JAST, Vol. 8, No. 1, pp 45–56

JAST
Journal of Aerospace Science and Technology

Trim and Maneuverability Analysis of a UAV Using a New Constrained PSO Approach

J. Karimi[1], S.H. Pourtakdoust[2], H. Nobahari[3]

Performance characteristic of an Unmanned Air Vehicle (UAV) is investigated using a newly developed heuristic approach. Almost all flight phases of any air vehicle can be categorized into trim and maneuvering flights. In this paper, a new envelope called trimmability envelope is introduced and sketched within the conventional flight envelope for a small UAV. Optimal maneuverability of the intended UAV is evaluated for minimum time pull-up and turn maneuvers. For both the trim and the maneuver problems, the nonlinear 6DOF dynamic models as well as the vehicle constraints are considered. A heuristic based constrained optimization approach is developed to solve both the trim and maneuver problems. Several interesting performance characteristics are extracted. The results are indicative of a good potential for the proposed algorithm to handle complex constrained optimization problems in aerospace engineering.

NOMENCLATURE

$F(\mathbf{x})$	Cost function
F_A	Aerodynamic force in body frame, N
F_T	Thrust forces in body frame, N
$\mathbf{g}(\mathbf{x})$	Constraints vector function
g	Gravitational acceleration, m/sec^2
h	Altitude, m
I	Moment of inertia, kg.m^2
J	Advance ratio
L	Lift force, N
$\bar{L}$	Rolling moment, N.m
m	mass, kg
$\bar{M}$	Pitching moment, N.m
n_z	Load factor
$\bar{N}$	Yawing moments, N.m
p	Body axis roll rates, rad/sec
q	Body axis pitch rates, rad/sec
r	Body axis yaw rates, rad/sec
T	Thrust, N
u	Velocity in body frame x-axis, m/sec
v	Velocity in body frame y-axis, m/sec
w	Velocity in body frame z-axis, m/sec
V	Total velocity, m/sec
x	Inertial longitudinal position, m
y	Inertial lateral position, m
α	Angle of attack, rad
β	Angle of Side slip, rad
δ_A	Aileron deflection, rad
δ_E	Elevator deflection, rad
δ_R	Rudder deflection, rad
ϕ	Roll angle, rad
θ	Pitch angles, rad
ψ	Yaw angle, rad

1. *PhD Candidate, Dept. of Aerospace Eng., Sharif Univ. of Tech., Tehran, Iran, Email: karimi_j@ae.sharif.edu.*
2. *Professor, Dept. of Aerospace Eng., Sharif Univ. of Tech., Tehran, Iran, Email: pourtak@sharif.edu.*
3. *Assistant Professor, Dept. of Aerospace Eng., Sharif Univ. of Tech., Tehran, Iran, Email: nobahari@sharif.edu.*

INTRODUCTION

Performance analysis of air vehicles are usually performed for steady state and maneuvering flights [1, 2]. In steady state flights, the flight performance parameters such as range, endurance and flight envelope are calculated based on a point mass model. However, it is possible to find the steady state condition for an

aircraft using the nonlinear 6DOF model. Flying in steady state conditions is usually termed as trimmed flight. A question that arises here is that, given the physical limitations of the aircraft and its dynamics, can it be trimmed within the whole flight envelope? The first part of this article is aimed at seeking a new envelope, namely the trimmability envelope. Evaluating the aircraft's ability to optimally perform certain maneuvers and the quality of performing them is another topic, tackled in this research. The fact that makes the optimal maneuvers design and analysis more interesting and complicated is the high degree of realism that should be considered to predict accurate results. In such a condition, one needs to consider the full nonlinear 6DOF model of the vehicle dynamics that reflects the coupling effects, as well as performance constraints. The advantage of such an analysis in comparison with flight tests is the low cost, while it proposes precise and valuable information, usable in analysis and design of aircraft.

Optimization algorithms are search methods that converge to the solution through iterative schemes. A typical optimization problem includes a set of cost functions, $F(\mathbf{x})$, a set of design variables, $\mathbf{x} = [x_1, x_2, ..., x_n]^T$, and a set of constraints, $g_i(\mathbf{x}) \leq 0$, $i = 1, 2, ..., m$. Optimization methods are categorized into unconstrained and constrained algorithms [3]. In a constrained optimization method, the objective is to find the vector $\mathbf{x}$ that optimizes the cost function $F(\mathbf{x})$ and at the same time satisfies the set of constraints $g_i(\mathbf{x}) \leq 0$. The set of constraints allows us to know which parts of the search space are feasible.

The optimization algorithms can also be categorized into classical and heuristic approaches. The direct search methods like simplex and the indirect search or gradient based methods such as steepest descent, Newton, and Marquardt are samples of classical methods [3]. There are many studies that have obtained trim and maneuvers on the basis of the classical optimization methods. The trim problem was solved using simplex optimization method [4]. In Ref. [5], the aircraft minimum time turn maneuvers have been investigated based on the 3DOF model. Utilizing the 6DOF model, a near optimal scheme is used to perform pitch and yaw-reversal maneuvers [6]. The 6DOF dynamic model was also used to perform optimal closed-air- combat maneuvers for an advanced fighter, equipped with thrust vectoring controls [7].

Aside from the classical optimization methods, the heuristic approaches have also been developed. An analogous aspect of these algorithms is that they are all inspired by the natural phenomenon. They have been successful in handling the complex optimization problems. Evolutionary Computation (EC), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are examples of heuristic methods [8].

Most of the optimization problems occur in continuous search spaces. Basic variants of EC and ACO work in discrete search spaces. ACO has been empowered to handle continuous search spaces [9] and there are many adaptations of evolutionary algorithms to continuous optimization problems. However, the basic PSO algorithm is inherently capable of handling continuous search spaces. So, PSO is chosen to be utilized in this paper.

Basic PSO is not specialized for constrained optimization. However, various modifications have been suggested to enhance it for constrained optimization problems. The penalty function method adds a second function to the cost function to penalize infeasible solutions [9, 10]. There are also methods based on keeping the feasible solutions [11]. Zang *et.al.* [12] proposed a special periodic mode technique to handle the constraints where the global best solutions in the vicinity of the boundaries are kept to steer the swarm toward the boundaries and thus enhance the exploration capability. There are methods that use multi-objective optimization techniques to handle the constraints [13]. Pulido *et.al.* [14] used the feasibility tournament method to handle the constraints in PSO. The feasibility tournament, proposed by Deb [15], uses rules similar to Pareto dominance in multi-objective optimization. Zavala *et.al.* [16, 17] have also proposed an efficient algorithm, namely Particle Evolutionary Swarm Optimization (PESO) that uses the feasibility tournament.

The current research uses efficient features of the PESO, and develops a new constrained optimization approach. The quadratic interpolation operator is utilized to enhance both the exploitation and exploration capabilities of PSO. The proposed algorithm hybridizes the simple optimality property of this operator with the global search ability of PSO as a heuristic optimization method. Using random inertia weight and randomizing the velocity vector of saturated particles more improves the proposed algorithm. The developed algorithm is used to optimally solve trimmability and maneuverability problems. The trim condition is obtained for several regimes in the flight envelope and a new concept, called trimmability envelope, is introduced. The optimal behavior of vehicle during maneuvers, considering the nonlinear dynamic model, is also obtained. Using cubic spline, an optimal continuous time history is earned that is more preferred than a discrete or a piecewise-linear time history.

BASIC PSO ALGORITHM

PSO algorithm is a population-based optimization methodology first introduced by Kennedy and Eber-

hart [18] in 1995. The basic idea was based on the simulation of simplified animal social behaviors such as fish schooling and bird flocking.

In basic PSO [8], the vectors $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t)$ denote the position and the velocity of particle i in the search space at time/iteration t. The position at iteration $t+1$ is updated in a current movement by:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \tag{1}$$

The velocity vector reflects both the experimental knowledge of a particle and the socially-exchanged information from the neighboring particles. The velocity vector of particle i at iteration $t+1$ is calculated as:

$$\mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + c_1 r_1 (\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2 r_2 (\hat{\mathbf{y}}(t) - \mathbf{x}_i(t)) \tag{2}$$

where c_1, c_2 are constant parameters, r_1, r_2 are random numbers between $[0,1]$, ω is the inertia weight that controls the exploration/exploitation abilities of the swarm, $\mathbf{y}_i(t)$ is the personal best position of particle i at iteration t, and $\hat{\mathbf{y}}(t)$ represents the best position found by the neighbors at iteration t.

NEW PROPOSED ALGORITHM

The position and velocity are updated based on Eq. (1,2). Moreover, the proposed algorithm utilizes the following new strategies:

First, the inertia weight is adjusted to be a dynamic random variable:

$$\omega = \text{rand}\,(0,1) \tag{3}$$

where rand (0,1) is a uniform random number between $[0,1]$. Random adjustment of the inertia weight has also been utilized by others; Eberhart and Shi [19] used $\omega \approx N(0.72, \sigma)$, where σ is small enough to ensure that ω is not predominantly greater than one. Alternatively, Peng *et.al.* [20] used $\omega \approx c_1 r_1 + c_2 r_2$ with no random scaling of the cognitive and social components. Randomly changing inertia weight ensures a good mix of focusing on both exploration and exploitation.

In basic PSO algorithm, the velocity can quickly diverge to large values. Consequently, particles have large position updates that result in particles leaving the boundaries of the search space or sticking to the boundaries. In order to solve this problem, usually the velocity clamping method [21] is utilized that saturates the velocity of particles with a predefined maximum value, V_{max}. Large values for V_{max} increase the exploration. We believe that randomizing the velocity of saturated particles can also increase the exploration ability of the swarm, because this strategy changes the direction of the velocity vector and in this way it can also improve the exploration. Therefore, in this study, the velocity vector of the saturated particles is randomized as:

$$\mathbf{v}_i(t+1) = \begin{cases} \mathbf{v}_i(t+1) & \text{if } |\mathbf{v}_i(t+1)| \le V_{max} \\ \text{rand}\,(-V_{max}, V_{max}) & \text{if } |\mathbf{v}_i(t+1)| > V_{max} \end{cases} \tag{4}$$

where V_{max} is the maximum value considered for each element of the velocity vector.

In addition, some other modifications are made on the structure of the basic PSO that are described in the following subsections. The pseudo code of the proposed algorithm is also presented in Table 1. The new algorithm has four control parameters which are presented in Table 2.

Table 1. The proposed algorithm pseudo code.

```
// Set the parameters N, c1 ,c2, Vmax
// Place N random particles into the search space each
   with a random initial velocity vector
While stopping criteria are not satisfied
For each particle do
// Position update
//Find the particles best position
  according to the feasibility tournament
// Perform M-perturbation operator
// Perform quadratic interpolation operator
End
End
```

Table 2. The proposed algorithm parameters.

Parameter	Value
Number of particles (N)	100
c_1	1
c_2	1
Max. velocity ($V_{\max}$)	2

The social network structure

In PSO, each particle moves following a leader; this feature is realized through the social term of Eq. (2). A leader can be global to all the swarm, or local to a particle's neighborhood. In the latter case, there are as many local leaders as the neighborhoods. Having more than one leader in the swarm translates into more attractors or good spots within the search space and avoids premature convergences. On the other hand, having one leader can result in faster convergence and a higher risk to converge to a local extremum. The flow of information through the swarm depends on the neighborhood structure. There are various neighborhood structures like star, ring, wheel and Von Neumann [8].

The singly-linked ring structure has risen from analyzing the ring neighborhood as a double-linked ring. Suppose that every particle is assigned a permanent label which is used to construct the neighborhoods. In a ring neighborhood structure, as in Figure 1,

a particle k has two neighbors, particles $k-1$ and $k+1$. In turn, particles $k-1$ and $k+1$ have particle k as one of their neighbors. In this way, there is a mutual attraction between consecutive particles, forming overlapped clusters. The slow convergence of the ring structure has been empirically shown in Ref. [22]. Thus, Ref. [16, 17] proposed a topology of singly-linked ring, presented in Figure 2. This topology keeps two neighbors for each particle, but breaks the mutual attraction between neighbors. In this way, the information is transmitted faster through the whole swarm than in the original ring topology. Therefore, the singly-linked ring keeps the exploration at the search space, and increases the exploitation of the best solutions. In the present work, the singly-linked ring structure is used.

Feasibility tournament

The feasibility tournament [15, 16, 17] has a tournament selection operator that compares two solutions at each instant of time. These solutions correspond to the cost function, $F(\mathbf{x})$, and a constraints violation evaluation function. The solution which violates one or more constraints is termed "infeasible", and the one which never violates the constraints is termed "feasible". The feasibility tournament enriches information on the search space, especially in border areas. The solution selection is achieved using the following criteria:

- Any feasible solution is preferred to any infeasible solution.
- Among two feasible solutions, the one having better cost function value is preferred.
- Among two infeasible solutions, the one having smaller constraint violation is preferred.

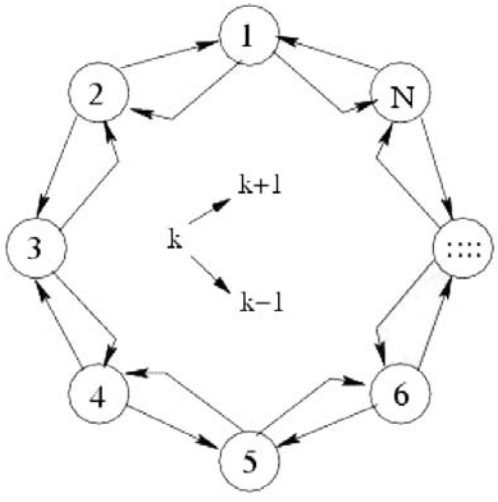

Figure 1. Ring neighborhood structure.

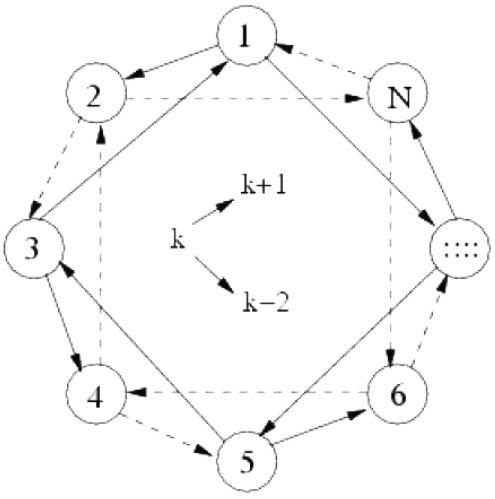

Figure 2. Singly-linked ring structure.

Quadratic interpolation operator

In view of the idea that a combination of classical and heuristic optimization algorithms can improve their performance, inspired by classical optimization techniques, a new operator called "quadratic interpolation" is utilized in the proposed algorithm.

According to this operator, for every particle, a new particle is generated in each iteration. The quadratic interpolation operator utilizes the current experience of the intended particle, termed as main parent, and two other randomly chosen particles of the swarm. Subsequently, the fitness of the new particle is compared with the best experience of its main parent implementing the feasibility tournament criteria. The main parent's best position will be replaced by the new particle, if it is better. Note that, in this strategy, the position of the parent is not changed and only its best position may be changed.

According to the quadratic interpolation, the position of the new particle, generated by particle j and two random particles $nb1$ and $nb2$, at t-th iteration and j-th dimension of the search space is calculated as:

$$x_{new}(t,j) = \begin{cases} \frac{a}{2b}, \\ \text{If } b > 0 \text{ for minimization} \\ \text{(If } b < 0 \text{ for maximization)} \\ \text{rand}, \\ \text{If b} \leq 0 \text{ for minimization} \\ \text{(If } b \geq 0 \text{ for maximization)} \end{cases} \tag{5}$$

and

$$
\begin{aligned}
a &= (x_{nb1}(t,j)^2 - x_{nb2}(t,j)^2).\mathrm{J}_i(t) \\
&+ (x_{nb2}(t,j)^2 - x_i(t,j)^2).\mathrm{J}_{nb1}(t) \\
&+ (x_i(t,j)^2 - x_{nb1}(t,j)^2).\mathrm{J}_{nb2}(t) \\
b &= (x_{nb1}(t,j) - x_{nb2}(t,j)).\mathrm{J}_i(t) \\
&+ (x_{nb2}(t,j) - x_i(t,j)).\mathrm{J}_{nb1}(t) \\
&+ (x_i(t,j) - x_{nb1}(t,j)).\mathrm{J}_{nb2}(t)
\end{aligned} \tag{6}
$$

where $x_i(t,j)$, $x_{nb1}(t,j)$, $x_{nb2}(t,j)$ and $x_{new}(t,j)$ indicate the j-th component of i-th, $nb1$-th, $nb2$-th and the new particle position vectors at t-th iteration, respectively.

$\mathrm{J}_i(t)$, $\mathrm{J}_{nb1}(t)$ and $\mathrm{J}_{nb2}(t)$ are the i-th, $nb1$-th, $nb2$-th particle fitness at t-th iteration. As Eq. (5) indicates, for a minimization (maximization) problem, there exists a minimum only if the denominator is positive (negative). In other cases that the denominator is zero or negative, for each dimension of the search space, an allowable random number will be generated.

The quadratic operator is used instead of the C-Perturbation operator in PESO [16, 17]. C-Perturbation operator is applied to the particles personal best position to yield a set of temporal particles using differential evolution algorithm. The temporal particles are generated by randomly moving the personal best position of each particle toward two random particles in the swarm.

The advantage of quadratic interpolation operator is that it can enhance exploitation and exploration in the cases that new particles are generated randomly.

M-Perturbation operator

Similar to Ref. [16, 17], the personal best position of particles in every dimension of the search space is perturbed with a predetermined probability equal to inverse of the search space dimension and temporal particles are generated. Then, each temporal particle is compared with the personal best experience of its corresponding particle via feasibility tournament. The corresponding particle's best position will be replaced by the new particle, if it is better.

VEHICLE DYNAMICS

The flying vehicle, studied here, is a propeller driven UAV, called RASCAL [24]. The nonlinear 6DOF dynamic equations, assuming the flat earth, are [4]:

$$\dot{u} = vr - wq + \frac{-\mathrm{m}g\sin(\theta) + F_{Ax} + F_{Tx}}{\mathrm{m}} \tag{7}$$

$$\dot{v} = -ur + wp + \frac{\mathrm{m}g\sin(\phi)\cos(\theta) + F_{Ay} + F_{Ty}}{\mathrm{m}} \tag{8}$$

$$\dot{w} = uq - vp + \frac{\mathrm{m}g\cos(\phi)\cos(\theta) + F_{Az} + F_{Tz}}{\mathrm{m}} \tag{9}$$

$$V = \sqrt{u^2 + v^2 + w^2} \tag{10}$$

$$\dot{V} = \frac{u\dot{u} + v\dot{v} + w\dot{w}}{V} \tag{11}$$

$$\dot{\alpha} = \frac{\mathrm{u}\dot{\mathrm{w}} - \mathrm{w}\dot{\mathrm{u}}}{\mathrm{u}^2 + \mathrm{w}^2} \tag{12}$$

$$\dot{\beta} = \frac{V\dot{v} - v\dot{V}}{V\sqrt{u^2 + w^2}} \tag{13}$$

$$
\begin{aligned}
\dot{p} &= -\frac{(-\mathrm{I_y}\mathrm{I_z} + \mathrm{I_{xz}^2} + \mathrm{I_z^2})qr + (\mathrm{I_{xz}}\mathrm{I_y} - \mathrm{I_{xz}}\mathrm{I_x} - \mathrm{I_{xz}}\mathrm{I_z})pq - A^*}{\mathrm{I_x}\mathrm{I_z} - \mathrm{I_{xz}^2}} \\
A^* &= \mathrm{I_z}\bar{L} + \mathrm{I_{xz}}\bar{N}
\end{aligned} \tag{14}
$$

$$\dot{q} = \frac{(\mathrm{I_z} - \mathrm{I_x})pr - \mathrm{I_{xz}}(p^2 - r^2) + \bar{M}}{\mathrm{I_y}} \tag{15}$$

$$
\begin{aligned}
\dot{\mathrm{r}} &= -\frac{(\mathrm{I_x}\mathrm{I_y} - \mathrm{I_x^2} - \mathrm{I_{xz}^2})pq + (\mathrm{I_x}\mathrm{I_{xz}} + \mathrm{I_{xz}}\mathrm{I_z} - \mathrm{I_{xz}}\mathrm{I_y})qr - B^*}{\mathrm{I_x}\mathrm{I_z} - \mathrm{I_{xz}^2}} \\
B^* &= \mathrm{I_{xz}}\bar{L}\mathrm{I_x}\bar{N}
\end{aligned} \tag{16}
$$

$$\dot{\phi} = p + q\sin(\phi)\tan() + r\cos(\phi)\tan(\theta) \tag{17}$$

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi) \tag{18}$$

$$\dot{\psi} = \frac{q\sin(\phi) + r\cos(\phi)}{\cos()} \tag{19}$$

$$
\begin{aligned}
\dot{x} &= u\cos\theta\, Cos\psi + v(\sin\phi\ \sin\theta\ \cos\psi\ - \cos\phi\ \sin\psi) \\
&+ w(\cos\phi\ \sin\theta\ \cos\psi + \sin\phi\ \sin\psi)
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
\dot{y} &= u\cos\theta\sin\psi + v(\sin\phi\ \sin\theta\ \sin\psi\ + \cos\phi\ \cos\psi) \\
&+ w(\cos\phi\ \sin\theta\ \sin\psi - \sin\phi\ \cos\psi)
\end{aligned} \tag{21}
$$

$$\dot{h} = u\sin\theta + v\sin\phi\ \cos\theta\ + w\cos\phi\ \cos\theta \tag{22}$$

AIRCRAFT TRIM PROBLEM

The concept of a singular or equilibrium point of a time invariant system is introduced in the theory of nonlinear control systems [25]. For a nonlinear system with the general form of $f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{U})$, a singular point is a point that satisfies $f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{U})$ with $\dot{\mathbf{x}} = \mathbf{0}$ and $\mathbf{U}$=constant. This concept has a strong intuitive appeal for aerospace vehicles since for a system at rest, all derivatives are zero and the vehicle flies with the minimum load with steady state conditions. Such a condition is denoted as trim condition for aircraft. In this work, three of the most conventional trim states are investigated;

- steady wings level (V =constant, $\dot{h} = \dot{\psi} = 0$),
- steady level turn (V =constant, $\dot{\psi}$ =constant, $\dot{h} = 0$) and
- steady climb (V =constant, $\dot{h}$ =constant, $\dot{\psi} = 0$).

In general, the conditions corresponding to V =constant, $\dot{h} = 0$, and $\dot{\psi} = 0$ can construct all possible trim states. The steady-state flight can be calculated using the flat-Earth equations of motion, Eq. (10-16) and taking $\dot{V}, \dot{\alpha}, \dot{\beta}, \dot{p}, \dot{q}, \dot{r} = 0$ and $\mathbf{U}$ =constant. Therefore, a set of nonlinear algebraic equations are obtained which should be solved to get the trim states and controls. However, instead of solving this set of nonlinear algebraic equations, that is usually cumbersome, an optimization problem with a design vector of $\mathbf{W} = [\alpha, \beta, \delta_E, \delta_A, \delta_R, J]^T$ can be solved by considering the following cost function [4]:

$$F = \min\left(\dot{V}^2 + 10\dot{\alpha}^2 + \dot{\beta}^2 + \dot{p}^2 + 100\dot{q}^2 + \dot{r}^2\right) \qquad (23)$$

where δ_E, δ_A, δ_R and J are elevator, aileron, rudder and propeller advance ratio. If the cost function converges to zero, with a predefined small tollerance set to 10^{-5} here, one can conclude that $\dot{V}$, $\dot{\alpha}$, $\dot{\beta}$, $\dot{p}$, $\dot{q}$ and $\dot{r}$ have also converged to zero and the corresponding flight condition is called trimmable. An advantage of solving this optimization problem, which provides the trim states, is its potential to be augmented with the vehicle performance and dynamic constraints as well.

The feasible range of the design and state variables are given in Table 3. By constructing the trim problem, several interesting results could be attained, as discused later.

The trimmability envelope

The trimmability of RASCAL UAV in steady wings level flights throughout the flight envelope is invseti-gated using the proposed methodology. The conventional flight envelope, H-V graph, is established by considering the minimum (stall) and maximum velocity for different flight altitudes [1, 2]. A cross plot could be drawn within the H-V graph, showing the trimmability range of the flight envelope in steady wings level flights. It is indicated in Figure 3 that the minimum velocity, required for the intended UAV to be trimmable, is larger than the minimum velocity by a factor about 3. On the other hand, it turns out that the maximum trimmable velocity and the maximum velocity curves concide. This reveals that RASCAL has a trimmability envelope tighter than its H-V envelope.

The feasible range of steady climbs

It is possible to find the allowable range in which the UAV is capable to perform steady climb/descend flights. After exploring the flight envelope via the trim algorithm at the altitude of 2000 (m), the intended

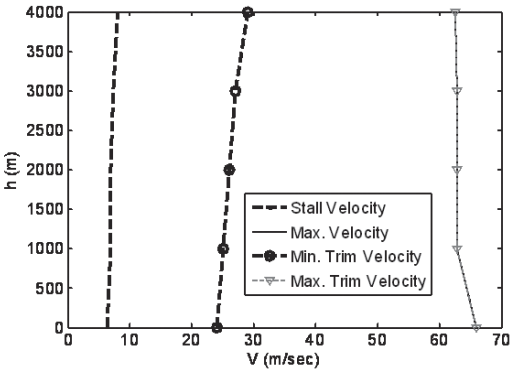

Figure 3. The trimmable range within the flight envelope.

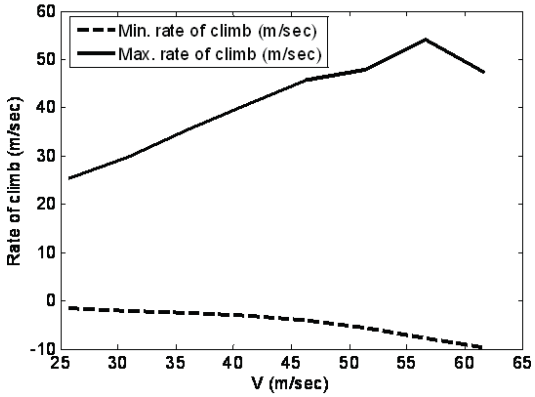

Figure 4. The feasible range of steady climbs for 2000 m altitude.

Table 3. Typical constraint values considered for the Rascal UAV.

Variable	V(m/sec)	α (deg)	β (deg)	p, q, r (deg/s)	ϕ, θ	$\delta_E, \delta_A, \delta_R$ (deg)	J
Min. value	26	-10	-5	-45	-90	-20	0
Max. value	61	18	5	45	90	20	1.13

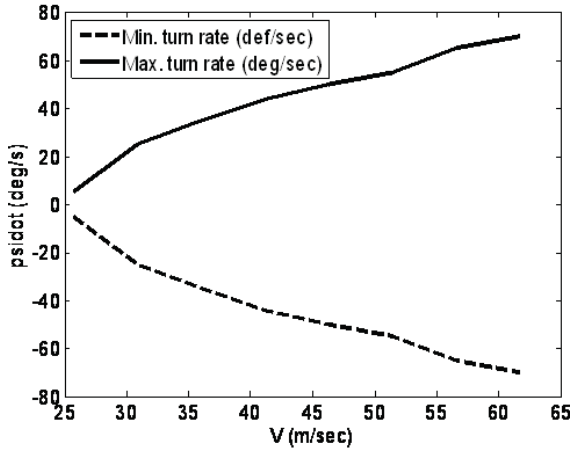

Figure 5. The feasible range of constant turn rate flights for 2000 m altitude.

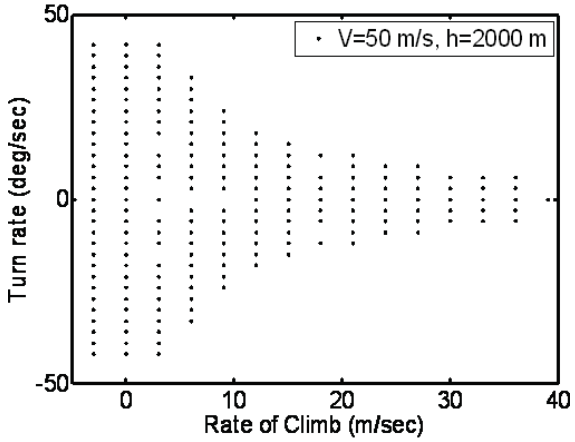

Figure 6. The feasible range of constant rate of climb and constant turn rate m trimmed flights.

range is determined and plotted in Figure 4. It can be seen that the UAV could not be trimmed for large negative rates of climb as it can for positive rates. The reason is that in descending flights, a component of the thrust force is also added to the gravitational force. Therefore, the longitudinal controls cannot keep the vehicle steady in sharp descending flights. On the other hand, due to the thrust capabilities (available thrust) and compressibility effects associated with piston powered, propeller driven vehicles, the maximum trimmable rate of climb has not a monotonic behavior and indicates a decreasing trend after reaching a maximum value.

The feasible range of steady level turns

Similar to the steady climb, the trimmablity range of steady level turn can be investigated for various operating flight conditions. Here, the operating altitude is taken at 2000 (m). As can be seen in Figure 5, the feasible range of steady turn expands with velocity. The positive and negative boundaries are identical, as the UAV is indifferent with regards to direction of turnning flight.

General trimmable range investigation

As mentioned before, the trim condition can be considered as V=constant, $\dot{h}$=constant, and $\dot{\psi}$=constant, in general. Thus, it is also possible to seek the allowable trim range for every altitude and velocity, for different rates of climb and turn rates. Figure 6 indicates a sample result for $V = 50$ (m/sec) and $h = 2000$ (m). This interesting graph indicates that the UAV's ability to perform trimed turns diminishes as the rate of climb increases.

Evaluating the algorithm performance

Performance of the proposed algorithms is evaluated for the case in which $\dot{\psi} = 5$ (deg/sec), $\dot{h} = 0$ and $h = 2000$ (m) and the velocity takes values of the following set.

$V \in 26, 30, 34, 38, 42, 46, 50, 54, 58, 61$ m/s.

Since the proposed algorithm is stochastic, the results may differ for each run. Therefore, for each flight condition, 10 successive runs are performed.

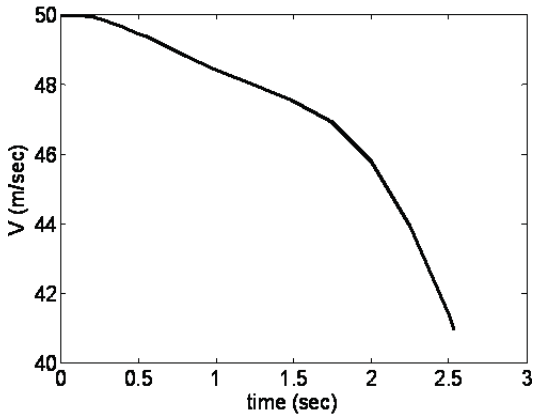

Figure 7. Time variations of velocity during a pull-up maneuver.

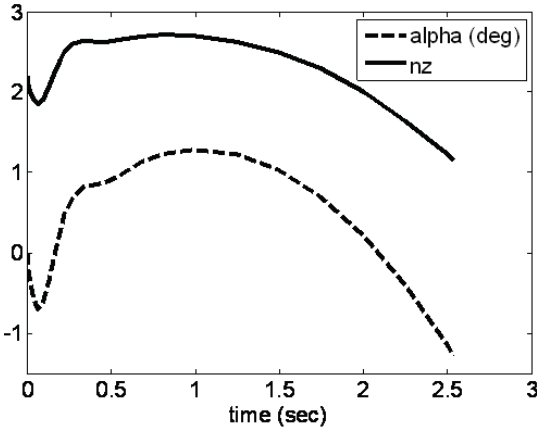

Figure 8. Time variations of angle of attack and load factor during a pull-up maneuver.

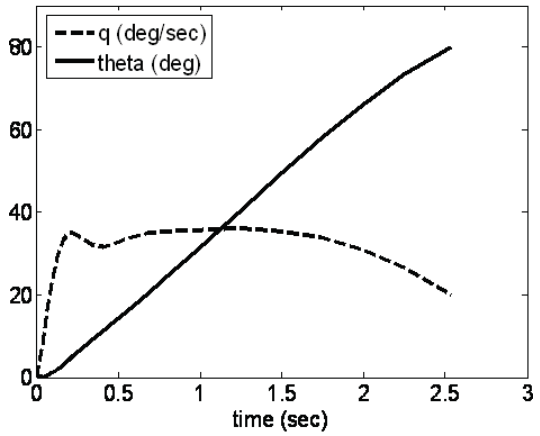

Figure 9. Time variations of pitch rate and pitch angle during a pull-up maneuver.

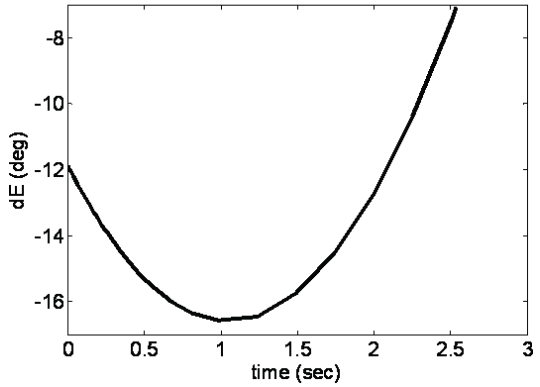

Figure 10. Time variations of elevator deflection during a pull-up maneuver.

The obtained mean and variance are 1.7970×10^{-7} and 9.3826×10^{-14}, respectively. So, it can be conducted that the proposed algorithm has an acceptable converging behavior.

THE MANEUVER PROBLEM

Two types of time optimal pull-up and turn maneuvers are considered and the state vector is taken as:

$$\vec{\mathbf{x}} = [u, v, w, p, q, r, , \theta, \psi, x, y, h]^{\mathrm{T}} \tag{24}$$

The design vector for maneuver problems includes the conventional controls, namely elevator, aileron, rudder and propeller advance ratio. Time optimal formulation of the maneuvering flights yields the optimal time histories of both state and control variables. In order to find practical and feasible maneuvers, it is preferred that the time histories of design variables be continuous. In this regard, a predefined number of equally spaced "control points" are generated for each design variable. Then, using these "control points" and cubic spline interpolation method, continuous time histories are obtained for design variables.

In addition, since the maneuver times are not known in the minimum time maneuver problem, the final time is another component of the design vector. Denoting final time by t_f, the time-variable, t, could be normalized by defining a new variable ς such that:

$$t = t_f.\varsigma \tag{25}$$

where $0 \leq \varsigma \leq 1$. The system dynamic, Eq. (7-22), can similarly be reformulated with respect to this normalized time parameter:

$$\frac{dA}{d\varsigma} = t_f \frac{dA}{dt} \tag{26}$$

where A represents any element of the state vector. In the maneuver optimization problem, the allowable ranges of state and control variables are considered as the constraints. Typical values, considered here, are presented in Table 3. In the following, the minimum time maneuvers are performed.

Minimum time pull-up maneuver

The pull-up maneuver is performed in the vertical plane. Assuming that the maneuver initiates with the velocity of 50 m/s at an altitude of 2000 (m), the initial

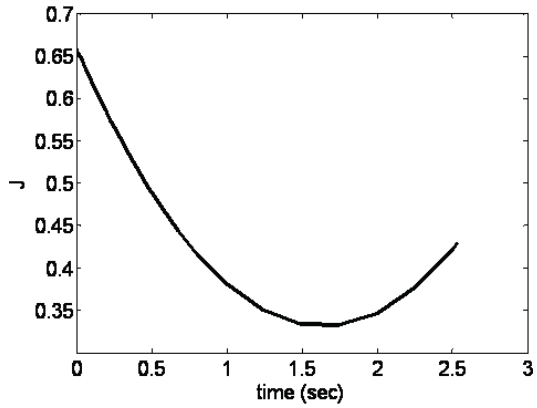

Figure 11. Time variations of advance ratio during a pull-up maneuver.

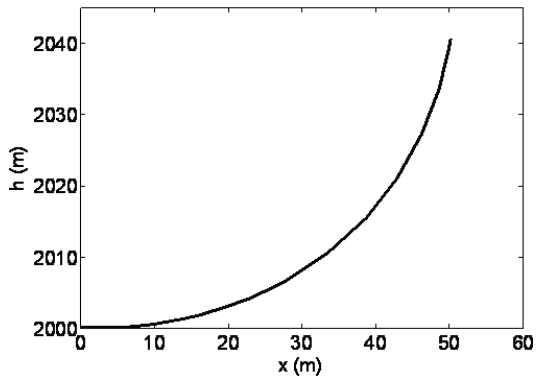

Figure 12. Trajectory during a pull-up maneuver.

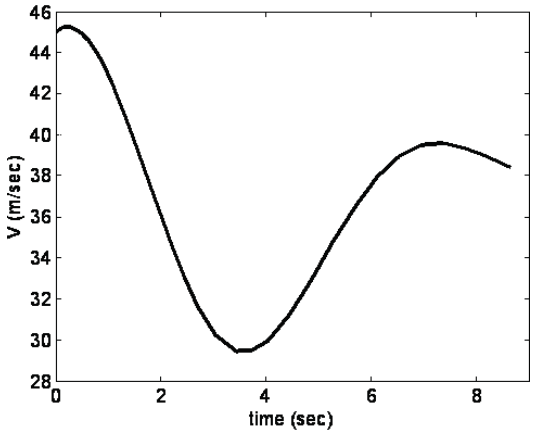

Figure 13. Time variations of velocity during a turn maneuver.

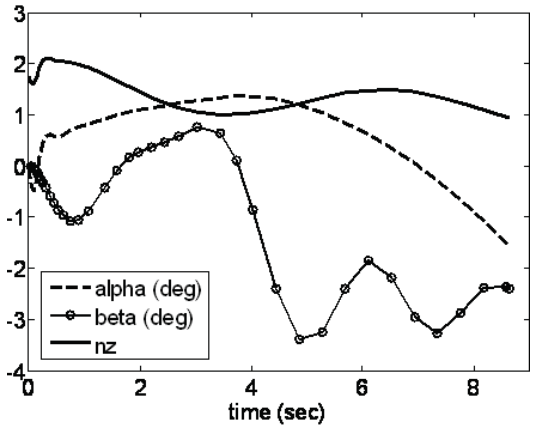

Figure 14. Time variations of angle of attack, side slip and load factor during a turn maneuver.

state vector, Eq. (24), is defined as:

$$\vec{\mathbf{x}}_0 = [50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2000]^{\mathrm{T}}$$

The objective is to start from the above initial conditions and to reach a desired pitch angle of $\theta_d = 80$ (deg) in minimum time. The corresponding cost function is defined as:

$$F = t_f + q_1|\theta_f - \theta_d| \tag{27}$$

where θ_f is the resulting pitch angle at the end of the maneuver and q_1 is a weighting factor that relatively weights the two objectives and is set to 60 in this problem. Since the maneuver is performed in the vertical plane, the lateral controls (aileron and rudder) are not utilized. In addition, three control points are considered for each of the controls (elevator and advance ratio) along the normalized time horizon. Therefore, the number of design variables equals $2 \times 3 + 1 = 7$ (the final time is also a member of the design vector).

Following the implementation of the optimization algorithm, the results obtained after 200 iterations are reported here. The UAV performs the maneuver with a total flight time of t_f =2.54 (sec). Results are sketched in Figures (10-15). The rational and predictable decreasing behavior of the velocity is clear from Figure 7. As Figure 7 shows, the angle of attack varies though its allowable range. The resulting vertical load factor, formulated by Eq. (28), is also drawn in Figure 8 where the maximum load factor becomes 2.71 during the maneuver.

$$n_z = \frac{L + T\sin(\alpha)}{mg} \tag{28}$$

Variations of the pitch rate and pitch angle are drawn in Figure 9. The pitch angle converges to $\theta_d = 80$ (deg) at the end of the maneuver. The continuous optimal elevator and advance ratio time histories are shown in Figure 10 and Figure 11, respectively. As expected, the elevator has negative values that reflect pitch up of the UAV. The pull-up trajectory is drawn in Figure 12.

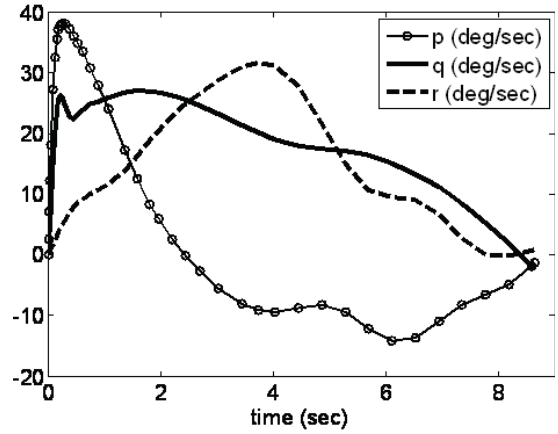

Figure 15. Time variations of roll, pitch and yaw rate during a turn maneuver.

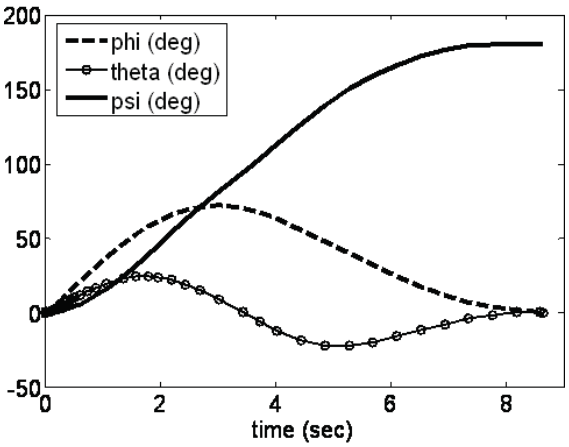

Figure 16. Time variations of roll, pitch and yaw angles during a turn maneuver.

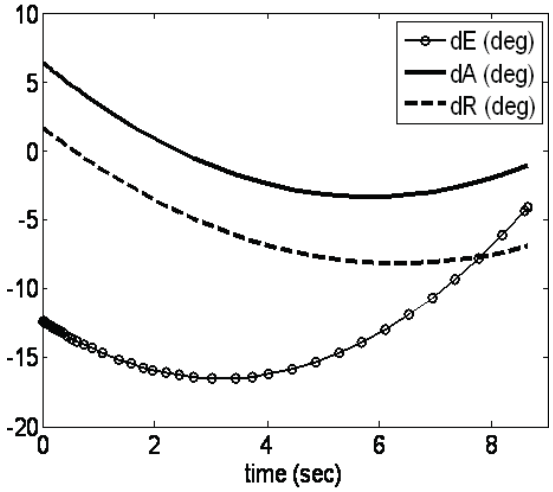

Figure 17. Time variations of elevator, aileron and rudder during a turn maneuver.

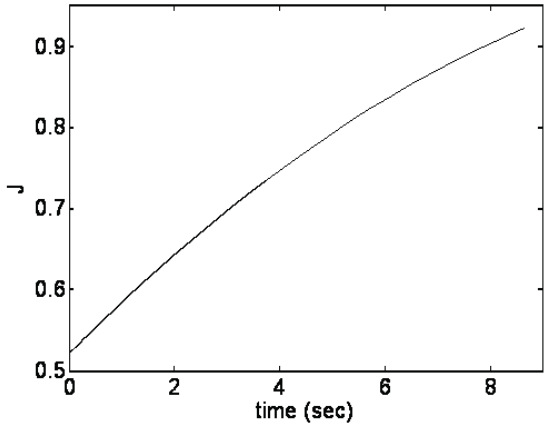

Figure 18. Time variations of advance ratio during a turn maneuver.

Minimum time turn maneuver

The objective of this maneuver is to perform a minimum time 180 (deg) turn. Therefore, the cost function is defined as:

$$F = t_f + q_1|\phi_f - \phi_d| + q_2|\theta_f - \theta_d| + q_3|\psi_f - \psi_d| + q_4|h_f - h_d| \quad (29)$$

where the parameters ϕ_f, θ_f, ψ_f and h_f, are the final values of roll, pitch and yaw angles, and the altitude, respectively. The desired values of these parameters are also selected as $\phi_d = 0$, $\theta_d = 0$, ψ_d =180 (deg) and h_d =2000 (m). The weighting coefficients are also taken as $q_1 = q_2 = q_3$ =60 and q_4 =1. Assuming the maneuver starts from a velocity of 45 (m/sec) at an altitude of 2000 (m), the initial state vector can be given as:

$$\vec{x}_0 = [45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2000]^{\mathrm{T}} \quad (30)$$

For the turning maneuver, all control inputs (elevator, aileron, rudder and advance ratio) will be activated. Again, three equally spaced control points are selected for each control variable. Thus, the design vector dimension is now equal to $4 \times 3 + 1 = 13$.

By solving this optimization problem with the constraints of Table 3, the time optimal turning maneuver will be obtained. The optimal maneuver time turns out to be 8.65 (sec) and the final values of the Euler angles and the final altitude are $\phi_f = -0.009$ (deg), $\theta_f = -0.04$ (deg), $\psi_f = 180.003$ (deg), and $h_f = 2000.03$ (m). This indicates that the algorithm is well able to converge to the desired values. The variation of the velocity during the maneuver is shown in Figure 13. The time history of angle of attack, side slip, and load factor are sketched in Figure 14. Both the angle of attack and side slip angle vary within their feasible ranges. The maximum normal load factor exerted to the UAV in this maneuver is 2.1.

The optimal roll, pitch, and yaw rates can be seen from Figure 15. As Figure 16 shows, the roll angle increases from zero to 72 degrees where the aircraft has reached the midway of its turn (yaw angle of 90 deg.) and then decreases back to zero. Semi oscillating variations of pitch angle and its convergence to the desired value of zero at the end of maneuver can also be seen from Figure 16. Yaw angle behavior during the maneuver and completion of 180 (deg) turn is shown in Figure 16. The time optimal history of elevator, aileron and rudder deflection angles are presented in Figure 17. The optimal advance ratio is sketched in Figure 18. Three dimensional trajectory of RASCAL UAV is also shown in Figure 19. As seen in this figure, the UAV, initially, has a pitch up trend and then lowers its altitude and reaches its initial altitude. This maneuver is very similar to the Candle maneuver [26].

CONCLUSION

A new constrained optimization algorithm is developed, within which the strategies of the quadratic interpolation, random inertia weight and randomized velocity of saturated particles are utilized in order to

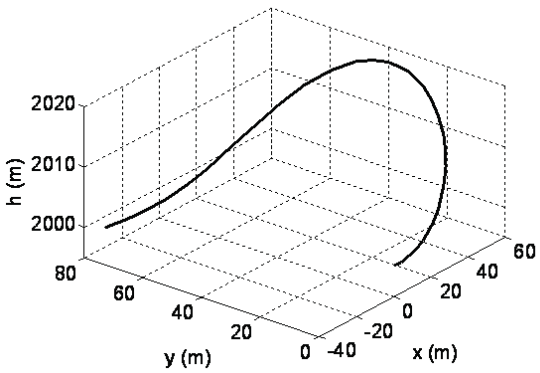

Figure 19. 3D trajectory for a turn maneuver.

enhance the exploration and exploitation abilities of the algorithm. Moreover, the feasibility tournament is used to handle the constraints and to enrich the exploration on the search space, especially in border areas. Trimmability ranges of a UAV are analyzed in various flight conditions within the flight envelope. One of the interesting results is a new performance envelope within the conventional flight envelope which dictates that by considering complete nonlinear dynamics, a flight vehicle may not be trimmable within its entire H-V envelope. For the optimal pull-up and turn maneuvers, using a finite number of control points and cubic spline interpolation method, an optimal continuous time history is determined for each control signal. The introduced methodology not only uses the nonlinear 6DOF model of the vehicle, but also considers all the physical and flight constraints. Implementing the new approach, in comparison with the conventional methods, based on point mass model, has brought about a more realistic performance characteristic, that is vital for aircraft design and analysis applications.

REFERENCES

1. Roskam J., Edward Lan C.T., *Airplane Aerodynamics and Performance*, DAR Corporation, (1997).
2. Anderson J.D., *Aircraft Performance and Design*, WCB/MC Graw Hill , (1999).
3. Rao S.S., *Engineering Optimization*, John Willy & Sons Ins., (1996).
4. Stevens B.L., Lewis F.L., *Aircraft Control and Simulation*, John Whiley & Sons Inc., (1992).
5. Well K.H., Faber B. and Berger E., "Optimization of Tactical Aircraft Maneuvers Utilizing High Angles of Attack", *AIAA, J. of Guidance, Control, Dynamics*, **5(2)**, (1982).
6. Arkadi L., Eliezer K. and Benjamin G., "Minimum-Time Maneuvers of Thrust-Vectored Aircraft", *Journal of Control, and Dynamics*, **21(2)**, PP 244-250 (1998).
7. Pourtakdust S.H., Karimi J., "Optimal Maneuver Tracking in Closed Air Combat Situations", *13th Annual Conference on Mechanical Engineering-ISME2005*, Isfahan University of Technology, Isfahan, Iran , (2005).
8. Engelbrecht A.P., *Fundamentals of Computational Swarm Intelligence*, John Willey & Sons, Ltd , (2005).
9. Pourtakdoust S. H., Nobahari H., "An Extension of Ant Colony System to Continuous Optimization Problems", *Lecture Notes in Computer Science, 3172*, PP 294-301(2004).
10. Yamada Y., Ookoudo K., and Koruma Y., "Layout Optimization of Manufacturing Cells and allocation Optimization of Transport Robots in Reconfigurable Manufacturing Systems using Particle Swarm Optimization", *In Proceeding of the IEEE/RSJ International Conference, the Intelligent Robots and Systems*, **2**, PP 2049-2054(2003).
11. El-Gallad A., El-Hawary M., and Sallam A., "Swarming of intelligent Particle for Solving the Nonlinear Constrained Optimization Problem", *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, **9(3)**, (2001).
12. Zhang W., Xie X. and Bi D., "Handling Boundary Constraints for Numerical Optimization by Particle Swarm Flying in Periodic Search Space", *Proceedings of Evolutionary Computation Congress*, Portland Marriott Downtown, **2**, PP 2307-2311 (2004).
13. Akhtar S., Tai K. and Ray T., "A Socio-Behavioral Simulation Model for Engineering Design Optimization", *Engineering Optimization*, **34(4)**, PP 341-354 (2002).
14. Pulido G.T. and Coello C.A., "A Constraint-Handling Mechanism for Particle Swarm Optimization", *Proceedings of IEEE Congress on Evolutionary Computation*, **2**, PP 1396-1403(2004).
15. Deb K., "An Efficient Constraint Handling Method for Genetic Algorithms", *Computer Methods in Applied Mechanics and Engineering*, **186(2-4)**, (2000).
16. Zavala A.M., Aguirre A.H., and Diharce E.V., "Constrained Optimization Via Particle Evolutionary Swarm Optimization Algorithm (PESO)", *Proceedings of Conference on Genetic and Evolutionary Computation*, PP 209-216(2005).
17. Zavala A.M., Aguirre A.H., and Diharce E.V., "Robust PSO-Based Constrained Optimization by Perturbing the Particle's Memory", *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization ,I-Tech Education and Publishing*, PP 57-76(2007).
18. Kennedy J., Eberhart R. C., "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, PP 1942-1948(1995).
19. Eberhart R. C., Shi Y., "Tracking and optimizing Dynamic Systems with Particle Swarms", *In Proceedings of IEEE Congress on Evolutionary Computation*, **1**, PP 91-100 (2001).
20. Peng J., Chen Y., Eberhart R.C., "Battery Pack State of Charge Estimator to Design using Computational Intelligence Approaches", *In Proceedings of the Annual Battery Conference on Applications and Advances*, PP 173-177(2000).
21. Eberhart R.C., Simpson P.K. and Dobbins R.W., *Computational Intelligence PC Tools*, Academic Press Professional, First Edition , (1996).
22. Kennedy J., Mendes R., "Population Structure and Particle Swarm Performance", *Proceedings of the 2002 Congress on Evolutionary Computation*, PP 1671-1676(2002).
23. Van Den Bergh F., "An Analysis of Particle Swarm Optimizers", Ph.D. Thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, (2002).

24. Jodeh N.M., "Development of Autonomous Unmanned Aerial Vehicle Research Platform: Modeling, Simulating, and Flight Testing", M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, (2006).

25. Slotine J.J., Li W., *Applied Nonlinear Control*, Prentice-Hall International Inc., (1991).

26. Walker J.R., *Air Superiority Operations*, Brassey[s Air Power: Aircraft Weapons Systems and Technology Series, Volume 5, UK, (1989).