1). Fig. 5 shows the flapping efficiency with respect to the flapping frequency and phase difference between heaving and pitching. Efficiency is defined as the ratio between thrust coefficient and power coefficient, i.e. CT/CP.

Fig. 6 is a carpet plot version of Fig. 5 and Fig. 7 is the RBF reconstruction of Fig. 6. Five uniformly spaced data was taken along each axis. Therefore total of 25 combinations of phase difference and flapping frequency were used to train the RBF. Using this RBF model, GA was employed to find the peak, i.e. the maximum of the efficiency. Table 1 compares two methods of optimization. One is the method just described and the other is the benchmark where aerodynamic forces were calculated every time the fitness function (efficiency in this case) was called. As can be seen, there was a substantial reduction in computational time. However, in higher dimension, i.e. larger number of design variables, such advantage could be offset by growing number of calculation needed to construct the approximated evaluation function (or fitness function). Table 2 shows such a case. Note also that the function approximation introduced errors that affected the optimum values as one can observe in the discrepancy in the solutions given by the two methods.

Table 3 shows a typical output of two optimization procedures, namely hybrid GA-SQP and GA. GA-SQP almost always gave a better solution than GA with higher value of efficiency. This is due to the superior search capability of GA-SQP, which exploits the global search of GA and the gradient based local search of GA. Comparison with SQP was not enlisted because it was highly sensitive to the initial guess of the solution. With a good guess, SQP is highly efficient and it converged to the solution in about 10% of the time required for GA. With SQP, a random pick of the initial solution almost always gave a totally different solution from that found by GA or GA-SQP

In Table 2, and 3 flapping frequency had a range of 10 to 100 Hz, heaving amplitude of 0 to 15 cm, pitching amplitude of 0 to pi/2 rad.

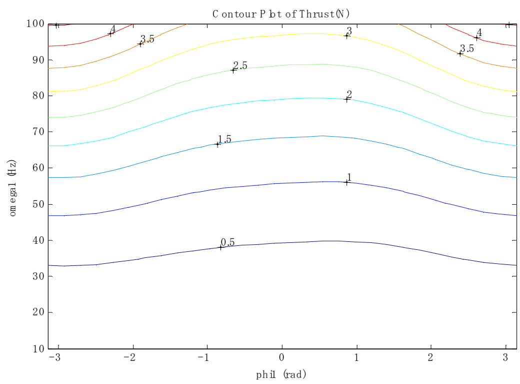

Fig. 4

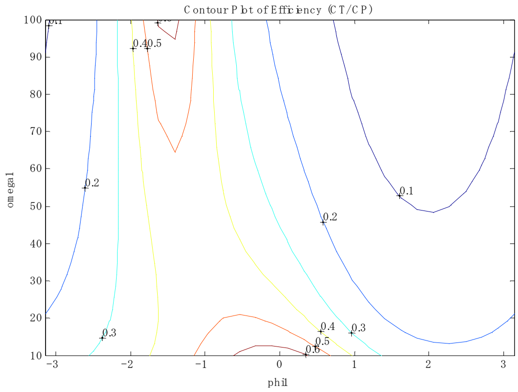

Fig. 5

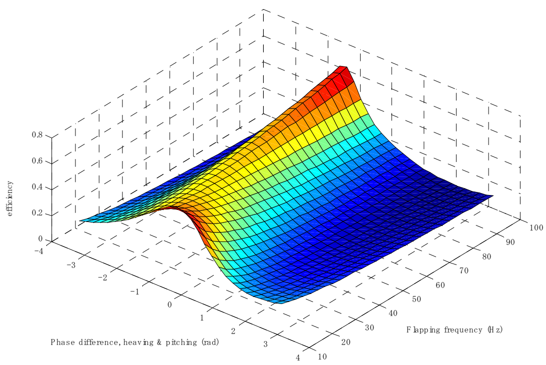

Fig. 6 Carpet Plot of efficiency w.r.t. phase difference between heaving and pitching and flapping frequency

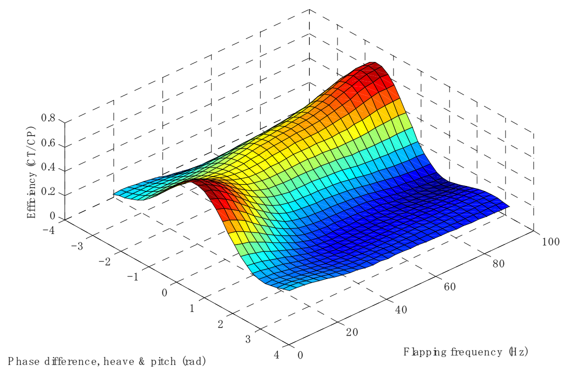

Fig. 7 Carpet Plot of efficiency w.r.t. phase difference between heaving and pitching and flapping frequency. This is the RBF approximation of Fig. 6

optimization using RBF	direct optimization
elapsed time = 36 sec.	elapsed time = 377 sec.
solution: freq.=10Hz, phase=-0.111	solution: freq.=10Hz, phase= -0.0644
flapping efficiency= 0.685	flapping efficiency=0.686

Table 1 Comparison between GA on RBF and GA on original function at higher dimension (i.e. 4 independent variables) with number of population = 100, number of generation = 15 thrust =0.5 N, and C_{Lmax}=inf.

GA on RBF approximation	GA on original fitness function
elapsed time = 5202 sec.	elapsed time = 981 sec.
solution: freq.=31.8 Hz, heave=10.9 cm, pitch=0 rad. elastic axis location=85% chord	solution: freq.=16.6 Hz, heave=12.8 cm, pitch=0 rad. elastic axis location=leading edge
flapping efficiency= 0.28	flapping efficiency=0.31

Table 2 Comparison between GA on RBF and GA on original function at higher dimension (i.e. 4 independent variables) with number of population = 300, number of generation = 15 thrust =0.5 N, and C_{Lmax}=3.0

GA-SQP	GA alone
elapsed time=8796 s	elapsed time=610 s
solution: freq.=70.3 Hz, phase diff=-0.0519 rad. heave=3.58 cm, pitch=0.714 rad.	solution: freq.= 65.1 Hz, phase diff=0.11 rad. heave=4.19 cm, pitch=0.506 rad.
flapping efficiency=0.500	flapping efficiency=0.223

Table 3 Comparison between GA-SQP and GA with number of population = 200, number of generation = 15 thrust =0.5 N, and C_{Lmax}=3.0

6 Conclusions

Optimization of flapping motion was conducted using Floating Point GA. A Neural Network (Radial Basis Function) was employed to model the mapping between the design variables and the flapping efficiency. It was observed that substantial reduction in computational time could be achieved using the Neural Network modeling of fitness function in the optimization procedure. However, in high dimensional search space, the time required to construct function approximation can offset the advantage of the fast optimization. This is due to the exponential increase in sampling points. If we sample 10 values from each independent variable, *n* variables means 10^n points to sample.

Further refinement in the construction of function approximation is necessary before it can be successfully applied to the optimization of flapping wings.

Hybrid GA-SQP algorithms was applied and better solution was found compared to GA and it is a promising method for optimizing flapping wings.

In this paper, only the coupling between heaving and pitching, and corresponding amplitudes were considered (i.e. 4 independent variables). Convergent optimal solutions were obtained. A more sophisticated model can be developed to optimize not only the flapping motion but also the planform shape, inertia and elastic properties. To achieve this objective, further refinement in optimization procedure is necessary. Particularly, with respect to

1) Faster convergence in optimization,
2) More efficient construction of function approximation.

7 References

[1] Azuma A. *The biokinetics of flying and swimming*. Springer-Verlag, pp. 100 – 109,1992.

[2] Bisplinghoff R L, Ashley H, and Halfman R L. *Aeroelasticity*. Addison-Wesley, pp. 260 – 261, 274, 285, 393 – 394, 1955.

[3] Haupt R L and Haupt S E. *Practical genetic algorithms*. John Wiley & Sons, pp. 1 – 64, 1998.

[4] Houck C R, Joines A J, and Kay M G. *A genetic algorithm for function optimization: a Matlab implementation.* Manual for GAOT (Genetic Algorithms Optimization Toolbox), North Calorina State University.

[5] Katz J. *Method for calculating wing loading during maneuvering flight along a three-dimensional curved path*. J. Aircraft, Vol. 16, November 1979, pp. 739 – 741.

[6] Levin D and Katz J. *Vortex-lattice method for the calculation of the nonsteady separated Flow over delta wings*. J. Aircraft, Vol. 18, December 1981, pp. 1032 – 1037.

[7] Rao Singiresu. *Engineering optimization: theory and practice*. 3rd edtition, JohnWiley & Sons, 1996

[8] Shyy W, Berg M, and Ljungqvist D. *Flapping and flexible wings for biological and micro air vehicles*. Prog. Aero. Sci., Vol. 35, 1999, pp. 455 – 505.

[9] Vest M S, and Katz J. *Unsteady aerodynamic model of flapping wings*. AIAA J. Vol. 34, July 1996, pp. 1435 – 1440.