Worst-case Analysis of Moving Obstacle Avoidance Systems for Unmanned Vehicles

Sivaranjini Srikanthakumar, Wen-Hua Chen
Department of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU, UK
sivaranjinisk@yahoo.co.uk

Abstract: This paper investigates worst-case analysis of a moving obstacle avoidance algorithm for unmanned vehicles in a dynamic environment in the presence of uncertainties and variations. Automatic worst-case search algorithms are developed based on optimization techniques, illustrated by a Pioneer robot with a moving obstacle avoidance algorithm developed using the potential field method. The uncertainties in physical parameters, sensor measurements and even the model structure of the robot are taken into account in the worst-case analysis. The minimum distance to a moving obstacle is considered as the objective function in the automatic search process. It is demonstrated that a local nonlinear optimization method may be not adequate, and global optimization techniques are necessary to provide reliable worst-case analysis. Monte Carlo simulation is carried out to demonstrate that the proposed automatic search methods provide a significant advantage over random sampling approaches.

***Keywords*— Collision avoidance, Optimization, Potential field method, Monte Carlo simulation, Robustness analysis.**

1. Introduction

Safety of motion planning is an important issue in mobile robotics applications. Path planning and navigation schemes aim at guiding unmanned vehicles reaching a goal safely while avoiding collision in a known/unknown environments. In dynamic environments, motions of moving obstacles are not known beforehand. The planner has to find immediate future trajectories using current sensor information of the identified moving obstacles. To this end, many collision avoidance algorithms have been proposed and tried on various applications. However, it is still far away to prove that those algorithms are reliable and always provide adequate performance under all the possible events in real operation. In addition to offering better performance, a key practical concern related to any new method 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 unmanned vehicles. The objective of this paper is to develop advanced algorithms to support the deployment of safety-critical moving obstacle avoidance systems (OAS) for unmanned vehicles.

Before the first vehicle maneuver can be executed, the clearance of control laws and collision avoidance algorithm must be performed to prove that the controlled vehicle meets all the clearance criteria. For this purpose, extensive computer simulations and robustness assessment are performed. This verification approach provides much useful information, for example worst-case parameters combinations, which can serve to increase the performance of the vehicle or to redesign the control laws and collision avoidance algorithms. Therefore, verification of OAS would potentially contribute to reduce the global costs of collision avoidance algorithm testing and controller tuning assessment. To a large extent, the difference between the simulation based worst-case analysis (or Monte Carlo simulations) and real applications (experimental tests) depends on the fidelity of the model used in the verification process.

Depending on the operation scenarios, several collision avoidance algorithms have been developed for mobile robot path planning in the presence of unknown obstacles. Ge and Cui proposed a potential field method for motion planning of mobile robots in a dynamic environment where both the target and the obstacles are moving [1]. Raja et al. [2] introduced the Waiting Time Concept algorithm to resolve the problem of motion planning for a robot. A Conflict Detection and Resolution method was described by using geometric approach for unmanned aerial vehicles in a dynamic environment [3], while the survey in [4] reveals that the potential field method has been applied to various robot motion planning in the last three decades. Therefore, in this paper, the artificial potential field method is chosen as a candidate technique of path planning and moving obstacle avoidance for the verification study as it is simple and widely used. The verification technique proposed in this paper may be applicable for other moving obstacle avoidance algorithms after appropriate modifications.

Fault Tree Analysis was also applied to the TCAS (Traffic Alert and Collision Avoidance Systems) for the safety analysis in [5]. The reachable sets were computed in [6] to verify the safety of autonomous cars in a dynamic environment. This method has been developed for hybrid systems. Fraichard proposed three safety criteria for the safety analysis of mobile robotics systems, and a number of existing collision avoidance schemes are evaluated with respect to these three safety criteria [7]. It has been established that, in all cases, Nearness Diagram, Dynamic Window, and Velocity Obstacle violated one or several of the safety criteria. Motion safety of these approaches, especially in the presence of moving objects, could not be guaranteed. The safety analysis also shows that only the Inevitable Collision States method satisfies those three safety criteria.

1.1 Sources of Uncertainties

In the development of collision avoidance algorithms, only a simple kinematic model of the vehicle is normally used. This greatly simplifies the analysis and design of the collision avoidance algorithms. However, the model in the verification stage must be as close to the real world as possible, which demands a much more complicated model.

Particularly three types of uncertainties are considered in this study: structural uncertainty, parameter uncertainty, and also data uncertainty in obstacle detection sensors. In general, a simplified model of a vehicle and its operational environment is used in the algorithm development process. However, the real vehicle and its operational environment are much more complicated, with possibly a much high order of dynamics, nonlinearity and much more complicated operation scenarios. This causes structural uncertainties in the verification of collision avoidance algorithms.

The parameter uncertainties represent the variations of parameters that capture the changes of the vehicle dynamics and its operational environment. 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 operation environment (e.g. tyre friction for different road surfaces).

In the online motion planning, unmanned vehicles must be able to sense obstacles, determine the obstacles positions and velocities, and reach the target position. However, there are inevitably uncertainty in the sensor data due to the limited accuracy of the robot's sensors and environmental noises. Therefore, it is necessary to verify whether or not an obstacle avoidance system under question is able to avoid moving obstacles with uncertain sensor data. To this end, sensor data uncertainty in an unknown environment is also considered in this study. First it is assumed that there is no error in sensor data at the nominal parameters, i.e., the sensors work as 100% correctly at the nominal case. Then, uncertainties are introduced in sensor data by adding the nominal value with errors within prescribed upper and lower bounds.

Optimization-based verification algorithm is applied to the moving OAS in the presence of all these uncertainties. The offline verification process is to prove that the vehicle is safe under all the conditions and variations. This is particularly important for safety critical functions such as collision avoidance.

1.2 Anti-collision condition for moving obstacle avoidance

The motion planning of a mobile robot in a dynamic environment is to plan and control the robot motion from the starting position to the goal position while avoiding moving obstacles. A dynamic obstacle avoidance algorithm in 2D is investigated in this study where a potential field based dynamic obstacle avoidance algorithm for non-cooperative robots is selected.

As shown in Fig.1, ρ_0 is a positive constant describing the potential field influence range of the obstacle. In general, one robot is considered as an ‘intruder’ (robot-B) whereas the other one (robot-A) is assumed to be equipped with an OAS which is capable of detecting and avoiding the intruder without knowing its intention.

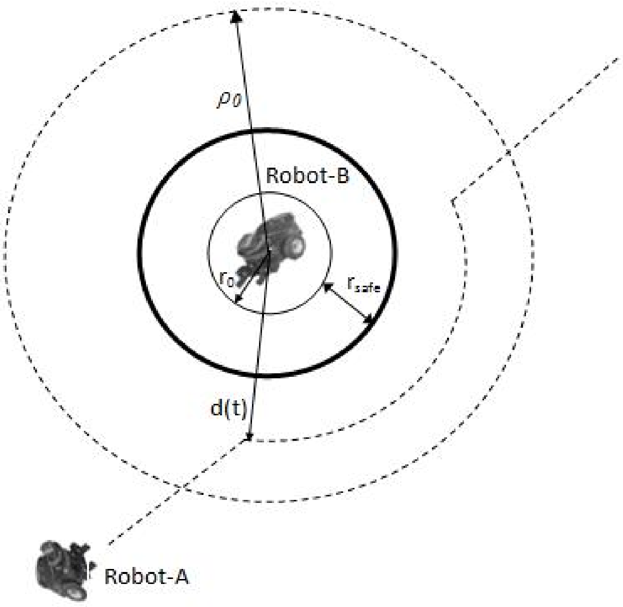

Fig. 1 Moving obstacle avoidance clearance criterion

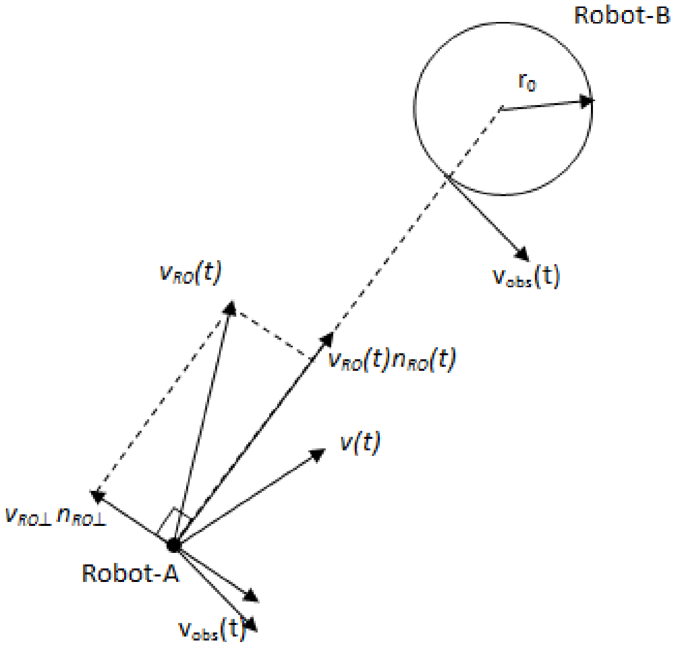

Fig.2 Relative velocity between the robot and the obstacle

In Fig.2, the relative velocity between the robot and the obstacles in the direction from the robot to the obstacle is defined as:

$$v_{RO}(t) = [v(t) - v_{obs}(t)]^T n_{RO} \tag{1}$$

where n_{RO} is a unit vector pointing from the robot to the obstacle; $v(t)$ and $v_{obs}(t)$ are the robot and obstacle velocities, respectively. If $v_{RO}(t) \leq 0$, then the robot is moving away from the obstacle. Therefore, no avoidance maneuver is needed. If $v_{RO}(t) > 0$, the robot is moving close to the obstacle and avoidance maneuver must be activated when the distance between the vehicle and the obstacle is predicted to be below a certain threshold.

The minimum distance from the vehicle to an obstacle (d_{min}) during a collision avoidance maneuver is chosen as the criterion for the performance assessment. Having an acceptable safe margin during all the operation conditions is a widely used criterion to assess the safety of a moving vehicle. This essentially creates a safety bubble around a moving vehicle [8]. So it is nature and intuitive to select the minimum distance between the vehicle and any moving obstacle during all maneuvers including collision avoidance for the performance assessment.

The robot-A can detect the moving obstacles shape, positions, orientation and velocity where a moving obstacle is considered as a circular object. For a moving OAS safety analysis, an intruder is defined with a radius of r_0 and a safety margin of r_{safe} (See. Fig.1). The intruder radius and safety margin can be chosen according to the dimensions of the robots. Pioneer 3-DX robots are considered in this case study [9]. Letting $r = r_0 + r_{safe}$, the anti-collision condition is defined as $d_{min}>r$.

In the moving OAS process, all violations of the minimum distance to the obstacle must be found and the corresponding worst-case combination of the uncertain parameters must also be computed. It shall be mentioned that this paper does not aim to develop a worst-case analysis approach providing explicit conditions which influence the minimum distance between the vehicle and the obstacle. Instead, it tries to answer the following question: given a designed collision avoidance algorithm, whether or not it fulfills the anti-collision conditions under all the possible described variations so is safe for operation.

1.3 The contributions of this paper

The worst-case analysis approach advocated in this paper is to combine optimization techniques with simulation. To make the worst-case analysis results reliable, the analysis shall be performed based on a much more detailed model which not only captures factors ignored in the design stage but also represents realistic operational environments. Due to the complexity of the model, it is unlikely to find analytic expressions. Therefore, simulation is essential for providing numerical solutions. This work is an extension of the work presented in [10], where the optimization based verification process has been proposed for a small scale unmanned aircraft. In addition to looking into a different application, there are two main extensions in this paper: first the moving object avoidance is considered in this paper which makes the prediction of the possible worst-cases more complicated and challenging; secondly the uncertainty in sensor measurements are considered in this paper.

The proposed approach in this paper is demonstrated by a unicycle-like mobile robot with collision avoidance algorithms developed using the potential field method. To this end, a complete unicycle mobile robot model is employed in the worst-case analysis, consisting of the kinematic and dynamic model, the speed control loop, external forces and wheel sliding. All possible variations in the vehicle and its operational conditions are then taken into account in the study. The worst-case analysis for collision avoidance algorithms is treated as a constrained nonlinear optimization problem with simulation being involved in each iteration. The worst-case analysis in the presence of all the possible uncertainties is cast as a problem of finding the combinations of the variations where the minimum of the minimum distance to the obstacle *(d_{min})* appears. Monte Carlo simulation is carried out to provide a benchmark comparison of the proposed automatic worst case search methods.

The rest of the paper is organized as follows. As a benchmark, the kinematic and dynamic model of a commercial unicycle mobile robot is introduced in Section 2. A moving OAS is developed using the potential field method. Simulation results at the nominal parameters are provided to show the collision avoidance system functioning appropriately. Initial robustness analysis of the collision avoidance algorithm is carried out in Section 3. Optimization-based verification is introduced in Section 4, where local and global optimization algorithms are studied. Stochastic global optimization algorithms including GA and GLOBAL algorithms are considered to find the global minimum of the minimum distance to the obstacle. However, these methods cannot guarantee the global minima. Therefore, the deterministic global algorithm of DIRECT (Dividing RECTangles) method is further studied in the context of the verification of the moving OAS. Monte Carlo simulation is carried out for the purpose of comparison. Finally, Section 5 concludes the paper.

2. Moving Obstacle Avoidance System Setup

A motion planning and collision avoidance algorithm is designed in this section using the potential filed method. Although this is not the contribution of this paper, it is necessary for demonstrating the mismatching between the model used in the algorithm development and the vehicle itself, and for presenting the proposed verification process.

2.1 Unicycle Mobile Robot Model

Pioneer 3-DX (See Fig.3) is an intelligent mobile robot. It can carry loads more robustly. P3-DX has been used in many applications including automating highway maintenance and constructions. The robot mass is 9kg with the payload of 25kg. A schematic figure of a unicycle-like mobile robot is shown in Fig.4 [9].

Fig.3 Pioneer 3-DXmobile robots [9] or cite where these pictures were took from as they not produced by you

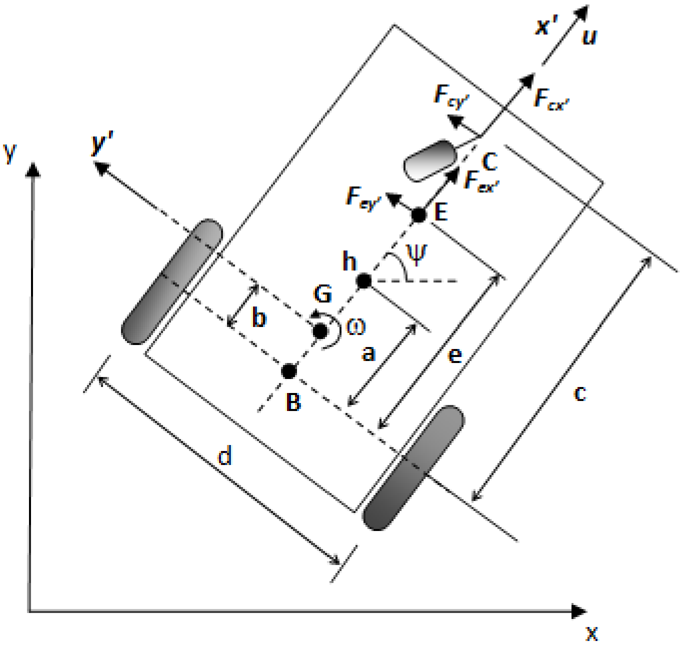

Fig. 4 Parameters of the unicycle-like mobile robot [9]

where G is the centre of mass; $h=[x\ y]^T$ the point that is required to tracks a trajectory; u the longitudinal velocity of the centre of mass; ω and ψ the angular velocity and heading of the robot, respectively; D, b, a, e and c various distances as defined in the figure; C the position of the caster wheel; $F_{cx'}$ and $F_{cy'}$ the longitudinal and lateral force exerted on C by the caster wheel; E the location of a tool on-board the robot; $F_{ex'}$ and $F_{ey'}$ the longitudinal and lateral force exerted on E by the tool, respectively.

In the robotic industry, most robots have low level PID velocity controllers to track input reference velocities and the motor voltage (V_u ,V_ω) is not driven directly. Therefore, linear and angular reference velocities are considered as control signals [9]. In order to express these control signals, the robot servos have PD controllers to control the velocities of each moto. The corresponding proportional gains k_{PT} and k_{PR} , and derivative gains k_{DT} and k_{DR} are described in Eq.2. These PD controllers are included in the model structure which is shown in Fig.5.

$$\begin{bmatrix} V_u \\ V_\omega \end{bmatrix} = \begin{bmatrix} k_{PT}\left(u_{ref} - u\right) - k_{DT}\dot{u} \\ k_{PR}\left(\omega_{ref} - \omega\right) - k_{DR}\dot{\omega} \end{bmatrix} \tag{2}$$

The complete mathematical model **[9]** is written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} u\, cos\, \psi - a\omega\, sin\, \psi \\ u\, sin\, \psi + a\omega\, cos\, \psi \\ \omega \\ \frac{\theta_3}{\theta_1}\omega^2 - \frac{\theta_4}{\theta_1}u \\ -\frac{\theta_5}{\theta_2}u\omega - \frac{\theta_6}{\theta_2}\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} \tag{3}$$

where u and ω are the current robot linear and angular velocities. u_{ref} and ω_{ref} are the linear and angular reference velocities. $\theta = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6]^T$ is the vector of model parameters which are given below:

$$\theta_1 = \left[\frac{R_a}{k_a}(mR_t r + 2I_e) + 2rk_{DT}\right] / \left(2rk_{PT}\right)$$

$$\theta_2 = \left[\frac{R_a}{k_a}(I_e d^2 + 2R_t r(I_z + mb^2)) + 2rdk_{DR}\right] / \left(2rdk_{PR}\right)$$

$$\theta_3 = \frac{R_a}{k_a}mbR_t / \left(2k_{PT}\right)$$

$$\theta_4 = \frac{R_a}{k_a}(\frac{k_a k_b}{R_a} + B_e) / \left(rk_{PT}\right) + 1$$

$$\theta_5 = \frac{R_a}{k_a}mbR_t / \left(dk_{PR}\right)$$

$$\theta_6 = \frac{R_a}{k_a}(\frac{k_a k_b}{R_a} + B_e)d / \left(2rk_{PR}\right) + 1 \tag{4}$$

$\delta_{un} = [\delta_x \;\; \delta_y \;\; 0 \;\; \delta_u \;\; \delta_\omega]^T$ is the uncertainty vector associated to the mobile robot:

$$\delta_x = -\bar{u}^s \sin\psi$$

$$\delta_y = \bar{u}^s \cos\psi$$

$$\delta_u = \frac{\frac{m\omega\bar{u}^s + F_{ex'} + F_{cx'}}{I_e} + \left(\frac{k_a k_b}{R_a} + B_e\right)\frac{u_r^s + u_l^s}{R_t r I_e} + \frac{\dot{u}_r^s + \dot{u}_l^s}{R_t r}}{\frac{m}{I_e} + \frac{2}{R_t r} + \frac{2k_{DT}k_a}{R_a R_t I_e}}$$

$$\delta_\omega = \frac{\left(\frac{k_a k_b}{R_a} + B_e\right)\frac{u_r^s - u_l^s}{2R_t r I_e} + \frac{\dot{u}_r^s - \dot{u}_l^s}{2R_t r} - \frac{mb\dot{\bar{u}}^s}{I_e d} + \frac{eF_{ey'} + cF_{cy'} + \tau_e}{I_e d}}{\frac{d}{2R_t r} + \frac{I_z + mb^2}{I_e d} + \frac{k_{DR}k_a}{R_t R_a I_e}} \tag{5}$$

where m is the robot mass; I_z is the robot moment of inertia about vertical axis located in G; r is the right and left wheel radius; u_r^s and u_l^s are the longitudinal slip speeds of the right and left wheel; $\bar{u}^s$ is the lateral slip speed of the wheels; k_a is the torque constant multiplied by the gear ratio; k_b is the voltage constant multiplied by the gear ratio; k_t is the nominal radius of the tire; R_a is the electric resistance constant; R_t is the radius of the tire; τ_e is the moment exerted on E by the tool; I_e and B_e are the moment of inertia and the viscous friction coefficient of the combined motor rotor, gearbox, and wheel.

2.2 Motion Control and Obstacle Avoidance

The control system involves two control loops (inner and outer) as shown in Fig.5. The outer-loop is the motion controller which generates the desired linear velocity u_d and angular velocity ω_d. The inner-loop is chosen as a Proportional-Integral (PI) controller because the robot servos already have built-in PD controllers to control the velocities of each motor. The inner-loop PI control law is responsible to compute the linear and angular reference velocities signals (u_{ref} and ω_{ref}). True and desired velocities are saturated without exceeding given limits.

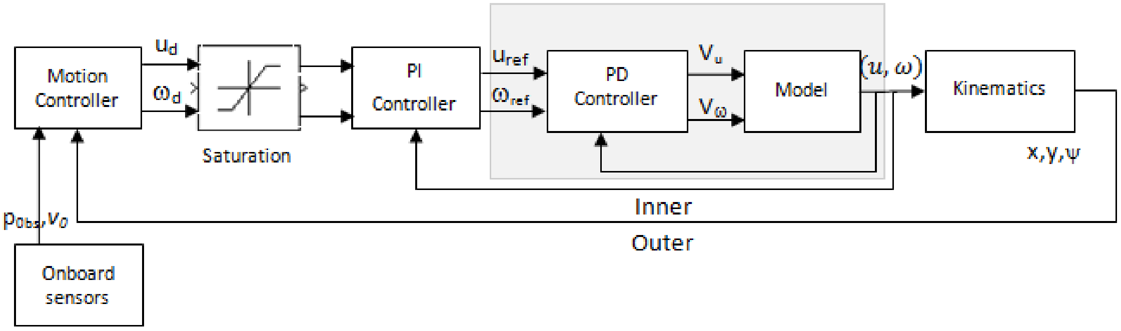

Fig. 5 Mobile robot motion planning control systems

A. *Inner-Loop Controller*

A PI control law with anti-windup is proposed as speed controllers which are given below. The goal of the inner loop is to achieve and maintain the desired linear velocity u_d and desired angular velocity ω_d.

$$\begin{aligned} u_{ref} &= K_1 e_u + K_3 \int_o^t e_u\,(\tau)d\tau \\ \omega_{ref} &= K_2 e_\omega + K_4 \int_o^t e_\omega\,(\tau)d\tau \end{aligned} \tag{6}$$

where K_1 and K_2 are proportional controller gains, and K_3 and K_4 are integral controller gains. $e_u = u_d - u$ and $e_\omega = \omega_d - \omega$ are the linear and angular velocity errors, respectively.

B. *Outer-Loop Motion Controller*

The kinematic model of the vehicle is considered for the collision avoidance algorithm development stage. In general, the kinematic model of the wheeled mobile robot is described as

$$\dot{X} = G(X)U \tag{7}$$

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 [11]. Given any desired smooth trajectory (start, goal and obstacle positions) X_d, a straight-forward approach is to design the input command *U* using the pseudo-inverse control law

$$U = G^{\#}(X)\dot{X}_d \tag{8}$$

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

For the unicycle robot, $X = (x, y, \psi)$ is the configuration vector. Comparing Eq.3 and Eq.8 gives [11]

$$G(X) = \begin{bmatrix} cos\psi & -asin\psi \\ sin\psi & acos\psi \\ 0 & 1 \end{bmatrix} \tag{9}$$

It follows from Eq.8 that the pseudo-inverse of *G(X)* takes the form

$$G^{\#}(X) = \frac{1}{(a^2+1)}\begin{bmatrix} (a^2+1)cos\psi & (a^2+1)sin\psi & 0 \\ -asin\psi & acos\psi & 1 \end{bmatrix} \tag{10}$$

With $U=[u_d\ \omega_d]^T$, the feedback law Eq.8 for tracking a desired trajectory $X_d = (x_d, y_d, \psi_d)$ becomes [11]

$$u_d = k_p(\dot{x}_d cos\psi + \dot{y}_d sin\psi) \tag{11}$$

$$\omega_d = \frac{k_q}{(a^2+1)}(-a\dot{x}_d sin\psi + a\dot{y}_d cos\psi + \dot{\psi}_d) \tag{12}$$

where gains k_p and k_q are introduced to allow for additional freedom in weighting the two input commands. In order to apply the control law Eq.11 and Eq.12, the desired velocities have to be specified. These desired values can be determined using with the potential field method as described in the next section.

C. Potential Field Method

Potential field methods are widely used in path planning, but mostly in a static environment. However, the environents in real-time applications are dynamic. In [1], the potential field method for motion planning of a mobile robot in a dynamic environment was proposed. This moving obstacle avoidance algorithm is applied to the robots to verify the algorithm in the presence of uncertainties. The attractive potential field is defined as a function of the robot position to the goal position. The repulsive potential is defined as the function of the relative position and velocity of the robot with respect to the moving obstacles. The virtual forces are defined as the negative gradient of the potential field. The assumption is made as the obstacles shapes, positions and velocities can be measured on-line.

Attractive Potential Function: The attractive potential field is defined as a function of the robot position to the target position where the target is a fixed point in space. The attractive potential field is defined as follows:

$$U_{att}(h) = \alpha_p \| p_{tar} - h \|^2 \tag{13}$$

The corresponding attractive force is defined as :

$$F_{att}(h) = \ k_{att}(p_{tar} - h\ ) \tag{14}$$

where α_p and k_{att} are the positive constants; p_{tar} is the goal position; h is the robot position.

Repulsive Potential Function: A repulsive potential function is defined as the relative positions and velocities between the robot and the obstacles.

If a maximum deceleration magnitude A$_{max}$ is applied to the robot to reduce its velocity, the distance travelled by the robot before v_{RO} *defined in Eq.(1)* reduces to zero is

$$\rho_m(v_{RO}) = \frac{v_{RO}{}^2(t)}{2A_{max}} \tag{15}$$

The velocity component perpendicular to $v_{RO}(t)n_{RO}$ (See. Fig.2) is given in the following equation

$$v_{RO\perp}n_{RO\perp} = v(t) - v_{obs}\ (t) - v_{RO}(t)n_{RO} \tag{16}$$

The corresponding repulsive force (See.Fig.7) is defined as the negative gradient of the repulsive potential in terms of both the position and velocity

$$F_{rep}(h,v) = \begin{cases} 0, \quad if\ \rho_s(h,p_{obs}) - \rho_m(v_{RO}) \geq \rho_0\ or\ \ v_{RO}\ \ \leq 0 \\ F_{rep1} + F_{rep2}, \quad if\ 0 < \rho_s(h,p_{obs}) - \rho_m(v_{RO}) < \rho_0\ and\ \ v_{RO} > 0 \\ not\ defined,\ \ if\ v_{RO} > 0\ \ and\ \ \rho_s(h,p_{obs}) < \rho_m(v_{RO}) \end{cases} \tag{17}$$

where

$$F_{rep1} = -\frac{\eta}{(\rho_s(h,p_{obs}) - \rho_m(v_{RO}))^2}(1 + \frac{v_{RO}}{A_{max}})n_{RO} \tag{18}$$

and

$$F_{rep2} = \frac{\eta v_{RO} v_{RO\perp}}{\rho_s(h,p_{obs}) A_{max}(\rho_s(h,p_{obs}) - \rho_m(v_{RO}))^2} n_{RO\perp} \tag{19}$$

where ρ_0 is a positive constant describing the influence range of the obstacle; η is a positive constant; and ρ_s is the shortest distance between the robot and the body of the obstacle.

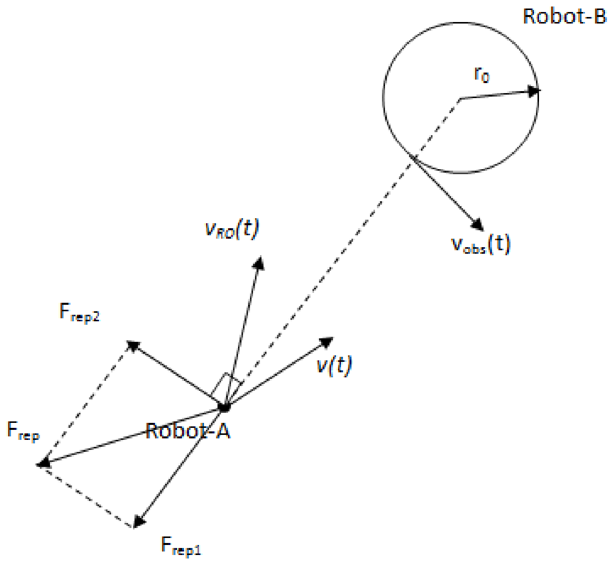

Fig.6 Repulsive forces

The total force F_{Total} is the combination of attractive force and repulsive forces. The total virtual force is used for motion planning. More details can be found in [1].

Therefore,

$$\begin{bmatrix}\dot{x}_d\\ \dot{y}_d\end{bmatrix} = F_{att} + F_{rep}(h,v)$$
$$\dot{\psi}_d = atan2\left\{\frac{\dot{y}_d}{\dot{x}_d}\right\} - \psi \tag{20}$$

By defining *atan2{0,0}=0*, the above function remains continuous along any approaching direction to the goal. The resulting command u_d and ω_d are determined by Eq. (11), (12), and (20).

2.3 Simulation Results at Nominal Parameters

Simulation is carried out to confirm that a desirable performance is achieved at the nominal case under the design described in the previous sections. The nominal parameter values of the robot are given in Table.1. The uncertainty vector δ_{un} is considered as $[-0.05\sin\psi \quad 0.05\cos\psi \quad 0 \quad 0.2 \quad 0.5]^T$. The PI controller gains and motion planner parameters for potential field force are also tuned and set to fixed values for the verification process. Proportional gains k_{PT} and k_{PR} are set to *11* and derivative gains k_{DT} and k_{DR} are set to *0.1*. The saturation limits of the true and desired values of linear and angular speeds of the mobile robot used in the simulations are *[0, 1.6](m/s)* and *[-3.5, 3.5](rad/s)* respectively. The safety radius including safe margin is chosen as *5m*. The simulation results at 10, 15, 20, and 40 seconds are shown in Fig.7 to 10. The intruder moves to the goal position without any avoiding maneuvers while robot avoids the intruder and reaches to the goal position. The minimum distance to the obstacle is obtained as 7.668m which is greater than the safety radius (d_{min}> r). Therefore, the moving obstacle avoidance algorithm functions correctly at the nominal parameters.

TABLE 1
UNICYCLE MODE, NOMINAL PARAMETERS

Symbol	Parameters	Values

m	Robot mass and payload	18 (kg)
I_z	Robot moment of inertia	20 kg.m^2
R_t	Radius of the tire	0.14 (m)
r	Right and left wheel radius	0.0977 (m)
k_a	Torque constant multiplied by the gear ratio	0.8808 (N.m/A)
R_a	Electric resistance constant	0.71 (Ω)
$\mathrm{I_e}$	Moment of inertia of the combined motor rotor, gearbox, and wheel	2 kg.m^2
B_e	Viscous friction of the combined motor rotor, gearbox, and wheel	0.8
k_b	Voltage constant multiplied by the gear ratio	0.8808 (V.s/rad)
d	Width of the robot	0.395 (m)
a	Distance to the point *h*	0.25 (m)
b	Position of center of mass	0.1 (m)

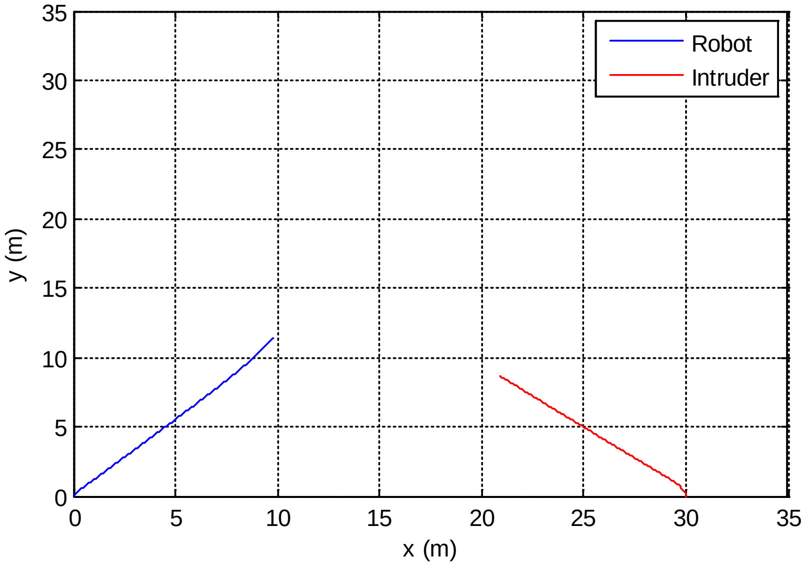

Fig. 7 Simulation response at t=10 sec

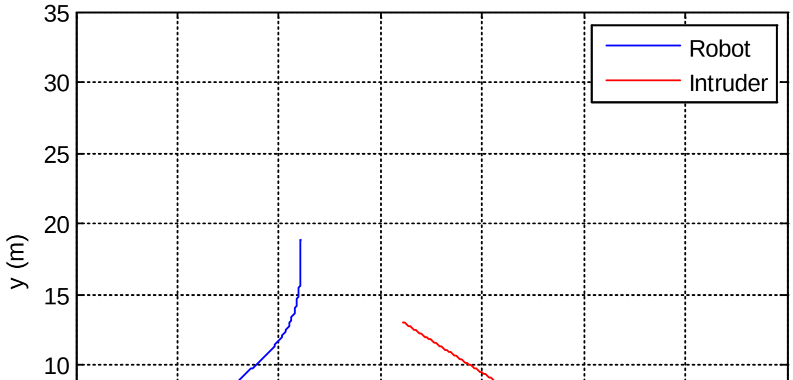

Fig.8 Simulation response at t=15 sec

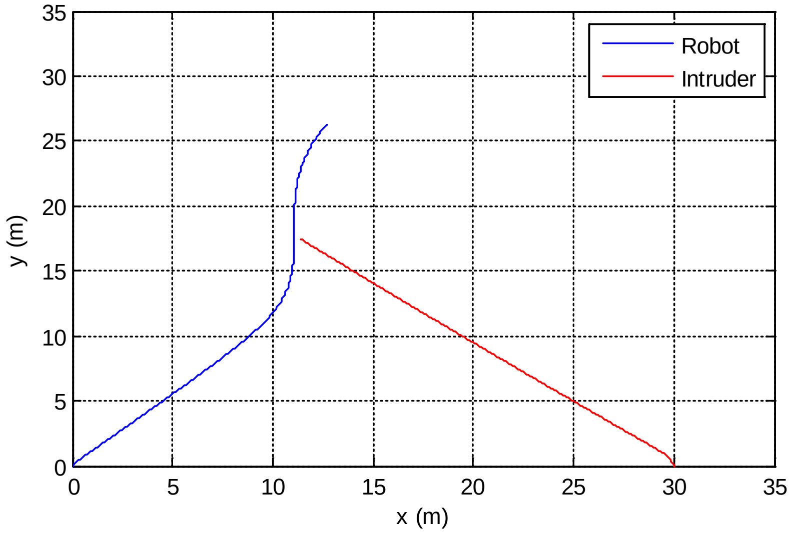

Fig.9 Simulation response at t=20 sec

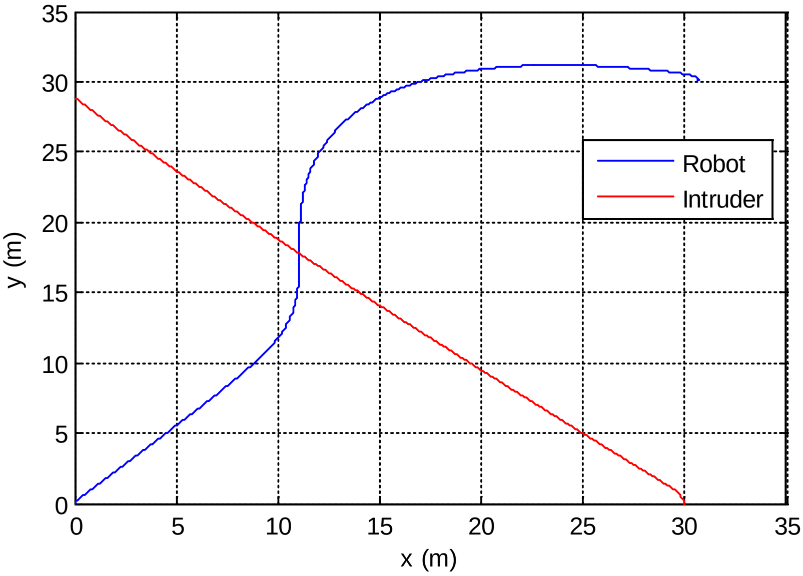

Fig.10 Simulation response at t=40 sec

3. Initial Robustness Analysis

Initial robustness analysis of the proposed algorithm is carried out in this Section. Uncertainties are introduced in sensor data as follows: x position of the obstacle: Px_0 = x_0 + Δx; y position of the obstacle: Py_0 = y_0 + Δy; Obstacle orientation : Pψ_0 = ψ_0 +$\Delta\psi$; Obstacle velocity : V_{v0}=v_{obs} + Δv, where p_{obs} =(x_0 , y_0 , ψ_0) is the true obstacle reading at the nominal case. Δx, Δy, and $\Delta\psi$ are sensor data errors in x_0, y_0 , and ψ_0 respectively. In the similar fashion, v_{obs} is the true obstacle velocity, and Δv is the velocity error. After analyzing the infulence of obstacle detection sensor data uncertainties, the most significant r uncertainties are found to be x and y position (i.e. Δx and Δy), which are chosen within the bounds to find the worst-case condition.

Eight uncertain parameters are considered in this case study. Lower and upper bounds of each uncertain parameter are given in Table.2. The structural uncertainty of δ_x, δ_y ,δ_u and $\delta\omega$ are considered in this study. Variation in lateral slip speed ($\bar{u}^s$) is applied within the range for the uncertainty of δ_x and δ_y. All the possible dynamic model parameters variations are considered, and most significant are selected for the optimization search process to find the worst-case. Fig.11 to Fig.18 show the variations of the minimum distance to the obstacle with respect to the parameter variations. It clearly shows that for different uncertain parameters, the influence on the minimum distance to the obstacle could be quite different. The minimum distance almost linearly depends on the variations of each papameter. The minimum distance to the obstacle decreases with the increase of I_e, m, δ_u, Δx.

TABLE 2

UNICYCLE MODEL, UNCERTAIN PARAMETERS

Parameter	Description	Bounds
Δx	Variation in sensor data, x (m)	[-0.5, 0.5]
Δy	Variation in sensor data, y (m)	[-0.5, 0.5]
m	Variation in robot mass and payload(kg)	[9, 34]
B_e	Variation in viscous friction of the combined motor rotor, gearbox, and wheel	[0.48, 1.12]
δ_u	Variation in uncertainty in the linear acceleration (m/s^2)	[0.1, 0.9]
δ_ω	Variation in uncertainty in the angular acceleration (rad/s^2)	[0.1, 0.9]
I_e	Variation in moment of inertia of the combined motor rotor, gearbox, and wheel (kg.m^2)	[0.2, 3.8]
$\bar{u}^s$	Variation in lateral slip speed (m/s)	[0.02,0.08]

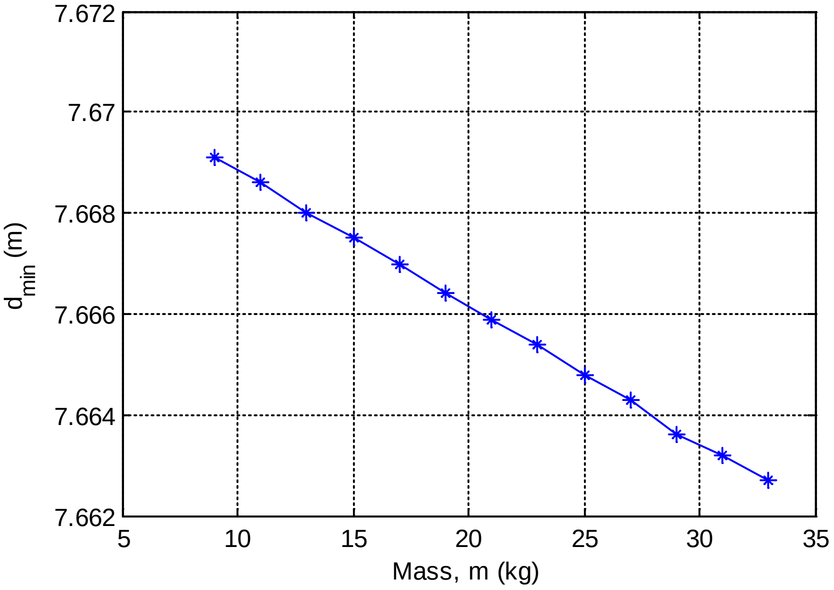

Fig. 11 Variation in robot mass and payload, m

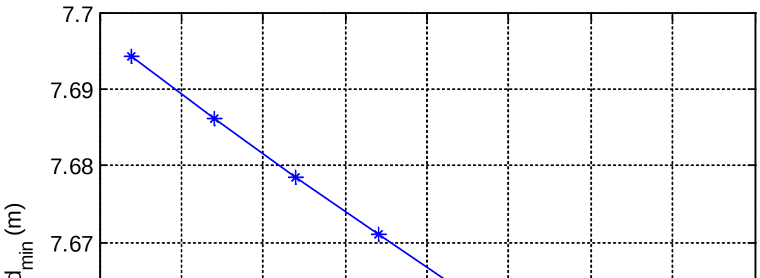

Fig. 12 Variation in the moment of inertia, I$_e$

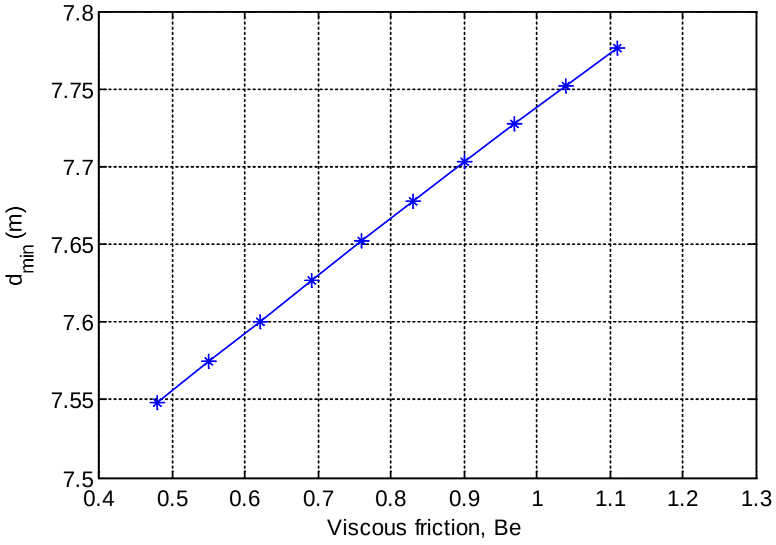

Fig.13 Variation in viscous friction, B$_e$

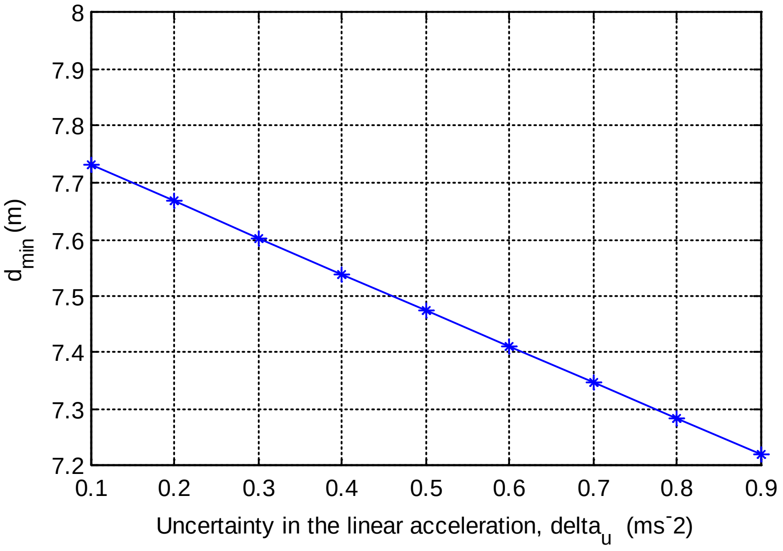

Fig.14 Variation in uncertainty in the linear acceleration, δ_u

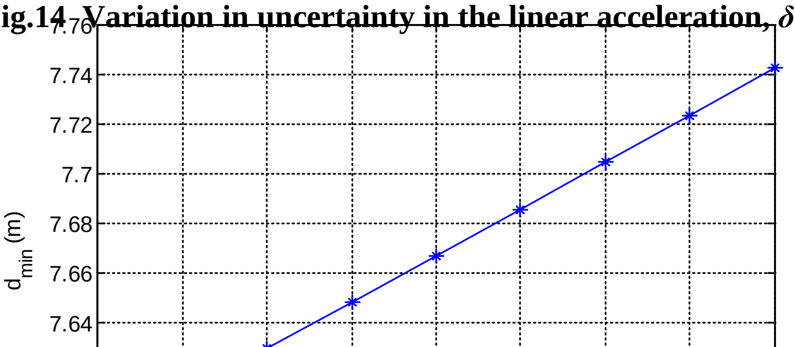

Fig.15 Variation in uncertainty in the angular acceleration, *δω*

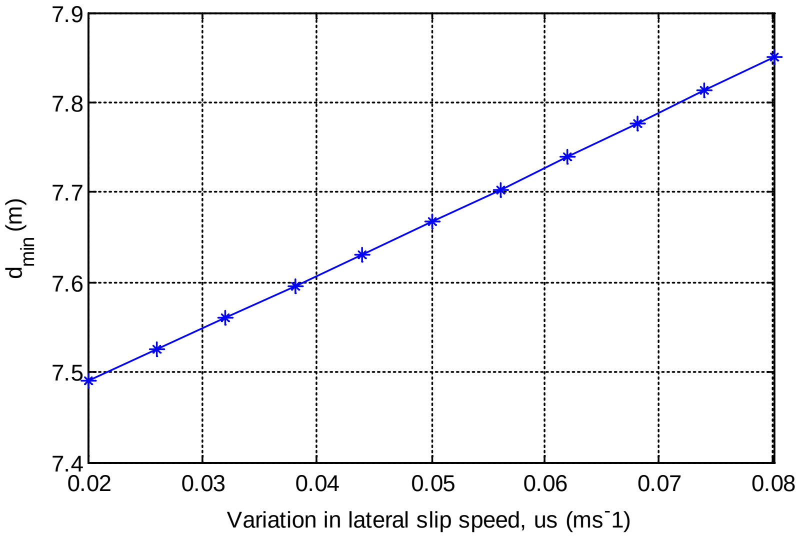

Fig.16 Variation in lateral slip speed of the wheels, u^{-s}

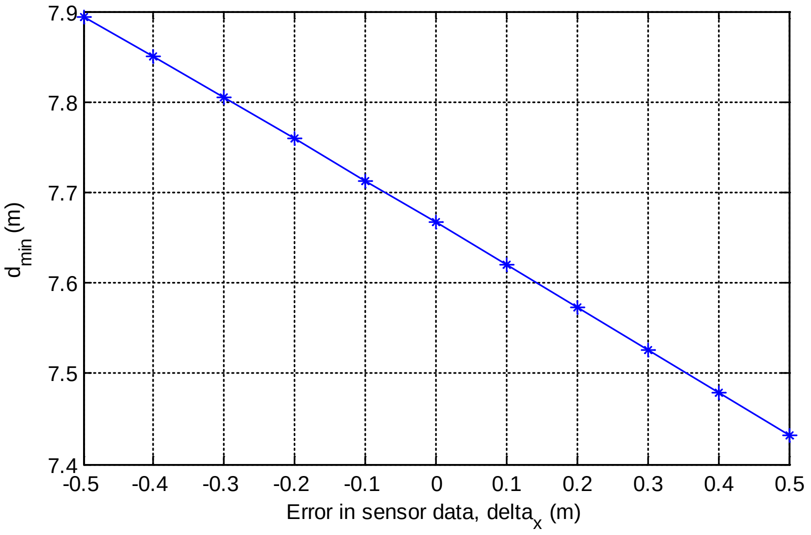

Fig.17 Variation in the sensor data, Δx

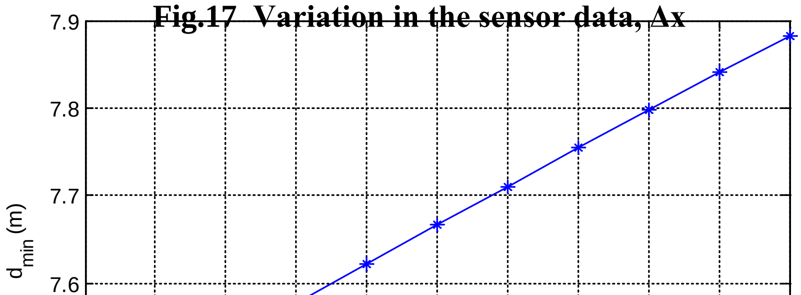

Fig.18 Variation in the sensor data, Δy

4. Optimization-based worst-case analysis Approach

The verification of collision avoidance systems can be stated as a robustness analysis problem, where a suitably defined anti-collision condition must be checked within the most significant variations of robot parameter and sensor uncertainties. In order to find the worst-case parameters and the worst-case condition, the efficient verification process developed based on optimization algorithm is applied to the moving obstacle avoidance system. This optimization-based verification method can be applied to linear and nonlinear robustness analysis, and also to different static and moving obstacle avoidance algorithms. Therefore, it is a very flexible and efficient method for the robustness analysis of collision avoidance systems.

Different algorithms for solving nonlinear optimization problems with bounds on the variables are applied to the moving obstacle avoidance system to find the worst-case. The parameters set is chosen within the bound range because they are uncertain or they may vary during operation. A non-linear optimization problem is difficult to solve because the nonlinear constraints form feasible regions that are difficult to find, and also the nonlinear objective may contain many local minima that traps the search process. Nonlinear optimization methods can be classified as local optimization and global optimization methods. Local optimization methods may fail to find the optimal solutions. Therefore, these methods may miss an unsafe point. To overcome the local minima problem, global optimization methods are applied to find the worst-case. Finding the global minimum of a nonlinear constrained optimization problem is a challenging task. A number of global optimization algorithms have been developed to find the globally optimal solutions. However, in many engineering applications, finding the global minima is a very time consuming process due to its computational complexity. The mechanism of escaping from local minima determines the efficiency of a global optimization algorithm. Global optimization methods can be classified as either stochastic or deterministic. Stochastic methods evaluate the objective function at randomly sampled points from the solution space. These stochastic global optimization methods depend on probability conditions to make decisions. Therefore, these algorithms cannot guarantee the global minima. On the other hand, the deterministic methods do not involve any elements of randomness, and these methods evaluate the objective function satisfies certain conditions, such as Lipschitz condition. Therefore, these algorithms can guarantee the optimal solution.

The objective function in the optimization is chosen as the minimum distance from the vehicle to the obstacle during the maneuver, i,e.

$$\begin{aligned} &d_{min} = min(d(t)) \qquad for\ t \leq T\ \ (sec) \\ &s.t \qquad P_L \leq P \leq P_U \end{aligned} \tag{21}$$

where T is the time period of the collision avoidance maneuver and the distance to the obstacle $d(t)$ is calculated using simulation with the completed model of the vehicle in Fig.5. The optimization problem is formulated as to finding the minimum of the objective function in Eq.(21) subject to $P_L \leq P \leq P_U$ where P is the uncertain parameters set. P_L and P_U are the lower and upper bounds of P. It shall be highlighted that in the development of the collision avoidance algorithms described in Section 2.2, the dynamics and the inner loop controller of the vehicle as shown in Fig.5 has been ignored. The other factors such as saturation are also ignored. This causes substantial differences in the complexity between the model used in the collision avoidance algorithm development and the model used in the simulation embedded in the automatic worst case searching process.

In the optimization-based moving obstacle avoidance verification process, the first step is initialization where an anti-collision condition is defined and the ranges of uncertain parameters are determined. Before applying an optimization algorithm, the anti-collision condition is checked at the nominal case. If it is satisfied, an optimization method is applied to identify the worst-case conditions and the worst-case parameters. The moving obstacle avoidance algorithm passes the verification process if the minimum distance to the obstacle at the identified worst cases satisfy the anti-collision condition. Otherwise, the obstacle avoidance algorithm and controller have to be redefined to satisfy the anti-collision condition [10]. Several optimization algorithms are investigated for the verification of the obstacle avoidance algorithms in this paper.

4.1 Local Optimization

Local optimization method is a very efficient method when the objective function is quasi-convex and the feasible region is convex. In linear optimization problems, they converge to a local minimum from some starting points. However, in nonlinear optimization problems, local search methods depend on the starting points, and converge to the local minima. Sequential Quadratic Programming (SQP) methods are a standard general purpose algorithm for solving smooth and well-scaled nonlinear optimization problems when functions and gradients can be evaluated with high precision [12]. It is an iterative method starting from an initial point and converging to a local minimum. The function *fmincon* is a MATLAB implementation. The local optimization method is applied with different starting points to the problem of evaluating a clearance criterion for the moving obstacle avoidance systems. Eight uncertain parameters are chosen for the robustness analysis. Lower and upper bounds of parameters are given to determine the worst-case parameters. The *fmincon* tries to find iteratively a minimum at an initial estimate. Therefore, different starting points are specified and compared the results. The iteration is repeated until a specified termination criterion (either maximum number of function evaluations or convergence accuracy) is met.

In Table.3, the results of the minimum distance to the obstacle and the worst case parameters with different starting points are given. At case-1, it converges to the minimum distance of 6.8164m while it is 5.8722m at case-2. More noticeably, it can be seen that there are huge differences in the converging parameters set. Therefore, the results clearly show that *fmincon* does not give the same solutions with the different starting points because a local optimization solution quality depends heavily on the initial points picked. Local optimization-based methods are not suitable for this study. Because of this worst-case violation of the optimal solution, the global optimization methods are considered to find the true worst-case.

TABLE.3. LOCAL OPTIMIZATION RESULTS

Algorithm	Starting point	Convergent point	d_{min}(m)
	$[m, B_{e,} \delta_{u,}\ \delta_{\omega,} l_{e,} \bar{u}^s, \Delta_x, \Delta_y]$	$[m, B_{e,} \delta_{u,}\ \delta_{\omega,} l_{e,} \bar{u}^s, \Delta_x, \Delta_y]$	
Fmincon-case 1	[20, 1.0, 0.2, 0.2, 0.6, 0.03, 0, 0]	[18.271, 0.48, 0.9, 0.1, 0.204, 0.02, 0.5, - 0.5]	6.8164

Fmincon-case2	[30, 1.0, 0.8, 0.8, 3.0, 0.07, 0.4, 0.4]	[34, 0.48, 0.9, 0.1 , 3.8, 0.02, 0.5, - 0.5]	5.8722

4.2 Stochastic Global Optimization

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 [13]. 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 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 the 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 [14, 15, 16].

B. GLOBAL Algorithm

GLOBAL algorithm was developed by Csendes in 1988. It is a modified version of the stochastic algorithm by Boender et al (1982) implemented in FORTRAN [17]. The new implementation GLOBAL.m has been written in MATLAB. It is a multistart clustering algorithm. It 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 starts from those points which have not been assigned to a cluster. GLOBAL uses the Single Linkage clustering rule [17]. GLOBAL.m is the bound constrained global optimization problems with a black-box type objective function. GLOBALm has different local optimization methods that are capable of handling constraints. The UNIRANDI local search method is part of the 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.

4.3 Deterministic Global Optimization

Both GA and GLOBAL algorithms are stochastic global optimization methods and cannot guarantee the worst case is found, which is vital for ensuring the safety of unmanned vehicles. Therefore, a deterministic global optimization method is investigated and applied to the moving obstacle avoidance system. DIRECT algorithm (DIviding RECTangles) is a kind of deterministic global optimization algorithms that is guaranteed to converge to the globally optimal if the objective function is continuous or at least continuous in the neighborhood of the global optimum. DIRECT algorithm was developed by Jones et al in 1993 [18]. The DIRECT algorithm was created in order to solve difficult global optimization problems with bound constrained and a real-valued objective function. DIRECT method does not require any derivative information. It is a modification of standard Lipschitzian optimization methods. The DIRECT algorithm will globally converge to the minimal value of the objective function. This global convergence may come at the expense of a large and exhaustive search over the domain. This global search algorithm can be very useful when the objective function is a "black-box" function. More details of the DIRECT algorithm can be found in [18, 19, 20]

4.4 Global Optimization Results

Stochastic algorithms including GA and GLOBAL and deterministic algorithms such as DIRECT are applied to the moving OAS to find the worst-case condition and the worst-case parameters set. The design optimization results and the performance of these optimization algorithms are compared. The iteration process is repeated until the specified stopping criteria for the optimization process is reached. And also, eight design variables are restricted within a lower and an upper bound during this process. In GA, 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 GA optimization is terminated after given iterations (100). The GLOBAL optimization with UNIRANDI as the local search method is applied to find the global solution for the moving obstacle avoidance systems. The DIRECT algorithm terminates as soon as it exceeds the given iterations of 200.

A comparison of the minimum distance to obstacle before and after the optimization is given in Table.4. A significant change in the minimum distance to obstacle is seen after the optimization. All the optimization algorithms are performed in MATLAB 2011b and Intel (R) Core(TM) 2 Duo CPU (3.16GHz). The minimum distances to the obstacle with the DIRECT , GA and GLOBAL algorithms are very closer. GLOBAL took 1112 functions evaluation with 200 sampling points while DIRECT took 8751 function evaluations. GA took 2 hours and 26 minutes to converge to the global minimum while GLOBAL and DIRECT algorithms took around 5 hours and 20 minutes, respectively. GA performs faster than other two algorithms; however, DIRECT algorithm can guarantee the global minimum.

TABLE.4. COMPARISON OF WORST-CASE CONDITION, d_{min}(m)

Before optimization d_{min} (m)	After Optimization d_{min} (m)		
Norminal Case	**DIRECT**	**GLOBAL**	**GA**
7.6668	5.8726	5.8719	5.8758

Final values of eight design variables after optimization are shown in Table.5. It can be seen that the mass is greatly increased from 18 to 34 *kg*. And also, there are huge differences in other parameters. All the three global algorithms are converged to nearly same values. The history of iteration versus fitness value for the DIRECT algorithm is shown in Fig.19. This figure shows that the fitness value of d_{min} is almost same from iteration 50 to 200.

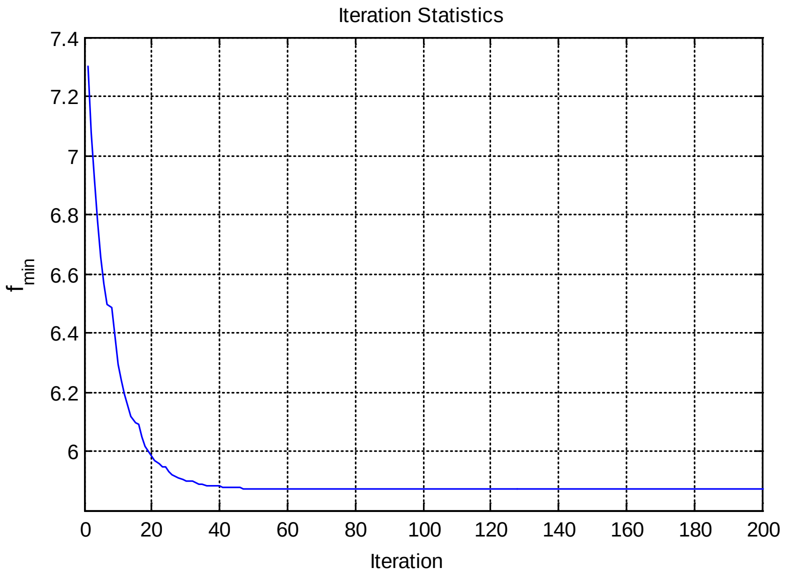

Fig.19 *DIRECT* algorithm- Iteration vs. Fitness value

TABLE.5. COMPARISON OF WORST-CASE PARAMETERS VALUES

Design Variable	Initial Value	Final Value		
		DIRECT	**GLOBAL**	**GA**
M	18	33.994	34	33.989
B_e	0.8	0.48	0.48	0.4806
δ_u	0.2	0.8999	0. 90	0.8997
δ_ω	0.5	0.1	0.1	0.1
l_e	2	3.7975	3.7998	3.7978
$\bar{u}^s$	0.05	0.02	0.02	0.02
Δ_x	0	0.4999	0. 5	0.4993
Δ_y	0	-0.5	- 0. 5	- 0.5

4.5 Validate the worst-case Results

Applying the optimization-based verification methods, the optimized minimum distance to the obstacle d_{min} is decreased from 7.6668 to 5.8727m. The performance of the moving obstacle avoidance algorithm at the worst-case parameters is checked with simulation response as shown in Fig.20. The worst-case minimum distance to the obstacle d_{min} is 5.8727*m* which is greater than the specified safety radius of the obstacle. This concludes that the moving obstacle avoidance algorithm and the controller provide adequate performance at the worst-case parameters. Furthermore, in the presence of all the described variations and uncertainties, the safety margin for anti-collision is respected. The time versus distance to the obstacle at the nominal and worst-case parameters is shown in Fig.21. It clearly shows that there is a significant difference in the minimum distance to the obstacle at the nominal and worst-case parameters during the maneuver.

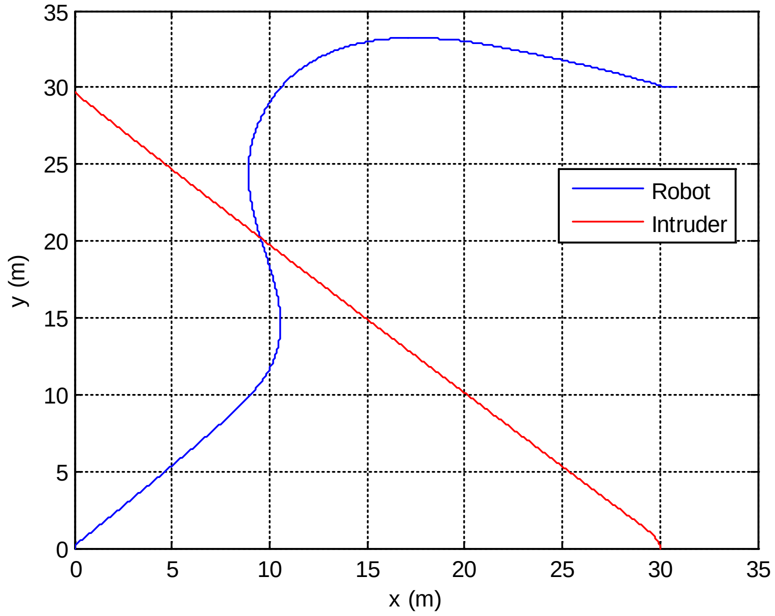

Fig.20 Simulation results at worst-case parameters, t = 40 sec

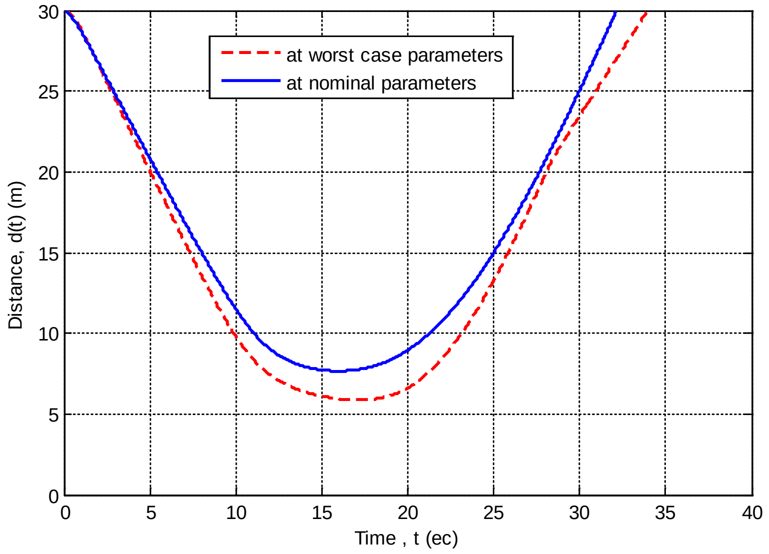

Fig.21 Time vs distance to the obstacle at nominal and worst-case parameters

4.6 Monte Carlo Simulations

To verify the proposed worst case analysis methods and benchmark their performance, the most widely used Monte Carlo method (MCM) is applied to the case study [21]. A rectangular uniform distribution is assigned to the eight uncertain parameters within their corresponding lower and upper bounds. Monte Carlo simulation is executed with 5,000 runs to find the worst case scenario and the results are shown in Fig.22. The minimum distance to the obstacle d_{min} at the worst case obtained by MCM is 6.43m while that identified by the optimization based automatic search methods proposed in this paper is 5.87m. The worst-case condition obtained from the MCM is not the true worst-case and there is a high chance of missing the true worst-case solution in this approach. Therefore, the proposed automatic worst-case analysis approach provided a more efficient and reliable verification method for the collision avoidance systems.

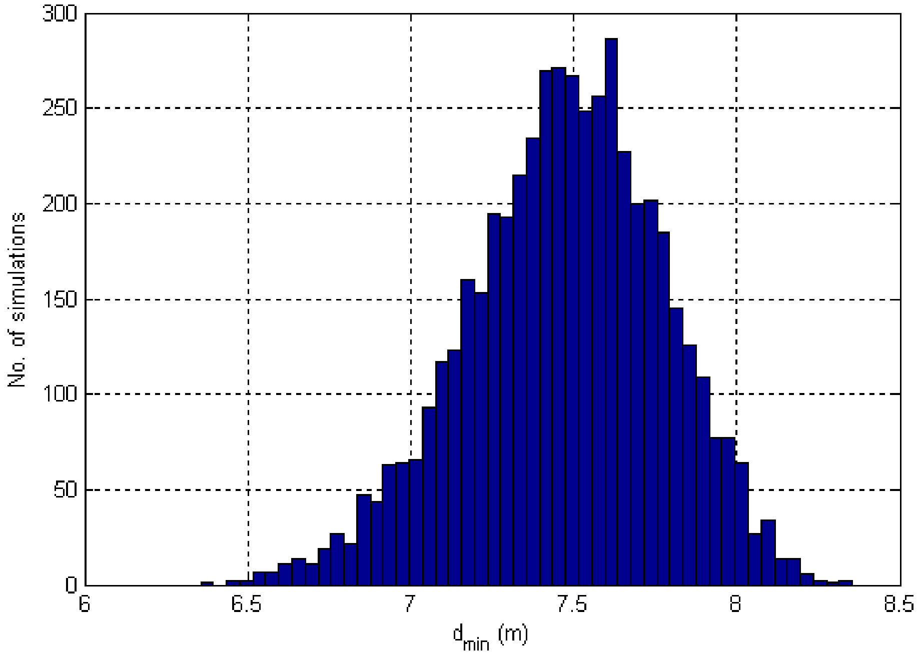

Fig.22 Monte Carlo simulations results

5 Conclusions

Safety is a paramount consideration in developing unmanned vehicles. In this paper, the safety analysis of moving obstacle avoidance systems is presented where optimization-based methods have been developed to automatically search the worst cases. The key idea in this 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 functions like obstacle avoidance. This requires as an optimization algorithm that may guarantee the global optimal solution. To demonstrate the challenges of the problem and the effectiveness of the proposed optimization bases verification process, a Pioneer unicycle robot is chosen for the benchmark study. Kinematic and dynamic equations of the unicycle robot are introduced and the controllers are introduced based on these equations. An inner-outer-loop control architecture is used for path planning, tracking and collision avoidance where a local planner in the outer-loop is developed using the artificial potential field method.

An optimization based automatic search approach is proposed to find the worst-cases and check whether the safety criterion is satisfied under all possible uncertainties. Parametric uncertainties, sensor uncertainties and structural mismatching between the model used for collision avoidance algorithm design and the real vehicle have been addressed. Eight uncertain parameters including the changes of mass, inertia, friction coefficients, side slip and sensor data are considered in this case study. For local optimization methods, different worst-cases have been identified when the optimization started from different initial conditions therefore the optimization solutions do not converge to the global minimum. This implies that the local optimization is not suitable for the verification of collision avoidance algorithms in this case study. Stochastic global optimization algorithms including GA and GLOBAL methods have been applied to the 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, a deterministic global optimization algorithm, DIRECT method, has been investigated for the worst-case analysis. Compared with other global optimization algorithms in this study, DIRECT algorithm can guarantee the worst-cases are found. The results show that it provides a most promising candidate for the optimization based verification process. Furthermore, the Monte Carlo simulations are carried out to verify the proposed methods. The study clearly demonstrates that the optimization based worst-case analysis methods achieve a better performance than the Monte Carlo approach.

References (You need to check the journal reference formnat and make sure the reference format in this paper meet the requirements and are consistent; I think you need to make a lot of change here)

1. S. S. Ge and Y. J. Cui, “Dynamic motion planning for mobile robots using potential filed method”, Autonomous Robots. 13, Kluwer Academic Publishers, (2002), pp. 207-222.

2. P. Raja, S. Pugazhenthi, “Path planning for a mobile robot in dynamic environments”, Int. J. of the Physical Sciences, Vol. 6, No. 20, (Sept 2011), pp. 4721-4731.

3. J. W. Park, H. D. Oh, M. J. Tahk, “UAV collision avoidance based on geometric approach”, SICE Annual Conference, (2008), pp. 2122-2126.

4. E. Masehian, D. Sedighizadeh, “Classic and heuristic approaches in robot motion planning- A chronological review”, Proceedings of World Academy of Science, Engineering and Technology. Vol. 23, (August, 2007), pp. 101-106.

5. J. Kuchar, “Safety analysis methodology for unmanned aerial vehicle (UAV) collision avoidance systems”, the 6th USA/Europe Seminar on Air Traffic Management Research and Development, Baltimore, MD, (June, 2005).

6. M. Althoff, O. Stursberg, M. Buss, “Online verification of cognitive car decisions”, Proceedings of the IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, (June, 2007), pp. 728-733.

7. T. Fraichard, “A short paper about motion safety”, IEEE International Conference on Robotics and Automation, Roma, Italy, (April, 2007).

8. S. Luongo, F. Corraro, U. Ciniglio, V. Di Vito. “A novel 3D analytical algorithm for autonomous collision avoidance considering cylindrical safety bubble”, IEEE Aerospace Conference, MT, USA, (2010).

9. C. De La Cruz, R. Carelli, “Dynamic modeling and centralized formation control of mobile robots”, Paper presented at the 32nd IEEE Conference on Industrial Electronics, Paris, (November, 2006), pp. 3880-3885.

10. S. Srikanthakumar, C. Liu, W. H. Chen, “Optimization-based safety analysis of obstacles avoidance systems for unmanned aerial vehicles”, Journal of Intelligent and Robotic Systems, Vol. 65, Issue. 1-4, (January 2012), pp. 219-231.

11. A. D. Luca, G. Oriolo, “Local Incremental Planning for Nonholonomic Mobile Robots”, Proceeding of IEEE International Conference on Robotics and Automation, San Diego, CA, (8-13 May 1994), pp. 104-110.

12. Mathworks.co.uk

13. J. Holland, “Adaption in natural and artificial Systems”, (University of Michigan press, Ann Arbor, MI 1975). What’s this, a report, paper or thesis?

14. D. Bates, M. Hagstrom, “Nonlinear Analysis and Synthesis Techniques for Aircraft Control”, Springer, (2007), pp. 259-300.

15. J. S. Arora, “Introduction to optimum design”, second edition, Elsevier, Academic Press.

16. 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).

17. Tiber Csendes et al, “The GLOBAL Optimisation Method Revisited”, (University of Szeged, Hungary, September, 2008).

18. D. Jones, “DIRECT Global optimization algorithm”, Encyclopedia of Optimization, Kluwer Academic Publishers, (2001).

19. D. E. Finkel, C. T. Kelley, “Convergence Analysis of the Direct Algorithm”, (North Carolina State University, July, 2004).

20. D. E. Finkel, “DIRECT optimization algorithm user guide”, Center for Research in Scientific Computation, (North Carolina State University, March 2, 2003).

21. GUIDE 98-3/SUPP. 1 Uncertainty of measurement part 3/Supplement 1: Propagations of distributions using a Monte Carlo method [S]. ISO/IEC, (2008).