32nd Congress of the International Council of the Aeronautical Sciences
ICAS 2021 SHANGHAI
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

AUTONOMOUS CONFLICT RESOLUTION ALGORITHM BASED ON ELLIPTIC GEOMETRY

S.S.Rawikara[1], R.A.Sasongko[1], S.A.Munthahar[1]

[1]Aerospace Engineering, Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung, Jl.Ganesha 10, Bandung 40132, Indonesia

Abstract

This paper presents the development of a conflict resolution algorithm that can be implemented in UAV autonomous guidance system so that the UAV has a capability to avoid possible collision with other UAVs that may interfere the flight path. In case of the UAV faces another UAV that is predicted to interfere its path, then the avoidance system will generate a set of new direction reference for correcting the flight path, so that collision can be avoided, and then the UAV can resume its flight to the desired destination or flight path. The proposed conflict resolution approach is based on the utilization of ellipsoid geometry for describing a restricted zone enclosing the interfering UAV, which is identified by the detection system. When the restricted ellipsoid zone has already been defined, the algorithm then determines the locations of contact points on the ellipsoid boundary by solving a linear minimization problem. These points then can be selected as the point for pointing the avoidance direction, that will be the new heading reference for the UAV. The algorithm then is simulated and analyzed in 2 dimensional cases describing conditions when an UAV detects an interfering UAV and has to change its flight direction to avoid a potential collision, and then returns to its desired flight path.

Keywords: ellipsoid, contact point, avoidance

1. Introduction

Autonomous guidance system which provides an ability to a UAV to fly following a predefined or desired flight path [1,2], is a key element of a UAV system which can affects the mission performance of the UAV significantly. In some missions, UAV may be expected to fly in an uncontrolled airspace, where it may unexpectedly interfere with other UAVs that operate in the same area. In this situation, the autonomous system must have a capability to avoid any possible collision, by generating appropriate avoidance maneuver. This means that an algorithm that can compute the required reference for managing an avoidance maneuver must be implemented in the UAV autonomous system.

Some algorithms for defining and characterizing obstructing objects have been proposed and discussed in several literatures, such as in [3], in which an avoidance system utilizing laser rangefinder and PID controller for UAV operating in uncertain environment is elaborated. Another avoidance approach utilizing rangefinder, GPS, and inertial sensor for characterizing obstacles in 3 dimensional domain, especially in hazardous environment, is developed and proposed in [4]. Development and implementation of an algorithm, based on visual information, for avoidance system is discussed for example in [5]. In this work, stereo visual information is exploited for determining and characterizing obstacles, which then is implemented in a real time UAV collision avoidance system.

In this paper, an algorithm for computing corrected direction, required by a UAV to avoid any possible collision with another UAV or moving object, is presented and discussed.

The proposed algorithm works by first quantifying the obstacle by using ellipsoid to define restricted zones which should be avoided by the UAV. After defining the ellipsoid restricted zone, the algorithm computes the flight path corrections by finding waypoints in the boundary of the restricted zone which should be taken by the UAV to avoid the object. The algorithm is designed to be simple and can easily be coupled with the waypoint following system. This avoidance algorithm will be simulated numerically to evaluate its performance in generating the correcting waypoints.

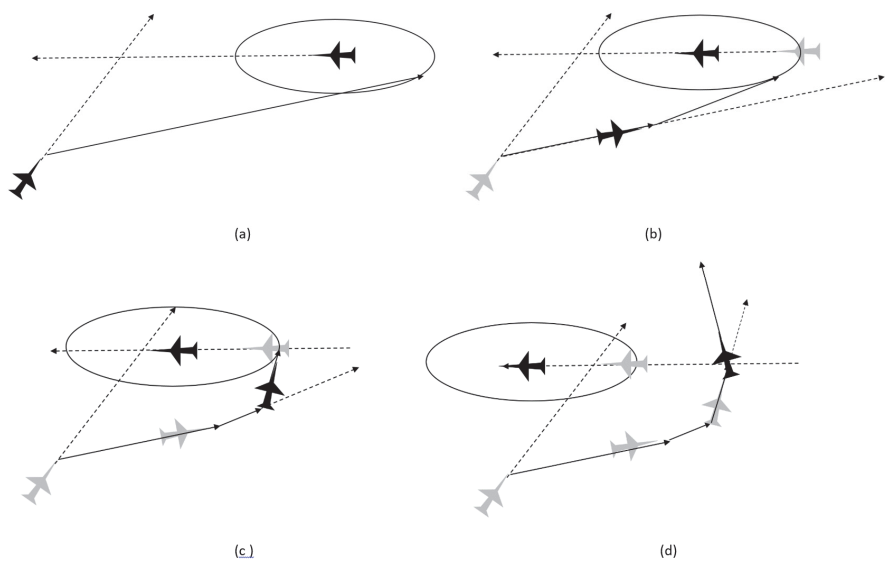

Figure 1 – Conflict avoidance

The algorithm for avoidance system is based on the determination of a restricted zone, that encloses and moves with the interfering UAV (moving object), which must be avoided by the UAV. In the proposed approach, ellipsoid geometry is used for defining the moving restricted zone. Provided that the position and movement of the interfering object is already obtained from the UAV detection system (for example radar), ellipsoid body then can be used for quantifying the restricted zones, which further is required for computing the avoiding direction that should be taken by the UAV to avoid collision.
Once the ellipsoid that encloses and moves with the object is defined, then a set of points in the edge of the restricted zone can be computed accordingly, and treated as the references to determine the avoiding flight direction. Each point on the edge of the zone is obtained by computing the contact points between the ellipsoid and a line (2D case) or a plane (3D case), the gradient of which are orthogonal to the flight directions of the UAV. The main principle of the algorithm is described in Figure 2.

2. Ellipsoid Based Algorithm

2.1 Ellipsoid Zone

Suppose during its flight, a UAV **A** detects an interfering object ***I***. Suppose in this situation, assuming that the interfering object ***I*** will not change its flight direction and speed, then there is a chance that a collision may happen. Hence, suppose the UAV detection system can always determine and track the position of ***I*** at any time instant, then an ellipsoid can be defined that always encloses and moves with ***I***. The dimension of the ellipsoid can be defined by its main axes, r_1 and r_2, whose major axis is defined to be aligned with the object flight direction. Note that additional space, denoted by ε, should be considered when defining r_1 and r_2 for safety reason. The information from detection system can also be used for determining the orientation of the ellipsoid axes, defined as the angle ψ_E, relative to the orientation of the UAV ***A***.
Basically, a 2 dimensional ellipsoid centered at (x_{1c}, x_{2c}) can be defined mathematically as [7]:

$$\bar{x}^T A\bar{x} + B^T\bar{x} = c \tag{1}$$

where $\bar{x} = [x_1 \quad x_2]^T$, $A \in \Re^{2\times1}$ is a symmetric positive definite matrix, $B \in \Re^{2\times1}$ is the vector of the linear term, and $c \in \Re$ is a scalar. It can be shown easily that the matrix A can be represented in the form of :

$$A = V^T \Lambda V \tag{2}$$

where Λ is a diagonal matrix whose diagonal elements are the Eigenvalues of *A*, the values of which represent the length of ellipsoid axes r_1 and r_2. Related to that, the matrix *V*, is an orthogonal matrix, i.e. its columns are orthogonal vectors, which are the corresponding Eigenvectors of *A*. This matrix determines the rotation of the axes of the ellipsoid relative to the orientation axis of the UAV ***A***, and can be computed as:

$$V = \begin{bmatrix} \cos\psi_E & -\sin\psi_E \\ \sin\psi_E & \cos\psi_E \end{bmatrix} \tag{3}$$

So in other words, having the length and orientation of the ellipsoid axes, the matrices Λ and *V* can be determined. Further, the symmetric matrix *A* can be obtained through Equation (2) and used for determining the ellipsoid as defined in Equation (1). It can be observed that the orientation and the location of the center point of the ellipsoid, as described above, must be set the first time the object is detected, or tracked by the UAV detection system. By setting the location and orientation of the ellipsoid at any time, the points on the edge of the ellipsoid can be computed, by considering the UAV flight direction vectors.

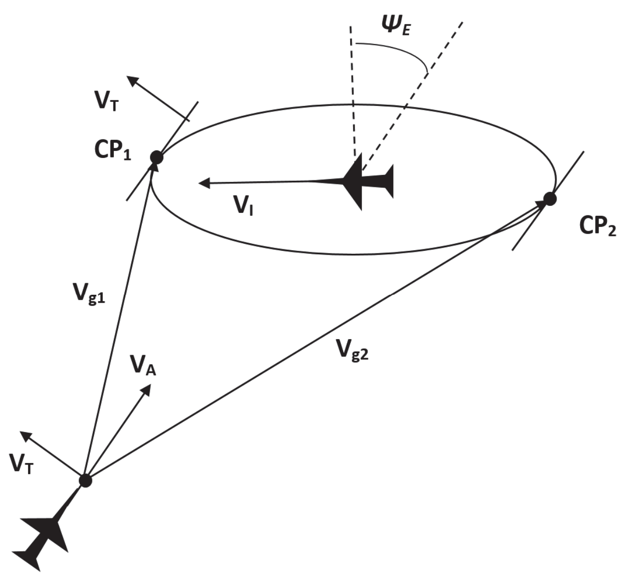

Figure 2 - Avoidance algorithm principle

2.2. Contact Point

It has been stated that once the ellipsoid restricted zone is already defined, then points that later will be used as the reference for avoiding direction can be computed in the edge of the zone.
Suppose $\bar{g}$ is a unit vector perpendicular to the UAV velocity V_A, hence $\bar{g}$ is equal to V_T in Figure 2, then a point $\bar{x}^*$ which is the contact point between the ellipsoid ***E*** with a line ***H*** (for 2D case), whose gradient is represented by its normal vector that is equal to $\bar{g}$, can be obtained from the solution of the following minimization problem [8]:

$$\bar{x}^* = \underset{\bar{x} \in E}{\arg\min} \quad \bar{g}^T \bar{x} \tag{4}$$

where *E* denotes an ellipsoid as described by Equation (1).
Given all parameters of the ellipsoid and the vector $\bar{g}$ are available, the problem (4) can be solved using available methods, giving a contact point as formulated below :

$$CP = \bar{x}^* = -\left[\left(\sqrt{\left(\bar{g}^T A^{-1} \bar{g}\right)} c^{-1}\right) A\right]^{-1} \bar{g} + \bar{x}_C \quad (5)$$

where $\bar{x}_C$ denotes the center point of the ellipsoid restricted zone.

3. Conflict Resolution Scheme

The contact point algorithm further is implemented for determining a set of avoiding points by considering the movement of the ellipsoid zone as it follows the interfering object movement. Recall that in this work it is assumed that the interfering object will not change its flight direction and speed as the avoidance maneuver is executed by the UAV.

Using the contact point computation described in the previous section, the avoiding scheme then is constructed, where the position and flight direction of the interfering object relaative to those of the UAV must be considered. As has been explained, the contact point computation will produce 2 points for every given vector $\bar{g}$. One of these points then is selected by taking into account the relative position and direction of the object with respect to the UAV. The approach that is used to construct the avoidance scheme can be explaineed by observing the figures below.

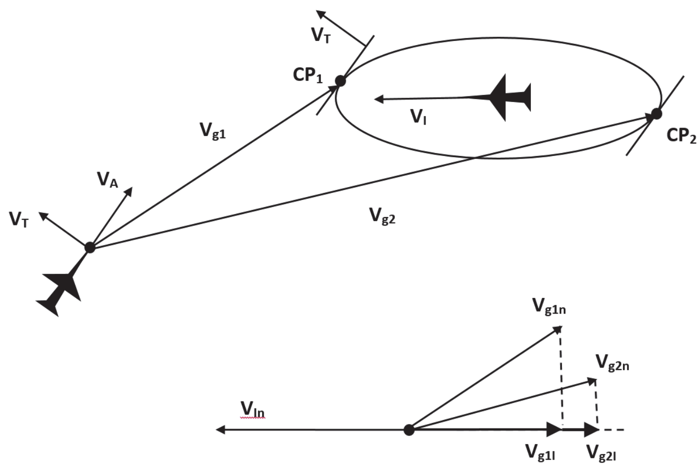

Figure 3 – Head on case

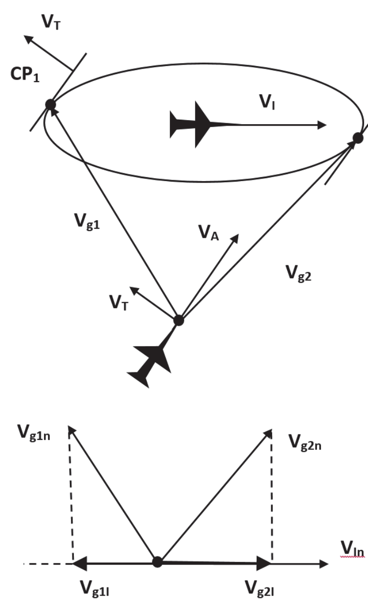

Figure 4 – Tail chase case

In Figure 3, it can be seen the situation when the position and flight direction of the UAV and the interfering object are in such a way such that there are components of both UAV that drive them to getting closer one to another. Suppose in the situation depicted in Figure 3, 2 contact points are obtained, and based on them 2 avoiding directions, V_{g1} and V_{g2}, can be computed. The avoiding vector that will be selected is the one that will make those 2 UAV move away each other faster, or in this case is the vector whose projection to the interfering object velocity V_I is the most negative (since the UAV has velocity components in opposite direction to that of the object). According to this rule, for the case in Figure 3, it can be seen that V_{g2} is the vector that is suitable as the avoiding direction. In Figure 4, suppose the speed of the UAV is faster than the object, then the situation is that the UAV chases the object and at some point may collide with it. In this case, there is a velocity component of the UAV that is in the same direction to that of the object. Again, 2 contact point are obtained, giving 2 avoiding directions, V_{g1} and V_{g2}. Using the same rule as in the case in Figure 3, the direction that makes those 2 object moves away each other is V_{g1}, since its projection to the object velocity vector V_I is negative.

Hence, it can be stated that the chosen avoiding direction $V_g{}^*$ is

$$V_g^* = \{V_{gi} | \min(V_{giI})\} \quad ; \text{i=1,2} \tag{6}$$

where

$$V_{g1I} = V_{g1n}.V_{In} \tag{7}$$

$$V_{g2I} = V_{g2n}.V_{In} \tag{8}$$

Here V_{g1n}, V_{g2n}, V_{In} are the normalized form of V_{g1}, V_{g2}, and V_I.

The process of computing contact point and determining the avoidance direction are repeated until a safe relative position and flight direction of both objects are obtained, or in other words until the UAV completes its avoiding maneuver.

4. Simulation Results

4.1 Avoidance Maneuver Simulation

In this section the proposed collision avoidance algorithm is numerically modeled and simulated. Several cases are simulated to evaluate the characteristic of the developed algorithm. At this stage, the dynamic characteristic of the flying object is still not considered yet.

At the first case, it is simulated the situation as described in Figure 3. A UAV (blue) flies in its desired path when at one point it detects another object (red) whose flight direction has a component that is toward the UAV, hence it has potential to collide with the UAV. The algorithm then computed the contact points and selected the appropriate avoiding direction several times during the execution of avoidance maneuver, until it reach a safe conddition to continue to its desired destination. The sequence of avoiding process is presenteed in Figure 5.

The second case is the one reproducing the situation described in Figure 4. In this simulation, the UAV flies in its flight path when then there is an object that goes into its path in front of the UAV. Since the speed of the UAV is faster than that of the object, then the situation became the one that the UAV chases the object from rear direction, and at some point may colliide.

The algorithm then computed the contact points and the avoiding direction accordingly. The direction that is selected in this case is the one that makes the 2 objects relatively move away each other, and provide a safe avoiding maneuver, as depicted in Figure 6.

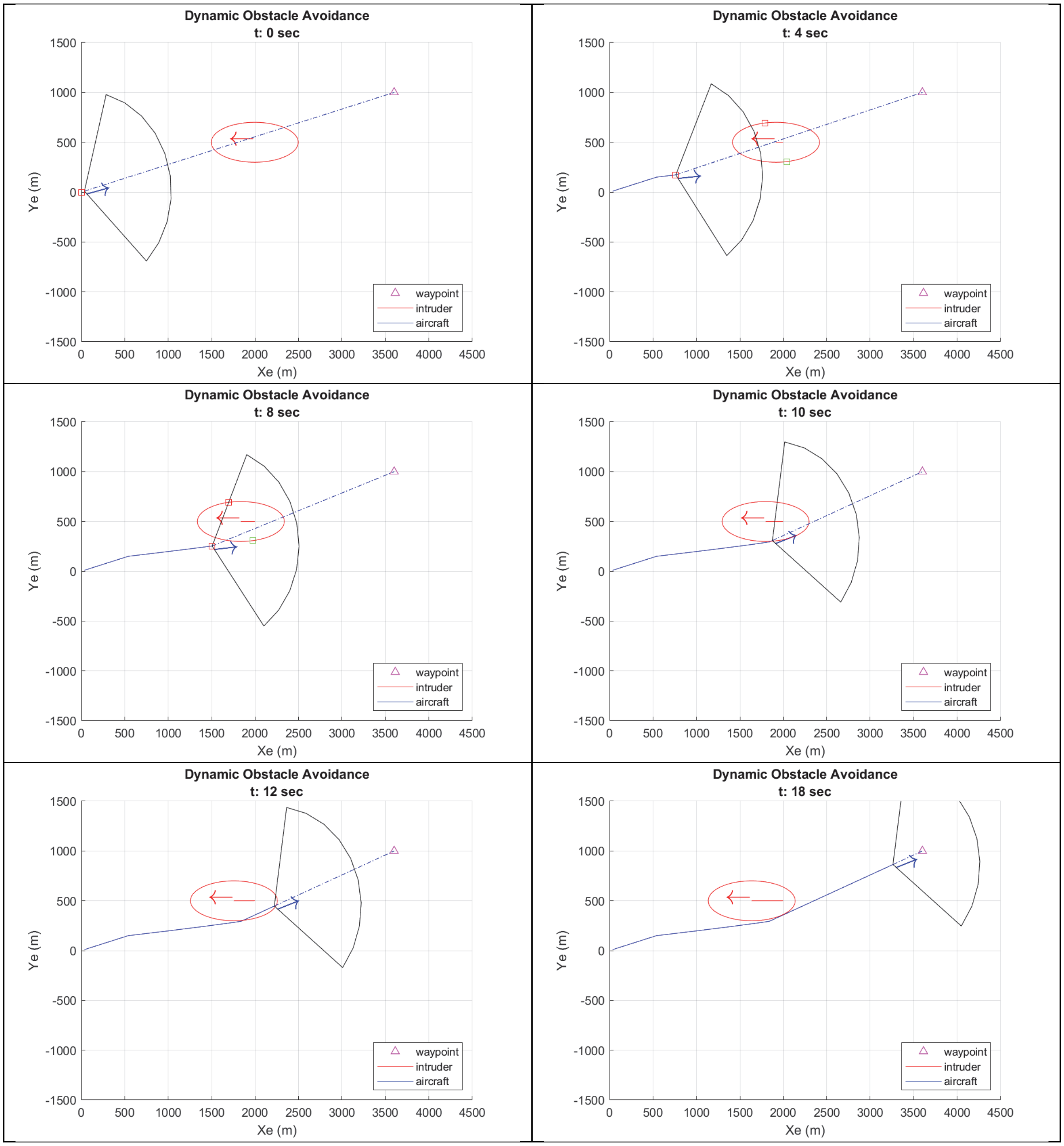

Figure 5 – Simulation – Case 1

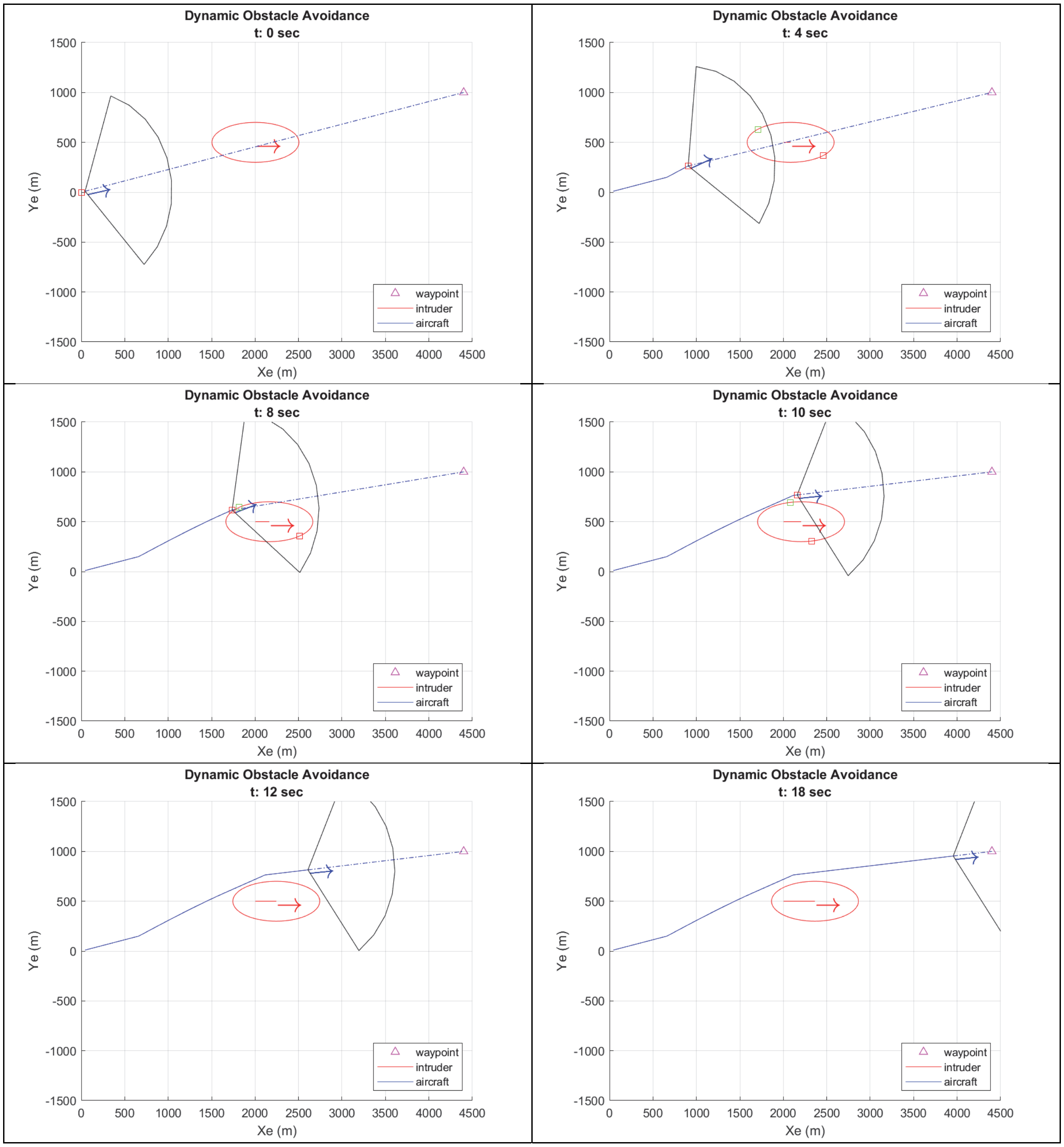

Figure 6 – Simulation – Case 2

4.2 Unsafe Check

During avoidance maneuver, when the obstacle is registered in the radar, contact point calculation from the avoidance algorithm will also active. However, the obstacle itself may positioned in such a way that doesn't obstruct the flight path of the aircraft. This condition is also present when the aircraft is close to the obstacle, as shown in figure below.

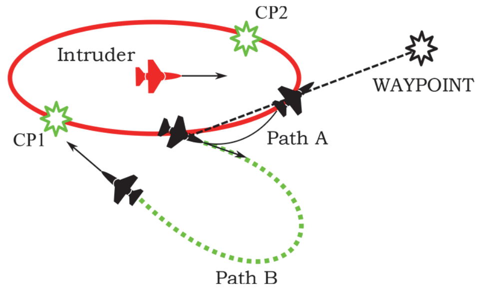

Figure 7 – Special Case

As shown in the figure above, the aircraft A can directly fly to waypoint, since the flight path is not obstructed by the intruder. Even so, the aircraft A will follow contact point calculation, since the obstacle is close to the aircraft, hence registered by the radar of the aircraft A. To alleviate such condition, an algorithm to check whether the aircraft is unsafe from the obstacle or not is proposed.

The unsafe check algorithm is developed from the contact point calculation algorithm, which uses the vector of the center between the aircraft and the obstacle, instead of the velocity vector. Define $\overline{V_c} \in \mathbb{R}^{2\times1}$as:

$$\overline{V_c} = \begin{bmatrix} x_{obs} - x_{acf} \\ y_{obs} - y_{acf} \end{bmatrix} \tag{9}$$

To find whether the current flight path to waypoint is safe from the obstacle, first a vector perpendicular to $\overline{V_c}$ defined as $\bar{g} \in \mathbb{R}^{2\times1}$ is calculated using the following equation:

$$\begin{aligned} \overline{g_1} &= \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & \sin\left(\frac{\pi}{2}\right) \\ -\sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix} \overline{V_c} \\ \overline{g_2} &= \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & \sin\left(\frac{\pi}{2}\right) \\ -\sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix}^T \overline{V_c} \end{aligned} \tag{10}$$

The two perpendicular vector represents both left and right perpendicular vector of $\overline{V_c}$. From the perpendicular vector, the contact point position $\bar{x}_s \in \mathbb{R}^{2\times1}$ can be calculated using the same equation as the contact point algorithm for obstacle avoidance, as following:

$$\begin{aligned} \bar{x}_{s1} &= -\left(\sqrt{\overline{g_1}^T M^{-1} g_1 LM}\right)^{-1} + \bar{x}_{obs} \\ \bar{x}_{s2} &= -\left(\sqrt{\overline{g_2}^T M^{-1} g_2 LM}\right)^{-1} + \bar{x}_{obs} \end{aligned} \tag{11}$$

The contact point is the used to calculate heading limit of the flight path which being obstructed by the obstacle, which is defined as:

$$\psi_{u1} = \tan^{-1}(\frac{x_{s1} - x_{acf}}{y_{s1} - y_{acf}})$$
$$\psi_{u2} = \tan^{-1}(\frac{x_{s2} - x_{acf}}{y_{s2} - y_{acf}}) \tag{12}$$

By using ψ_{u1} and ψ_{u2}, the condition of heading reference ψ_{ref} which should be followed by the aircraft can be defined using the following condition:

$$\psi_{ref} = \begin{cases} \psi_{wp}, & \min\{\psi_{u1}, \psi_{u2}\} \leq \psi_{wp} \; and \; \max\{\psi_{u1}, \psi_{u2}\} \geq \psi_{wp} \\ \psi_{cp}, & else \end{cases} \tag{13}$$

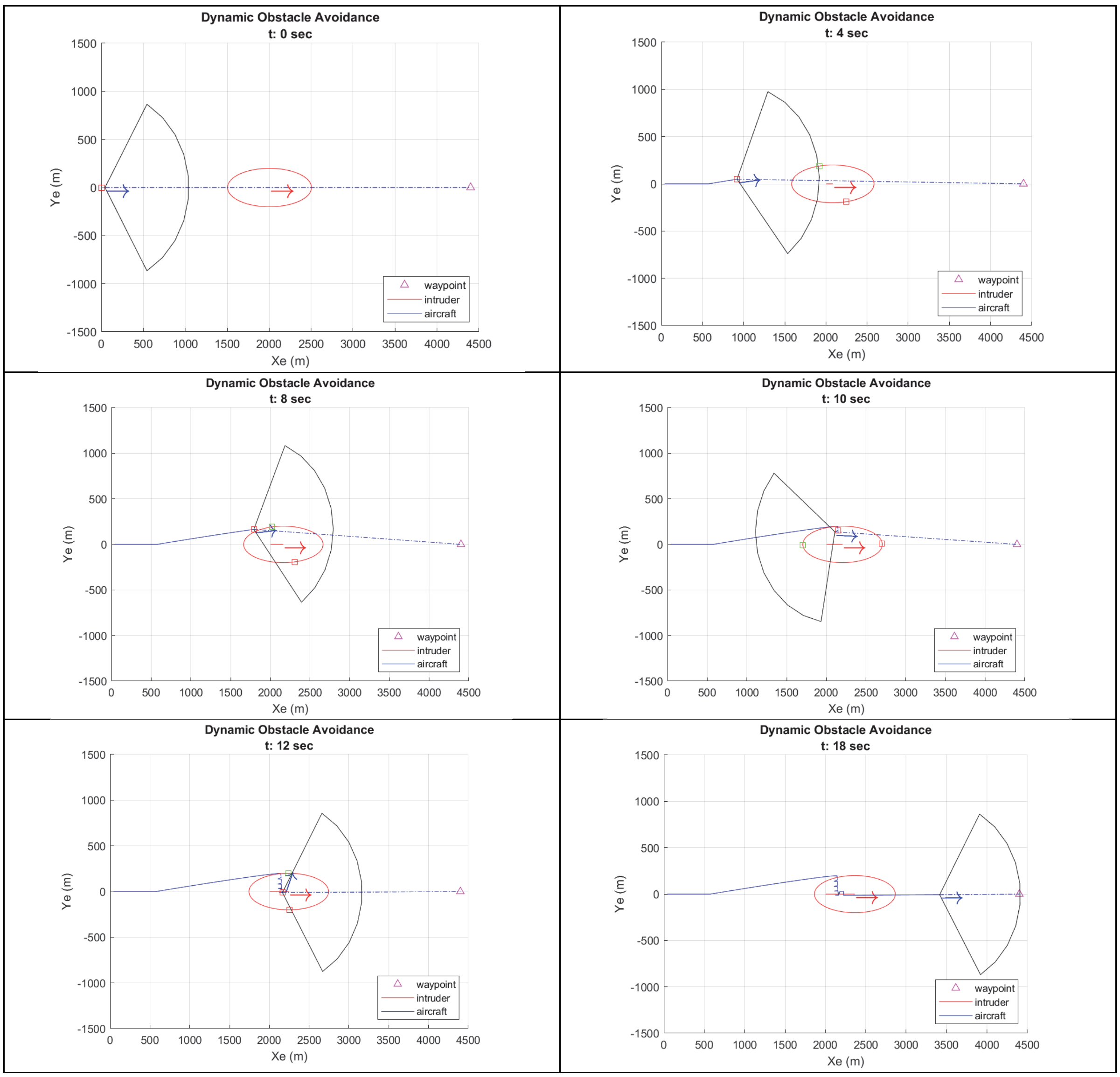

Figure 8 – Tail-chase without Unsafe Check

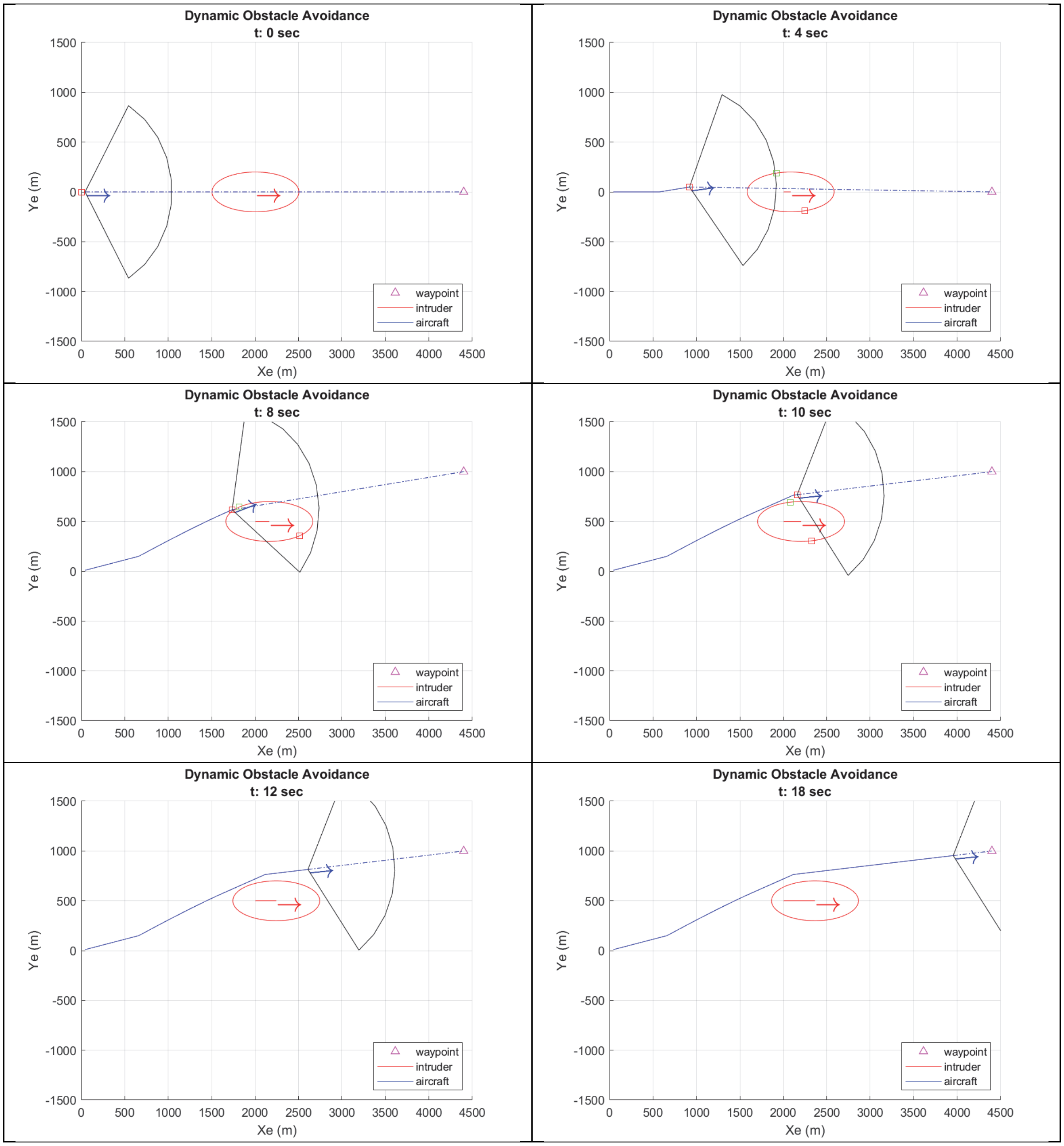

Figure 9 – Tail-chase with Unsafe Check

The role of unsafe check is demonstrated in the simulation results depictd in Figures 8 and 9. It can be seen that there is situation when the algorithm produces avoidance directions which vary significantly at each step during the maneuver, giving a series of abrupt changes in flight directions that ultimately failed to steer the UAV to avoid the object. When the unsafe check routine is activated, it can be seen that a safe avoidance maneuver can be generated.

5. Conlusions

An algorithm for providing collision avoidance capability on UAV autonomous guidance system is presented and evaluated. The algorithm, which is based on the implementation of ellipsoid restricted zone, which is extended from the implementation of the same approach for non-moving obstacle. The algorithm determine an ellipsoid area enclosing the detected interfering object (intruder). The algorithm then compute the contact points on the ellipsoid edge and use these points as the reference for determining the avoiding direction. The computation of avoiding direction is repeated during the maneuver so that a set of updated flight direction can be obtained to steer the UAV avoiding the intruder safely.

The simulation results show that the proposed algorithm can provide a good avoidance maneuver for the UAV to avoid the interfering object. There is still conditions where the algorithm may generate wildly varying direction during the maneuver, that ultimately makes the intended avoidance movement can not be obtained. In this case, some additional procedure can be embedded into the algorithm, one of which is by executing unsafe check procedure. This additional routine has been demonstrated, and the results show that it can resolve the problem.

Further the algorithm can be extended to 3 dimensional cases, where the types of avoidance maneuver can be varied to include motion in dimensional space. This can provide a way to add more degree of freedom in determining the safe and effective avoiding maneuver, by taken into account the UAV maneuver capability and Imiitations.

6. Copyright Statement

References

[1] R.Rysdyk, UAV Path Following for constant line-of-sight, 2nd AIAA .Unmanned Unlimited. Systems,Technologies, and Operations Aerospace, Land, and Sea Conference, Paper #6626, September, 2003, San-Diego, CA.

[2] R.Cunha, et.al., A Path-Following Preview Controller for Autonomous Air Vehicles, AIAA GNC Conference. Keystone, CO., 2006

[3] C. Kownacki, Guidance and obstacle avoidance of MAV in uncertain urban environment, The European Micro Aerial Vehicle Conference and Flight Competition 2009, Delft, the Netherlands, September 14–17, 2009

[4] M. S. Geyer, E.N. Johnson, 3D Obstacle Avoidance in Adversarial Environments for Unmanned Aerial Vehicles, AIAA Guidance, Navigation, and Control Conference and Exhibition, 21-24 August 2006, Keystone, Colorado

[5] J. Byrne, M. Cosgrove, and R. Mehra, Stereo Based Obstacle Detection for an Unmanned Air Vehicle, Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA), May 15-19, 2006, pp. 2830 – 2835

[6] J.Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, Roskam Aviation and Engineering Corp., Kansas, 1982.

[7] T.Apostol, Calculus Vol. 2, 2nd edition, John Wiley & Sons, New York, 1968

[8] D.Bertsekas, Convex Analysis and Optimization, Athena Scientific, Massachusett, 2003

[9] M.Corless and A.Frazho, *Linear Systems and Control : Operator Approach*, CRC Press, London ; 2003.

[10] MATLAB and SIMULINK user guide, Mathworks inc., 2007