

Loughborough University
Institutional Repository

*Optimization-based safety
analysis of obstacle
avoidance systems for
unmanned aerial vehicles*

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: SRIKANTHAKUMAR, S., LIU, C. and CHEN, W-H, 2012. Optimization-based safety analysis of obstacle avoidance systems for unmanned aerial vehicles. *Journal of Intelligent and Robotic Systems*, 65 (1-4), pp. 219 - 231

Additional Information:

- This article was published in the *Journal of Intelligent and Robotic Systems* [© Springer Science+Business Media B.V.]. The definitive version is available at: <http://link.springer.com/article/10.1007%2Fs10846-011-9586-0?LI=true#>. The original publication is available at www.springerlink.com.

Metadata Record: <https://dspace.lboro.ac.uk/2134/11136>

Version: Accepted for publication

Publisher: © Springer Science+Business Media B.V.

Please cite the published version.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Optimization-based Safety Analysis of Obstacle Avoidance Systems for Unmanned Aerial Vehicles

Sivaranjini Srikanthakumar, Wen-Hua Chen

Department of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU, UK

sivaranjinisk@yahoo.co.uk, W.Chen@lboro.ac.uk

Abstract: The integration of Unmanned Aerial Vehicles (UAVs) requires new methods to certify collision avoidance systems. This paper presents a safety clearance process for obstacle avoidance systems where worst case analysis is performed using optimization based approaches under all possible parameter variations. The clearance criterion for the UAV obstacle avoidance system is defined as the minimum distance from the aircraft to the obstacle during the collision avoidance manoeuvre. Local and global optimization based verification processes are developed to automatically search the worst combinations of the parameters and the worst-case distance between the UAV and an obstacle under all possible variations and uncertainties. Based on a simplified 4 Degree of Freedom (4DoF) kinematic and dynamic model of a UAV, the path planning and collision avoidance algorithms are developed in 3D. The artificial potential field method is chosen as a path planning and obstacle avoidance candidate technique for verification study as it is a simple and widely used method. Different optimisation algorithms are applied and compared in terms of the reliability and efficiency

Keywords— Clearance process, Obstacle avoidance, Optimization, Potential field method, Unmanned Aerial Vehicle.

1. Introduction

Due to the absence of a pilot, the use of UAVs has become increasingly popular in military and civilian applications. Path planning of UAVs with known and unknown obstacles is considered as one of the key enabling technologies in unmanned vehicle systems. Indeed, a significant amount of research has been devoted to this subject in recent years. In addition to offering better performance, the main industrial concern related to new methods is to reduce the risk of collisions in the presence of all possible parameter variations and various failure conditions. Therefore, all proposed collision avoidance algorithms have to be verified under all operational conditions and variations that may be experienced during the life of the UAVs. The objective of this paper is to develop advanced algorithms to support safety-critical obstacle avoidance systems in UAVs. The certification process essentially aims at providing the evidence in order to certify that the aircraft is safe to fly in the presence of obstacles and parameter variations. This task is a very time consuming and expensive process, particularly for high performance aircraft [1].

Without a pilot, computer algorithms must be developed to generate a feasible path in real time. Depending on the operation scenarios, there are different kinds of path planning methods. The UAV has to find a collision-free path between the departure and the destination configurations in a static and a dynamic environment containing various obstacles. Several

algorithms have been applied to path planning for UAV in the presence of known obstacles. In [2], the combined method of ray tracing and limit cycle navigation is used for UAV in 2D and 3D. Griffiths et al. use the rapidly exploring random tree (RRT) based path planner through 3D environment for an autonomous aerial vehicle [3]. In [4], both probabilistic roadmap-based and RRT algorithms are used for generating 3D collision free path for an autonomous helicopter. Bortoff presents a collision free path planning method using Voronoi graph search method [5], whereas a model predicative control based trajectory optimization method is used to avoid obstacle for Nap-of-the-Earth flight in [6]. UAV motion planning techniques based on potential field functions have been extensively studied; e.g. [7, 8]. UAV path planning using the artificial potential field method will be used as a candidate collision avoidance technique for safety assessment.

Three major concerns in regard to autonomous vehicle operation are efficiency, safety and accuracy. As the safety of autonomous vehicles is dependent on control systems and obstacle avoidance algorithms, it must be proven that the control systems and obstacle avoidance algorithms function correctly in the presence of all possible vehicle and environmental variations. Two particular difficulties faced by designers are a mismatching between the model used for algorithm development and the real vehicle dynamics, and various uncertainties in vehicle operations. To simplify the process of the algorithm development, in general a much simplified model that captures the main characters of the vehicle is used in the design stage under a number of assumptions or simplifications. This causes the mismatching between the model used in the design and real vehicle behaviour. Furthermore, the variations of the autonomous vehicle dynamics in operation may arise due to the changes of the vehicle itself (e.g. the change of mass or the centre of gravity) or the change of the operational environment. Assessment of the safety must be performed not only on the nominal model, but also for all possible vehicle and environmental variations, and in the presence of the mismatching between the model used for the design and the real vehicle. Therefore, techniques and procedure are demanded to understand the behaviour of the UAVs in the presence of such uncertainties. They must cover all possible combinations of UAV parameters so that guarantee the worst-case performance is adequate, which is particularly important for safety critical functions such as collision avoidance.

Fault Tree Analysis method was applied to the TCAS (Traffic Alert and Collision Avoidance Systems) for UAVs safety analysis in [9], while Failure Modes and Effects Analysis (FMEA) was used [10]. In [11], Functional Failure Analysis (FFA) was performed for safety analysis of UAV operation including collision avoidance. Two critical hazards in UAV operation were defined in this analysis: midair collision and ground impact. Obstacle Analysis was applied to rotorcraft UAV in [12], where potential side effects and missing monitoring and control requirements were identified by step-by-step use of the obstacle analysis technique. In [13], the Markov Decision Process and Observable Markov Decision Process solvers was proposed to generate avoidance strategies optimizing a cost function that balances flight-plan deviation with anti-collision. The performance of this collision avoidance system was evaluated using a simulation framework developed for TCAS studies. A framework for provably safe decentralized trajectory planning of multiple aircraft was presented in [14]. Each aircraft plans its trajectory individually using a receding horizon strategy based on mixed integer linear programming.

In this study, the minimum distance from the aircraft to an obstacle during a collision avoidance manoeuvre is chosen as the criterion for the performance assessment. To successfully perform collision avoidance manoeuvres, the minimum distance to the obstacle

(d_{min}) must be greater than the radius of the obstacle (r) including a safe margin. The worst case analysis in the presence of all the possible uncertainties is cast as a problem to finding the combinations of the variations where the minimum distance to the obstacle (d_{min}) appears. Instead of exhaustive searching the worst cases as in many practices, it is well known that optimization can effectively find a (local) minimum or maximum without evaluating all possible parameters in a solution space [15]. In this paper, the worst case analysis for collision avoidance algorithms is treated as a constrained nonlinear optimization problem with simulation being involved in each iteration. In order to pass the safety assessment of an anti-collision system, d_{min} in the worst cases must be greater than r ($d_{min} > r$) in the presence of all possible uncertain parameter variations. Otherwise, the obstacle avoidance algorithm and associated controllers have to be redesigned to satisfy the anti-collision specification. The proposed approach in this paper is applied to the collision avoidance algorithm using with artificial potential field methods for a simple UAV model, however, the basic idea is applicable for other unmanned vehicles with collision avoidance algorithms developed by other methods. It shall be highlighted that it is not the intention of this paper to refine or develop a collision avoidance based on artificial potential field methods. Instead, it is to develop a new procedure to support the safety assessment of an existing collision avoidance algorithm for UAV operating in 3D environment.

The rest of the paper is organized as follows: the simplified kinematic and dynamic model of a 4 DOF UAV is introduced in Section 2. The clearance criterion for obstacle avoidance is then discussed in this section. Motion planning in 3D and collision avoidance algorithms is designed in Section 3 using the artificial potential field method. In Section 4, the obstacle avoidance algorithm is validated at nominal parameters. In Section 5, initial robustness analysis of the collision avoidance algorithm is carried out. The optimization based verification process is introduced and local optimization algorithms are first presented. Two stochastic global optimization algorithm based verification processes are developed in Section 6. One is genetic algorithms (GA) and the other GLOBAL algorithm. Furthermore, in order to guarantee finding the worst-cases, a deterministic global optimization method, i.e. Dividing RECTangles (DIRECT), is applied to the worst case analysis of the collision avoidance algorithm in Section 7. Simulation results are presented to verify the proposed verification processes. Finally, Section 8 concludes the paper and outlines future research directions.

2. UAV MODEL AND CLEARANCE CRITERION

A. UAV Model

In order to present a clearance criterion of obstacle avoidance systems, first a simplified UAV model is considered for the algorithm development and assessment. 4 DOF mass point model of the UAV with seven states in 3D environment is considered in this study. The kinematic and dynamic model of UAV is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ v_z \\ \omega \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/m & 0 & 0 \\ 0 & 1/J & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (1)$$

where $q=[x, y, z]^T$ represents the x, y and z position of the UAV and $[v, v_z]^T$ represents velocity vector where v is the linear velocity in x - y direction and v_z is the velocity in z direction. θ represents the planar orientation of the UAV, ω is the angular velocity of the UAV. The control input vector is $u = [u_1, u_2, u_3]^T$ [7].

B. Clearance Criterion

The UAV has to find a collision-free path between the starting and the goal configurations in a static environment containing various obstacles. During the UAV flight path, spherical obstacles are considered in this study. For the UAV's safe path, the minimum distance to the obstacle (d_{min}) limit exceeding criterion is defined as clearance criterion in the time domain. There are two cases of UAV's path is considered: one is at level flight and other one is at different altitude.

In Fig.1.(a), the UAV altitude is assumed to be kept at a constant value of h_0 m from the ground level. UAV starts from an initial point with initial linear velocity of V_0 and goes to the goal position. If the UAV departure and destination points are at different altitude (Fig.2.(b)), then the motion planning algorithm includes the height (z), i.e. in 3D. UAVs typically fly at much faster speed than ground robots. Therefore, it requires faster and more accurate decision making. In order to maintain safety, obstacle avoidance state can be defined as shown in Fig.1. The influence range for an obstacle is determined from the radius of the obstacle and a specified safe margin. The specified safe margin can be chosen according to the UAV's dimensions. As mentioned above, the artificial potential field method is used for developing obstacle avoidance algorithms under investigation in this paper. In this framework, obstacle avoidance manoeuvre can be performed within the repulsive potential field influence range. In this obstacle avoidance state, UAV is repulsed from the obstacle and attracted to its goal position.

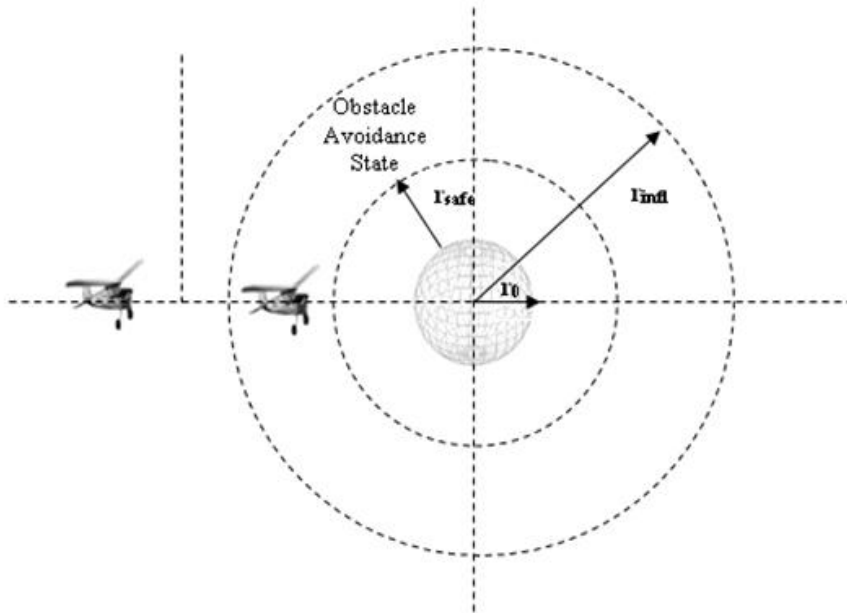


Fig. 1(a). Obstacle avoidance clearance criterion

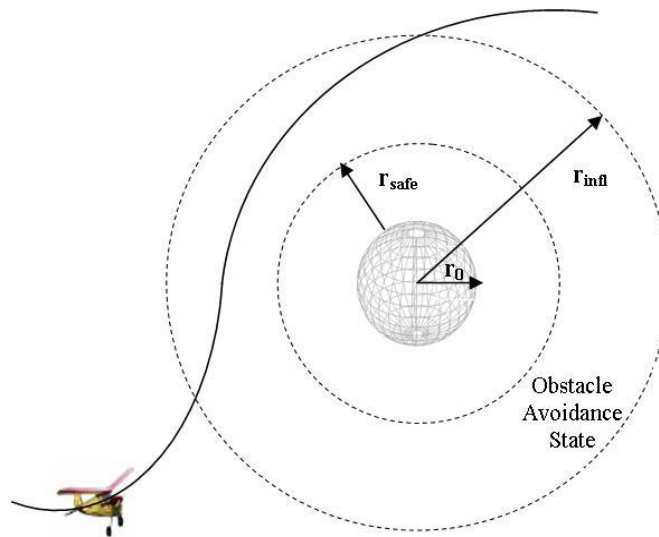


Fig. 1(b). Obstacle avoidance clearance criterion

For a spherical obstacle, the influence range is chosen as the radius of r_{infl} which is greater than the radius of the obstacle (r_0) and the safe margin (r_{safe}). Let $r = r_0 + r_{safe}$, then the anti-collision condition is defined as $d_{min} > r$. In the obstacle avoidance clearance process, all violations of these clearance criteria must be found and the worst-case result for each criterion computed. The corresponding worst-case combination of uncertainties must also be computed.

2. MOTION CONTROL AND OBSTACLE AVOIDANCE

Figure 2 provides an overview of the control law architecture. The control system is proposed to have an inner-outer-loop structure. For the mission, the UAV is maintained in a constant altitude of h_0 m and in a different altitude h m. The inner-loop control law is responsible to compute the input signals that drive the motors to force the UAV to fly at a desired linear and angular velocity. These desired velocities are the control signals generated by the outer-loop controller [16].

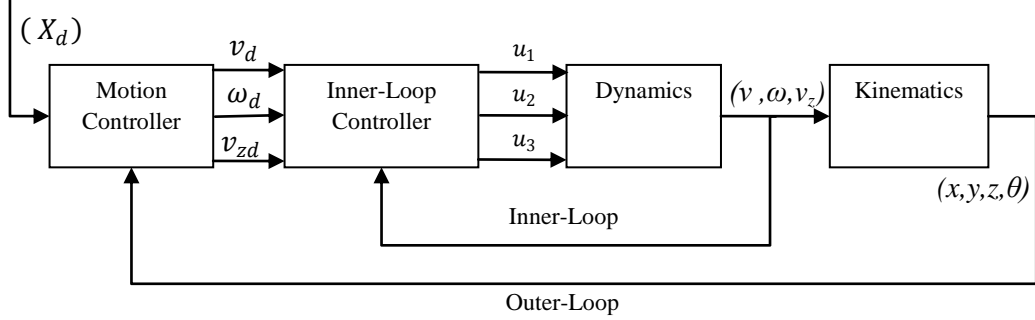


Fig. 2. Model of the mobile-robot including kinematics, dynamics and the controllers

A. Inner-Loop Controller

To accomplish the goal of driving the UAV flying at a desired linear velocity v_d and v_{zd} , and angular velocity ω_d , the first step is to compute the error between the true velocities and the desired ones. To this effect, let $e_v = (v_d - v)$, $e_{vz} = (v_{zd} - v_z)$, and $e_\omega = (\omega_d - \omega)$ be respectively the linear and angular velocity errors. A simple proportional integral control law is proposed as speed controllers.

$$\begin{aligned}
 u_1 &= K_1 e_v + K_4 \int_0^t e_v(\tau) d\tau \\
 u_2 &= K_2 e_\omega + K_5 \int_0^t e_\omega(\tau) d\tau \\
 u_3 &= K_3 e_{vz} + K_6 \int_0^t e_{vz}(\tau) d\tau
 \end{aligned} \tag{2}$$

B. Outer-Loop Controller

The incremental motion planning for a nonholonomic UAV is considered in this section. In general, a kinematic model is used for motion planning and collision avoidance.

The kinematic model of the UAV can be represented in a general state space form as

$$\dot{X} = G(X)U \tag{3}$$

where $X \in \mathbb{R}^n$ is the vector of generalized coordinates, and $u \in \mathbb{R}^m$ ($m < n$) is the control input vector [17], [18]. Given any desired smooth trajectory $X_d(t)$, a straightforward approach is to design the input command U using the pseudo-inverse control law

$$U = G^\#(X)\dot{X}_d \quad (4)$$

where $G^\#(X) = [G^T(X)G(X)]^{-1}G^T(X)$ is the pseudo-inverse of $G(X)$.

This solution locally minimizes the error $(\dot{X}_d - \dot{X})$ in a least-squares sense. If the desired velocity \dot{X}_d is feasible at the current X , the control law (4) results in zero velocity error. In order to balance error components, the state X can be pre-weighted or, equivalently, a weighted pseudo-inverse can be used. Note that the pseudo-inverse gives the command input U as a feedback law depending on the current state X [17], [18].

For the UAV, $X=(x, y, z, \theta)$, is the configuration vector. Comparing Eq. (1) and (3) gives

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & 0 \\ \sin\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ v_z \\ \omega \end{bmatrix} \quad (5)$$

Let $U=[v_d, v_{zd}, \omega_d]$. It follows from (4) that the pseudo-inverse of $G(X)$ takes the form

$$G^\# = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Correspondingly, the feedback law (4) for tracking a desired trajectory $X_d=(x_d, y_d, z_d, \theta_d)$ becomes

$$U = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{z}_d \\ \dot{\theta}_d \end{bmatrix} \quad (7)$$

which can be written as

$$v_d = k_p(\dot{x}_d \cos\theta + \dot{y}_d \sin\theta) \quad (8)$$

$$v_{zd} = k_z \dot{z}_d \quad (9)$$

$$\omega_d = k_\theta \dot{\theta}_d \quad (10)$$

where gains k_p , k_z and k_θ are introduced to give additional freedom in weighting the two input commands. This is equivalent to use a weighted pseudo-inverse in (4). In order to apply the control law (8-10), we need to specify the desired values for \dot{x}_d , \dot{y}_d , \dot{z}_d and $\dot{\theta}_d$. \dot{X}_d may be determined by a local holonomic planner with the potential field method as described in the next section.

C. Potential Field Method

The potential field method was first used by Khatib for manipulators and mobile robots path planning in the 1980s [19]. The basic concept of the potential field method is to fill the robot's workspace with an artificial potential field in which the robot is attracted to its goal position and repulsed away from obstacles. The UAV's path planning is, in a sense, similar to the path planning of a mobile robot. The combination of the attractive force to the goal and repulsive forces away from the obstacles drives the UAV in a safe path to the goal.

Let $q=(x, y, z)$ denote the UAV current point in the air space. The usual choice for the attractive potential is the standard parabolic that grows quadratically with the distance to the goal,

$$U_{att}(q) = \frac{1}{2} k_a d_{goal}^2(q) \quad (11)$$

where $d_{goal} = \|q - X_{goal}\|$ is the Euclidean distance of the UAV's current position q to the goal X_{goal} and k_a is a scaling factor [20]. The gradient is calculated by

$$\nabla U_{att}(q) = k_a(q - X_{goal}) \quad (12)$$

The attractive force considered in the potential field based approach is the negative gradient of the attractive potential

$$F_{att}(q) = -\nabla U_{att}(q) = -k_a(q - X_{goal}) \quad (13)$$

Setting the vehicle velocity vector proportional to the vector field force, the force $F_{att}(q)$ drives the UAV to the goal with a velocity that decreases when the UAV approaches the goal.

The repulsive potential keeps the vehicle away from obstacles. This repulsive potential is stronger when the UAV is closer to the obstacles and has a decreasing influence when the UAV is far away. A possible repulsive potential generated by obstacle i is

$$U_{rep_i}(h) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{d_{obst_i}(h)} - \frac{1}{d_0} \right)^2, & \text{if } d_{obst_i}(h) \leq d_0 \\ 0 & \text{if } d_{obst_i}(h) > d_0 \end{cases} \quad (14)$$

where i is the number of obstacle that are close to the UAV, $d_{obst_i}(q)$ is the closest distance to the obstacle i , k_r is a scaling constant and d_0 is the obstacle influence threshold. The negative gradient of the repulsive potential, $F_{rep_i}(q) = -\nabla U_{rep_i}(q)$, is given by,

$$F_{rep_i}(q) = \begin{cases} k_r \left(\frac{1}{d_{obst_i}(q)} - \frac{1}{d_0} \right) \frac{1}{d_{obst_i}^2(q)} \hat{e}_i, & \text{if } d_{obst_i}(q) \leq d_0 \\ 0 & \text{if } d_{obst_i}(q) > d_0 \end{cases} \quad (15)$$

where $\hat{e}_i = \frac{\partial d_{obst_i}(q)}{\partial(q)}$ is a unit vector that indicates the direction of the repulsive force [21].

Therefore,

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{z}_d \end{bmatrix} = -\nabla (U_{att}(q) + U_{rep}(q)) = F_{att}(q) + F_{rep_i}(q) \quad (16)$$

In order to complete the planning method, the rotational part of $\dot{\theta}_d$ is defined. For an aircraft, it is convenient to use

$$\dot{\theta}_d = atan2\left\{\frac{\dot{y}_d}{\dot{x}_d}\right\} - \theta \quad (17)$$

By defining $atan2\{0,0\}=0$, the above function remains continuous along any approaching direction to the goal [17]. Therefore, the resulting command u_1 , u_2 and u_3 are determined by Eq.(8-10) using with Eq.(16-17).

4. COLLISION AVOIDANCE VALIDATION AT NOMINAL CASE

In this section, the proposed collision avoidance algorithm and controller are validated at the nominal parameters. The simulation results for a UAV approaching a spherical obstacle are presented at the nominal parameters. The nominal parameter values are $m=2kg$ and $J=0.02kg.m^2$. The initial linear velocity V_0 is $15 m/s$ for the nominal case. The influence range is also defined with a radius (r_{infl}) $26m$ and safe margin is chosen as $7m$. The PI controller gains are chosen as $K_1=40$, $K_2=45$, $K_3=8$, $K_4=0.05$, $K_5=0.05$, and $K_6=0.02$. Controller gains $k_p=0.0085$, $k_\theta=80$ and $k_z=0.03$, and holonomic planner parameters $k_a=27$ and $k_r=75$ are tuned and set to fixed values for the verification process.

The different altitudes of the departure and destination points are considered in this case. The destination position is located at $(200, 200, 70)m$. The initial departure point is $(0, 0, 20)m$. The spherical obstacle is located at $(100, 100, 50)m$ with a radius (r_0) of $15m$. Therefore, the safety radius is $22m$ including safe margin. The simulation result at the nominal parameters is shown in Fig. 3. The minimum distance to the obstacle is obtained as $22.1773m$ which is greater than obstacle safety radius $22m$ ($d_{min}>r$). This concludes that the obstacle avoidance algorithm works correctly at the nominal parameters. The optimization based clearance approach will be applied to this proposed obstacle avoidance algorithm in 3D, i.e, for different altitude.

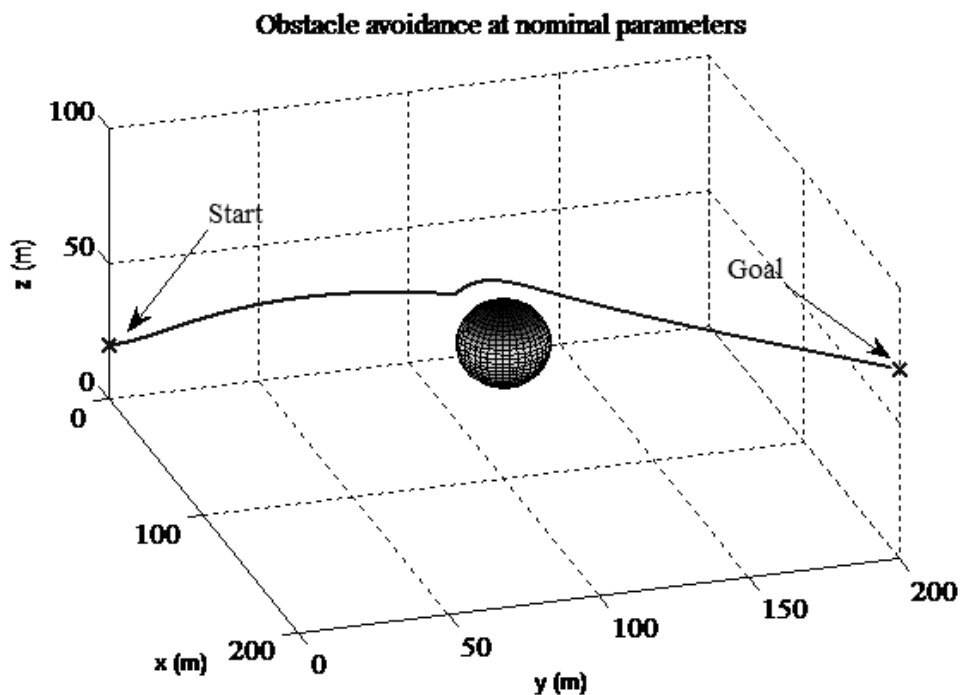


Fig. 3. Simulation result for UAV collision avoidance at nominal parameters

The proposed controller is validated for level flight. The UAV performs level flight at a height of $h_0=50m$ from the ground level. The target position is located at $(200, 200, 50)m$ and the UAV starts from an initial point $(0, 0, 50)m$. The spherical obstacle is located at $(100, 100, 50)m$ with a radius (r_0) of $15m$. The simulation result at the nominal parameters is shown in Fig. 4. It is interesting to observe that the UAV performs a lateral manoeuvre under the collision avoidance algorithm, which is different from the first case where the UAV climbs over the obstacle to avoid collision. The minimum distance to the obstacle is obtained as $22.1007m$ which is greater than obstacle safety radius $22m$ ($d_{min}>r$).

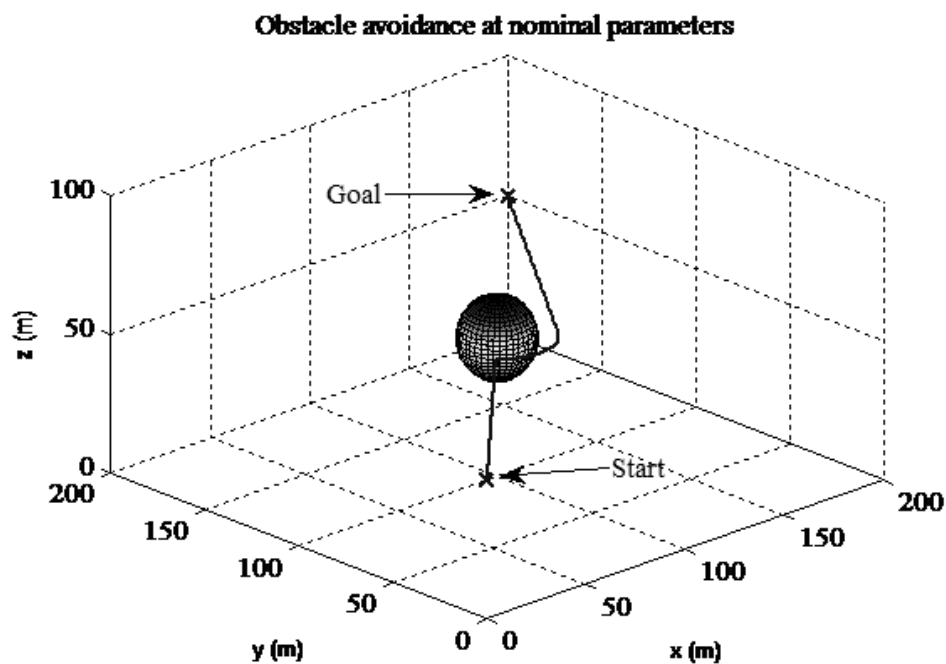


Fig.4. Simulation result for UAV collision avoidance at nominal parameters

5. Initial Robustness Analysis and Local Optimization Method

Initial robustness analysis of the proposed algorithm is firstly carried out. Uncertainties are considered in the dynamic model (mass and inertia), and each uncertain parameter is allowed to vary within $\pm 30\%$ of its nominal value. Initial linear velocity is also considered as an uncertain parameter within the range of $10\text{-}20\text{ m/s}$. These are firstly considered within lower and upper bounds, i.e. $m = [1.4, 2.6]\text{ kg}$, $J = [0.016, 0.024]\text{ kgm}^2$, and $V_0 = [10, 20]\text{ m/s}$. Fig. 5 - 7 show variations of the minimum distance to the obstacle with respect to the initial velocity, mass and inertia. There is a small variation in the distance with the variations of the initial velocity, but in a nonlinear form, whereas the minimum distance to the obstacle monotonically decreases with the increase of the mass and inertia.

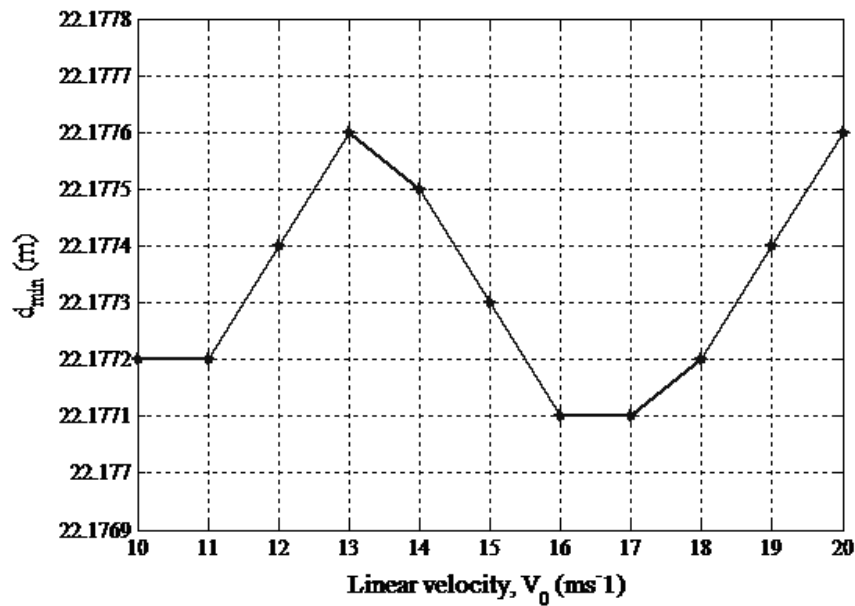


Fig.5. Linear velocity variations

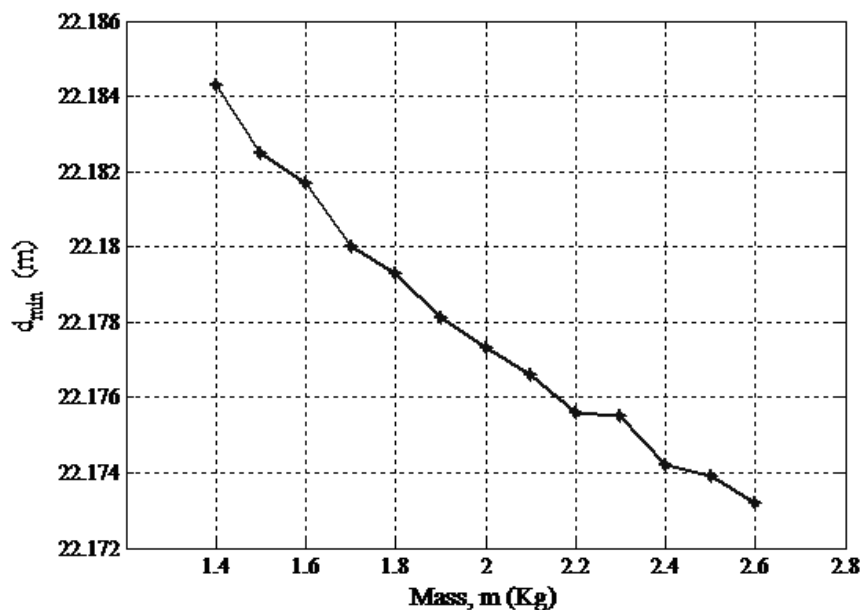


Fig.6. Mass variations

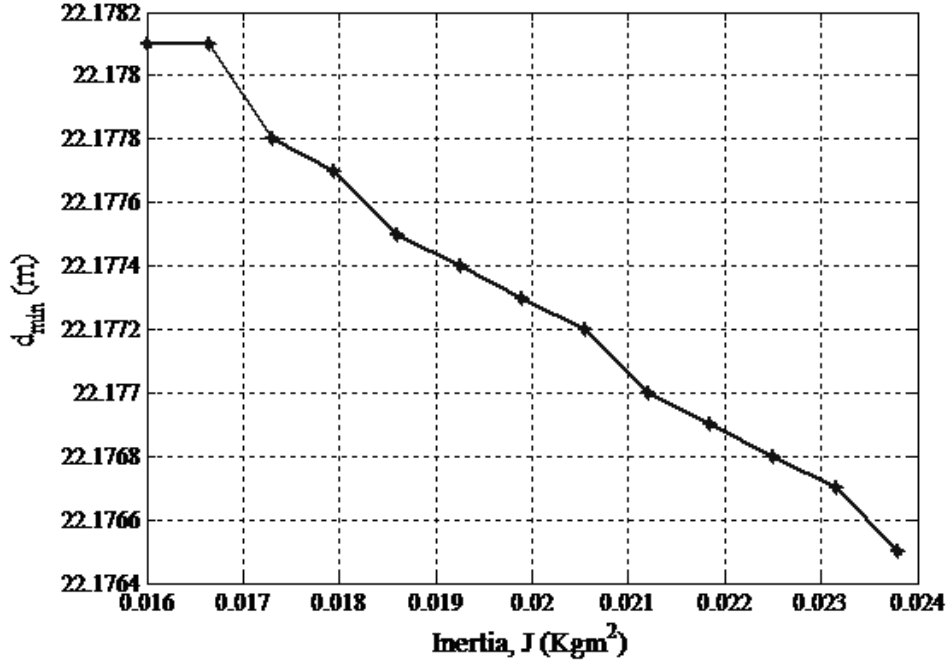


Fig.7. Inertia variations

5.1. Optimization-based Worst-Case Analysis

In this paper, the optimization clearance process is applied to the UAV obstacle avoidance systems. If the minimum distance to the obstacle is greater than a safety radius of obstacle ($d_{min} > r$) during the UAV moving, then the proposed anti-collision algorithm is safe. When the optimization clearance process is applied to the system, this anti-collision condition is checked for all possible variations. The local and global optimization methods are applied to the problem of evaluating a worst-case condition and parameters for the UAV collision avoidance systems. Uncertain parameters are considered that lies between given upper and lower bounds. The objective function is

$$d_{min} = \min(d(t)) \quad \text{for } t \leq T \text{ (sec)} \tag{18}$$

$$s.t \quad P_L \leq P \leq P_U$$

where P is the uncertain parameters set. P_L and P_U are the lower and upper bounds of P . $d(t)$ is the distance to the obstacle, T is the collision avoidance manoeuvre during the period and d_{min} is the minimum distance to the obstacle.

5.2 Local Optimization-based Worst-Case Analysis

Sequential Quadratic Programming (SQP) methods are standard general purpose algorithms for solving smooth and well-scaled nonlinear optimization problems when the functions and gradients can be evaluated with high precision. It is an iterative method starting from an initial point and converging to a local minimum. The function *fmincon* is a MATLAB implementation. The optimization processing of *fmincon* consists of three main stages:

(i) updating of the Hessian matrix of the Lagrangian function, (ii) quadratic programming problem solution, and (iii) line search and merit function calculation. This iteration is repeated until an optimal or feasible solution is found [22]. The local optimization method is applied with different starting points to the problem of evaluating a clearance criterion for the UAV obstacle avoidance systems.

This iteration is repeated until a specified termination criterion (either maximum number of function evaluations or convergence accuracy) is met. The results of the minimum distance to the obstacle and worst case parameters with different starting points are given in Table.1. The results clearly show that *fmincon* does not give the same solutions for this problem because the solution for a local optimization algorithm depends on the starting point. It does not give the true worst case. Therefore, global optimization methods are applied to find the true worst-case.

TABLE.1. LOCAL OPTIMIZATION RESULTS

Algorithm	Starting point	Convergent point	$d_{min}(m)$
<i>fmincon</i>	[1.5, 0.017, 11.0]	[1.5, 0.017, 11.0]	22.1834
<i>fmincon</i>	[2.5, 0.02, 18]	[2.4447, 0.0239, 17.9642]	22.1734

6. Stochastic Global Optimization-based worst case analysis

A. Genetic Algorithms

Genetic Algorithms (GA's) are general purpose stochastic search and optimization algorithms, based on genetic and evolutionary principles. The theory and practice of the GA was originally invented by John Holland in 1960s and was fully elaborated in his book *Adaption in Natural and Artificial Systems* published in 1975 [23]. The basic idea of the approach is to start with a set of designs, randomly generated using the allowable values for each design variable. Each design is also assigned a fitness value. The process is continued until a stopping criterion is satisfied or the number of iterations exceeds as a specified limit. Three genetic operators are used to accomplish this task: Selection, Crossover, and Mutation. Selection is an operator where an old design is copied into the new population according to the design's fitness. There are many different strategies to implement this selection operator including roulette wheel selection, tournament selection and stochastic universal sampling. The crossover operator corresponds to allowing selected members of the new population to exchange characteristics of their designs among themselves. Crossover entails selection of starting and ending positions on a pair of randomly selected strings, and simply exchanging the string of 0's and 1's between these positions. Mutation is the third step that safeguards the process from a complete premature loss of valuable genetic material during selection and crossover. The foregoing three steps are repeated for successive generations of the population until no further improvement in fitness is attainable [24, 25, 26].

GA can be applied to the UAV collision avoidance system to find the global minimum. The uncertain parameter set is considered here as the genetic representation, i.e. the chromosome. Each of the uncertainties corresponds to one gene. A binary coded string is generated to represent the chromosome, where each of the uncertain parameters lies between the lower and upper bounds. The selection function of roulette wheel is used for this study. The population size and crossover fraction are selected as default value of 20 and 0.8 respectively. The iteration continues until a specified number of generations (50) exceeded. The GA results with different starting points are given in Table.2. The results show that GA almost converges to the same worst case condition at different starting points. Figure.8 shows the number of generations versus the best fitness and the mean fitness values at starting point [1.5, 0.017, 11].

TABLE.2. GA RESULTS FOR A UAV OBSTACLE AVOIDANCE SYSTEM

Algorithm	Starting point	$m(kg)$	$J(kgm^2)$	$V_o(m/s)$	$d_{min}(m)$	Time(minutes)
GA	[1.5, 0.017, 11]	2.5998	0.024	19.7065	22.1722	49.12
GA	[2.5, 0.02, 18]	2.5998	0.024	18.8370	22.1722	54.39

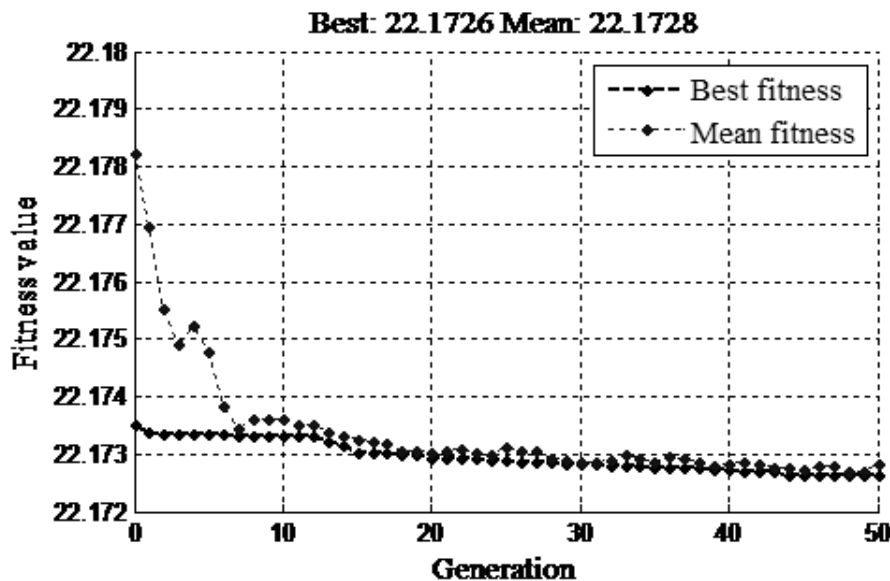


Fig.8. No of generations vs. Fitness value

B. GLOBAL Algorithm

The multistart clustering algorithm presented in this work is based on GLOBAL developed by (Csendes in 1988), which is a modified version of the stochastic algorithm by Boender et al (1982) implemented in FORTRAN. The GLOBAL method has two phases i.e. a global and a local one. The global phase consists of sampling and clustering, while the local phase is based on local searches. A general clustering method starts with the generation of a uniform

sample in the search space (the region defined by lower and upper bounds). After transforming the sample (by selecting a user set percentage of the sample points with the lowest function values), the clustering procedure is applied. Then, the local search is started from those points which have not been assigned to a cluster. GLOBAL uses the Single Linkage clustering rule [27].

The new implementation GLOBALm, which has been written in MATLAB, is freely available for academic purposes. It is the bound constrained global optimization problems with a black-box type objective function. GLOBALm has different local optimization methods which are capable of handling constraints. The UNIRANDI local search method is part of GLOBAL package while the BFGS (Broyden-Fletcher-Goldfarb-Shanno) local search is part of the MATLAB package. GLOBAL has six parameters to set: the number of sample points, the number of best points selected, the stopping criterion parameter for local search, the maximum number of function evaluations for local search, the maximum number of local minima to explore, and the used local method. All these parameters have a default value [27].

The GLOBAL optimization with UNIRANDI local search method is applied to find the global solution for the UAV obstacle avoidance system. The results with different numbers of the sampling points are given in Table.3. GLOBAL algorithm converges to a unique solution with different number of sampling points. It takes 2460 functions evaluation with 100 sampling points while 3380 functions evaluations with 200 sampling points. GLOBAL takes more time for convergence. Therefore, GA is faster than GLOBAL for this case study. However, both of these algorithms cannot guarantee the worst case is found.

TABLE.3. GLOBAL RESULTS FOR UAV OBSTACLE AVOIDANCE SYSTEM

Algorithm	No of SAMPLE	m (kg)	$J(kgm^2)$	V_0 (m/s)	$d_{min}(m)$	Fun.Evalu taken	Time (hours)
<i>GLOBAL-with UNIRANDI</i>	100	2.60	0.0238	19.8243	22.1722	2460	2.18
<i>GLOBAL-with UNIRANDI</i>	200	2.60	0.024	18.8998	22.1722	3380	3.00

7. Deterministic Global Optimization-based worst case analysis

A. DIRECT Method

The disadvantage of the stochastic global optimization methods including GA and GLOBAL algorithms is that there are no formal proofs of convergence. In order to avoid this problem, a deterministic global optimization algorithm known as DIRECT method (DIviding RECTangles) is also considered in the verification process for the obstacle avoidance. The DIRECT algorithm was developed by Jones et al in 1993 [28], which guarantees to the convergence to the globally optimal if the objective function is continuous or at least

continuous in the neighborhood of the global optimum. The global convergence may come at the expense of a large and exhaustive search over the domain. The DIRECT algorithm was created in order to solve difficult global optimization problems with bound constraints and a real-valued objective function. The DIRECT method does not require any derivative information. It is a modification of the standard Lipschitzian optimization method. This global search algorithm can be very useful when the objective function is a “black-box” function. The DIRECT algorithm is described below [24, 29, 30]

Normalization and Division of the Hyper-cube

DIRECT begins the optimization by transforming the domain of the problem into a unit hyper-cube. That is,

$$\bar{\Omega} = \{x \in R^N : 0 \leq x_i \leq 1\} \quad (19)$$

The algorithm works in this normalized space. Let c_1 be the center point of this hypercube and evaluate $f(c_1)$. The next step is to divide this hyper-cube by evaluating the function values at the points $c_1 \pm \delta e_i, i = 1, 2, \dots, N$, where δ is one-third of the side length of the hyper-cube, and e_i is the i th unit vector. That is, a hyper-cube is divided into three hyper-rectangles in each dimension.

The DIRECT algorithm chooses to leave the best function values in the largest space; therefore, the smallest ω_i can be defined as

$$\omega_i = \min\{f(c_1 + \delta e_i), f(c_1 - \delta e_i)\}, \quad 1 < i < N \quad (20)$$

and then divide the dimension with the smallest ω_i into thirds, so that $(c_1 + \delta e_i), i = 1, \dots, N$ are the centers of the new hyper-rectangles. This pattern is repeated for all dimensions on the “centre hyper-rectangle”, choosing the next dimension by determining the next smallest ω_i .

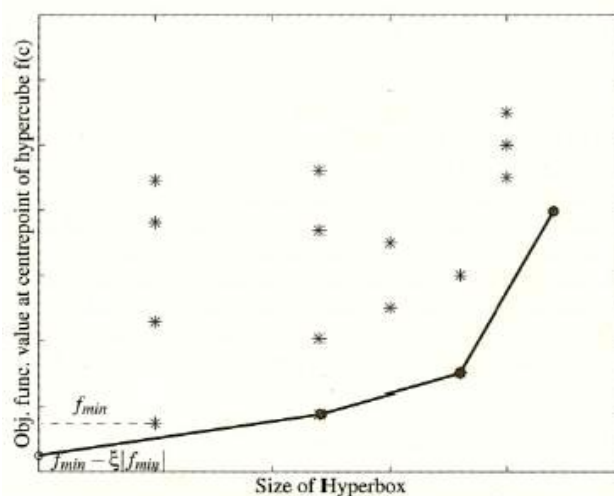


Fig.9. Hyper-rectangles on the piecewise linear curve are potentially optimal [24]

Potentially Optimal Hyper-rectangles

DIRECT then determines which rectangles are potentially optimal, and should be divided in this iteration.

Let $\epsilon > 0$ be a positive constant and let f_{min} be the current best function value. A hyper-rectangle j is potentially optimal if there exists some $K > 0$ such that

$$\begin{aligned} f(c_j) - Kd_j &\leq f(c_i) - Kd_i, \forall_i \text{ and} \\ f(c_j) - Kd_j &\leq f_{min} - \epsilon |f_{min}| \end{aligned} \quad (21)$$

In (21), c_j is the center point of the hyper-rectangle j , and d_j defines a measure for the hyper-rectangles. Jones et al. chose to use the distance from center point c_j to its vertices as the measure and also concluded that a good value for ϵ is 1×10^{-4} . Fig.9 illustrates this definition.

Division of the Hyper-rectangles

Once a hyper-rectangle has been identified as potentially containing the optimal solution, DIRECT divides this hyper-rectangle into smaller hyper-rectangles. DIRECT divides the hyper-rectangles by performing division only in the dimensions with the longest side length. The sequence of the dimensions to be divided is determined by ω_j which is defined as

$$\omega_j = \min_{i \in I} (f(c_i + \delta_i e_j), f(c_i - \delta_i e_j)), \quad j \in I \quad (22)$$

where δ_i is one-third the length of the longest side of hyper-rectangle I , e_j is the j th unit vector, and I is the set of all dimensions of the longest side length. This process is repeated for all dimensions in I .

B. Simulation Results

The DIRECT algorithm is applied to the UAV obstacle avoidance verification process, and the results are given in Table.4. The DIRECT method requires no initial guesses but operates on the parameters upper and lower bounds. The DIRECT algorithm terminates as soon as it exceeds the given iterations. The history of iteration versus fitness value is shown in Fig.10. This figure shows that the fitness value of d_{min} is almost same from iteration 110 to 200. All optimization algorithms are performed in MATLAB 2010a and Intel (R) Core(TM)2 Duo CPU (3.16GHz). DIRECT takes 2 hours to converge to the global minimum. Compared to the stochastic global algorithms, GA performs faster, but there is no confidence to establish the true worst case; GLOBAL algorithm converges to the same global minimum in this study, however it cannot be guaranteed to reach the worst case. DIRECT takes 2283 function evaluations at 200 iterations. The DIRECT algorithm can guarantee finding the worst case in this application, but the computation time is high.

These worst-case condition and worst-case parameters identified in the verification process are further validated with simulation response shown in Fig.11. The time versus distance to the obstacle at the nominal and worst-case parameters is shown in Fig.12. The worst-case minimum distance to the obstacle d_{min} is 22.1723m which is greater than the specified safety radius of the obstacle. The simulation response for level flight at worst case parameters is shown in Fig.13. This concludes that the obstacle avoidance algorithm and the controller provide adequate performance at the worst-case parameters. Furthermore, in the presence of all the described variations, the safety margin for anti-collision is respected.

TABLE.4. DIRECT RESULTS FOR A UAV OBSTACLE AVOIDANCE SYSTEM

Algorithm	Iterations	m (kg)	J (kgm^2)	V_0 (m/s)	d_{min} (m)	Fun.Evalu taken	Time (hours)
DIRECT	200	2.5778	0.0239	19.8148	22.1723	2283	2

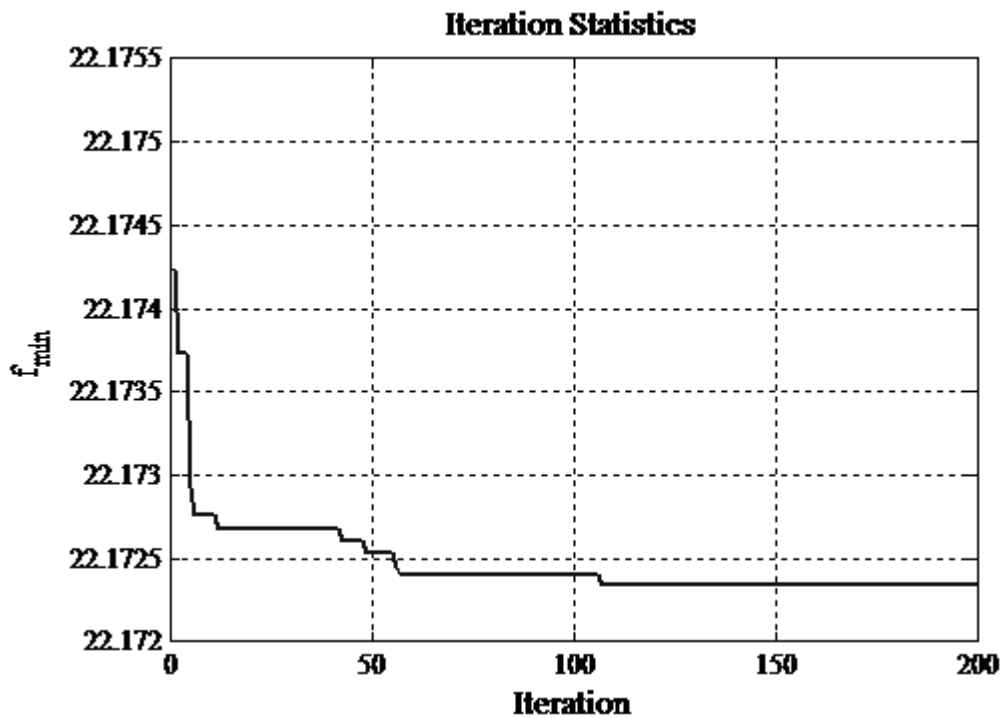


Fig.10. *DIRECT* algorithm- Iteration vs. Fitness value

Obstacle avoidance at worst case parameters

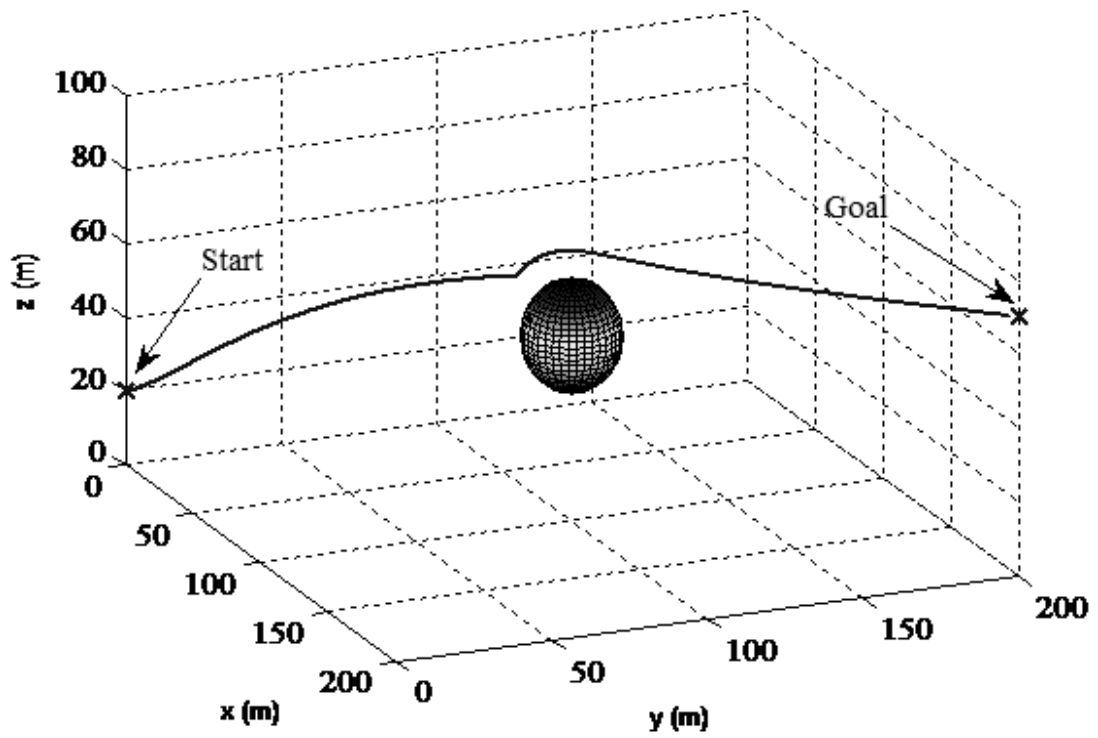


Fig.11. Simulation response at worst-case parameters

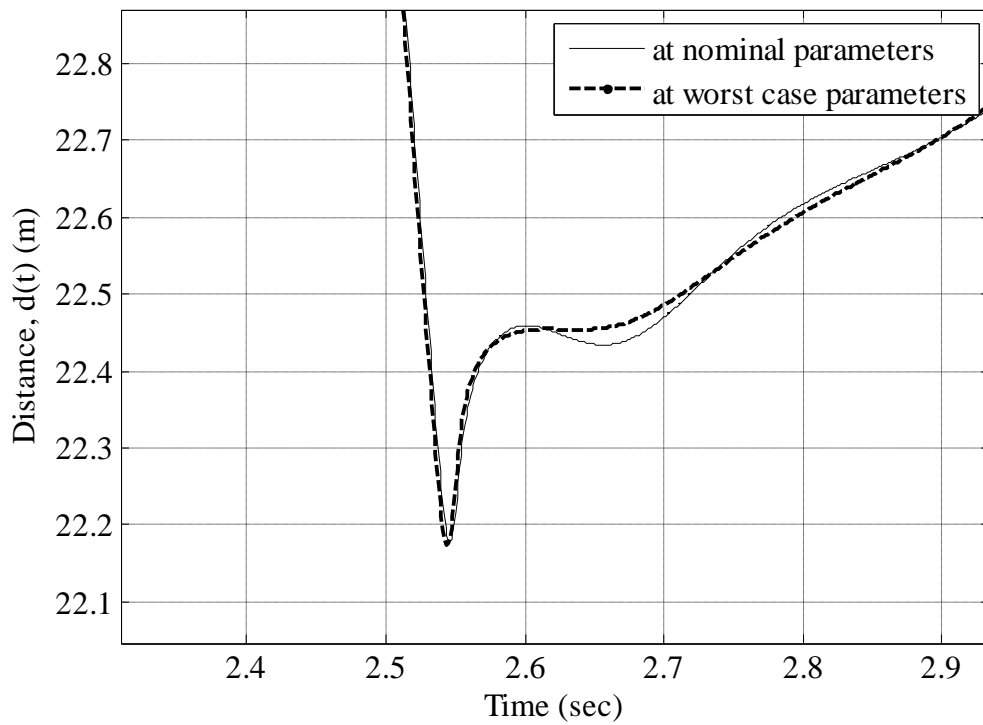


Fig.12. Time vs distance to the obstacle at nominal and worst case parameters

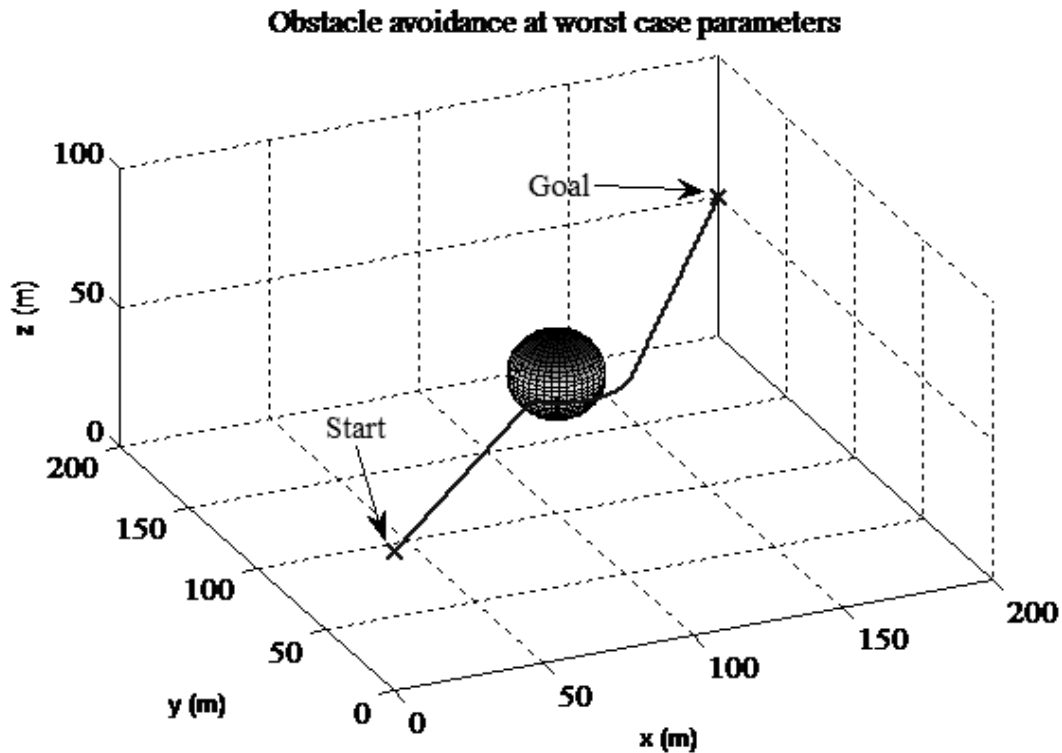


Fig.13. Simulation response at worst-case parameters

3. Conclusions

In this paper, optimization based clearance process of obstacle avoidance systems is applied to a simplified UAV 4DOF model. The spherical obstacle is considered, and an obstacle avoidance algorithm is developed and verified. The potential field based collision avoidance algorithm is selected as the candidate technique for verification although the proposed approach shall be equally applicable to other collision avoidance techniques after proper modification. The key idea in this verification approach is that in optimization, it is not necessary to evaluate a cost function over all possible solutions to find the optimal solution. However different from many optimization problems, it is important to find all the possible worst cases in the worst case analysis of safety critical functionality like obstacle avoidance. This requires an optimization algorithm that may guarantee the convergence of the global optimal solution.

In developing optimization based worst case analysis for verification of collision avoidance algorithms, the minimum distance to the obstacle during collision avoidance manoeuvre is defined as the cost function and then the task is to automatic search the worst cases without the need to exhaustively evaluate all possible combinations of variations. Mass, inertia and initial velocity variations are considered in this case study. The local optimization method does not give a unique solution as different worst cases are identified when the optimization starts from different initial conditions. Therefore, the local optimization is not suitable for verification of collision avoidance algorithms for this case study. It is a non convex nonlinear

optimization problem so it is possible to miss the most critical cases. To overcome this problem, global optimization algorithms are studied.

Stochastic global optimization algorithms including GA and GLOBAL methods have been applied to the problem. GA algorithm performs better than the GLOBAL for this UAV problem. However, as they are stochastic global optimization algorithms, they cannot guarantee the optimization process converge to the global solutions, i.e. the worst-cases. To overcome this drawback, the deterministic global optimization of the DIRECT method has been investigated for the worst-case analysis. Compared with other global optimization algorithms in this study, the DIRECT algorithm can guarantee finding the worst case, however, it takes more time to converge. Further work will be on applying the proposed verification approach to a more complicated simulation environment for Unmanned Aerial Vehicles (UAVs).

References

- [1] C. Fielding, A. Varga, S. Bennai, M. Selier(Eds). Advanced techniques for clearance of flight control laws. Springer LNCIS.283, pp. 3-33, 107-117, Heidelberg 2002.
- [2] B. Cheol Min, H. Y. Kwon, D. Kim. Path planning algorithm for VTOL type UAVs based on the methods of ray tracing and limit cycle. Computational Intelligence in Robotics and Automation (CIRA), IEEE International Symposium, pp. 296-301, 15-18 Dec 2009.
- [3] S. Griffiths, J. Saunders, A. Curtis, T. McLain. Obstacle and terrain avoidance for miniature aerial vehicles. IEEE Robotics and Automation Magazine, Vol-13, No-3, pp. 34-43, September 2006.
- [4] P. O. Pettersson, P. Doherty. Probabilistic roadmap based path planning for an autonomous unmanned helicopter. Journal of Intelligent and Fuzzy systems, Vol-17, No-4, pp. 395-405, September, 2006.
- [5] S. A. Bortoff. Path planning for UAVs. Proceedings of the American Control Conference, Vol-1, pp. 364-368, June 2000.
- [6] T. Lapp, L. Singh. Model predictive control based trajectory optimization for nap-of-the-earth (NOE) flight including obstacle avoidance. Proceeding of the 2004 American Control Conference, Boston, Massachusetts , June 30-July 2 2004.
- [7] S. Kanchanavally, R. Ordonez, C. J. Schumacher. Path planning in three dimensional environment using feedback linearization. American Control Conference, Minneapolis, MN, pp. 6, 14-16 June 2006.
- [8] T. Paul, T. R. Krogstad, J. T. Gravdahl. UAV formation flight using 3D potential field. Control and Automation 2008, 16th Mediterranean Conference on. pp. 1240-1245. 25-27 June 2008.

- [9] K.K. James, Lexington, Safety analysis methodology for unmanned aerial vehicle (UAV) collision avoidance systems, United States Air Force, #F19628-00-C-0002.
- [10] N. Snooke, C. Price, C. Downes, C. Aspey. Automated failure effect analysis for PHM of UAV. Proceedings of the International System Safety and Reliability Conference (ISSRC 2008), ISBN:978-981-08-0446-6, April 2008.
- [11] E. W. Roland, R. John Hansman. Safety considerations for operation of unmanned aerial vehicles in the national airspace system. MIT International Center for Air Transportation, Massachusetts Institute of Technology, Cambridge, USA, March 2005.
- [12] R. Lutz, S. Nelson, A. Patterson-Hine, C. R. Frost, D. Tal. Identifying contingency requirements using obstacle analysis. Proceedings 13th IEEE International Conference on Requirements Engineering, pp- 263-272, 29 Aug- 2 Sept 2005.
- [13] S. Temizer, M. J. Kochenderfer, L. P. Kaelbling, T. L. Perez, J. K. Kuchar, Collision Avoidance for Unmanned Aircraft using Markov Decision Processes, AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, Aug 2-5, 2010.
- [14] T. Schouwenaars, J. How, Eric Feron, Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees, AIAA Guidance, Navigation, and Control Conference and Exhibit, 16-19 August 2004, Providence, Rhode Island.
- [15] S. Srikanthakumar, W. H. Chen. Optimisation-based clearance process for obstacle avoidance systems. International Conference on Automation and Computing, Birmingham, 11th September 2010.
- [16] R. Carona, A. P. Aguiar, J. Gaspar. Control of unicycle type robots tracking, path following and point stabilization. International Proceeding of IV Electronics and Telecommunications, pp. 180-185, Lisbon, Portugal, November 2008.
- [17] A. D. Luca, G. Oriolo. Local Incremental Planning for Nonholonomic Mobile Robots. Proceeding.of.IEEE International Conference on Robotics and Automation, San Diego, CA, pp. 104-110, 8-13 May 1994.
- [18] A. Bemporad, A. D. Luca, G. Oriolo. Local Incremental Planning for a car-like robot navigating among obstacles. Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, pp. 1205-1211, 22-28 April 1996.
- [19] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, International Journal Robotics Research, Vol.5, N0-1, pp,90-99, 1986.
- [20] Mario Isabel Ribeiro, Obstacle avoidance, Instituto de Sistemas e Robotica, Nov, 2005.

- [21] Eliana Costa e Silva, E. Bicho, W. Erhagen, The potential field method and the nonlinear attractor dynamics approach: what are the difference?, 7th Portuguese Conference on Automatic Control”, CONTROLO 2006.
- [22] Mathworks.co.uk
- [23] J. Holland, Adaption in natural and artificial Systems, University of Michigan press, Ann Arbor, MI 1975.
- [24] D. Bates, M. Hagstrom, Nonlinear Analysis and Synthesis Techniques for Aircraft Control, Springer, 2007.
- [25] J. S. Arora, Introduction to optimum design, second edition, Elsevier, Academic Press.
- [26] W. M. Rand, Controlled observations of the Genetic algorithm in a changing environment: case studies using the Shaky ladder hyperplane-defined functions, Thesis, The University of Michigan, 2005.
- [27] Tiber Csendes et al, The GLOBAL Optimisation Method Revisited, University of Szeged, Hungary, September, 2008.
- [28] D. Jones, DIRECT Global optimization algorithm, Encyclopedi of Optimization, Kluwer Academic Publishers, 2001.
- [29] D. E. Finkel, C. T. Kelley, Convergence Analysis of the Direct Algorithm, North Carolina State University, July, 2004.
- [30] D. E. Finkel, DIRECT optimization algorithm user guide, Center for Research in Scientific Computation, North Carolina State University, March 2, 2003.