Dynamic obstacle avoidance system for the unmanned aerial vehicle (UAV)

S S Rawikara*, and R A Sasongko

Flight Physics Research Group, Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung, Indonesia.

*E-mail: sahisnu@ae.itb.ac.id

Abstract. This paper presents a method for collision avoidance for moving obstacle based on ellipsoid geometry. Ellipsoid algorithm itself has been implemented in small unmanned aerial vehicle (UAV) for static obstacle avoidance. In extending the algorithm to dynamic object, two adjustments have been made, first is to calculate safe heading and selection of head or tail contact point. Using these two adjustments, the algorithm successfully generates safe waypoint for dynamic obstacle avoidance and select which direction to use, whether it is to the head or the tail of obstacle moving direction.

Keywords: collision avoidance, ellipsoid zone, flight guidance

1. Introduction

In recent years, UAV’s has been deployed not only for military purpose but also for civilian use, ranging from agricultural mapping, asset surveillance and inspection, and even logistic delivery for remote areas. The relatively low operating cost and simplicity of UAV’s deployment are main advantage of UAV’s over manned aircraft, especially for operation in remote areas and dangerous missions [1]. Nowadays, there are efforts to integrate UAV’s into civil airspace, which would be beneficial especially for air taxi operation and urban drone logistics delivery. However, to fully integrate UAV’s into civil airspace, UAV’s will need to comply to the policies of aviation authorities [2]. Even though UAV’s now able to autonomously operate from take-off to landing, conflict situation for both between UAV’s and manned aircraft still presents challenging problem [3]. There have been research in developing conflict resolution algorithm for UAV using several approaches and paradigms.

Generally, there are two different approaches in collision avoidance algorithm [4]. The first category is the global planner, which requires knowledge about the aircraft operation environment. The advantage of this approach is that the trajectory generated by the algorithm can always yield to optimal path, even though this approach have difficulties in dealing with sudden changes in the environment of UAV operation. Another approach is the local trajectory planner, which the algorithm runs continuously to deal with unknown obstacle during the flight. In [5] discuss the evolution of UAV obstacle avoidance philosophy, with several methods being evaluated, namely graph search, rapidly exploring random tree, minimum effort guidance, and potential field method. One of the main conclusion in this research is that even though an optimal path can be generated from the collision avoidance algorithm, the computational time required to generate such path must be taken into account, since fixed-wing UAV doesn’t have the luxury to stop in mid air. Proportional navigation which is popular for missile termination guidance is also adapted to generate avoidance guidance in [6]. In this research, Proportional Navigation Collision Avoidance Guidance (PNCAG) is generating guidance velocity vector based on object position and velocity, with predetermined object size. Another approach for real time collision avoidance is presented in [7]. In this paper collision avoidance trajectory is generated using B-splines, based on optimal control...
problem. While the generated trajectory can guarantee optimal solution, the optimizer could be trapped in local minimal, hence increasing computational time required to generate the trajectory. There is also an approach using reachable set as elaborated in [8]. This research focused on defining control constraints set that are reachable by the UAV, so that the maneuver can always be executed by aircraft. However, the computation requirement for solving reachable set is relatively expensive problem and requires significant computation time.

In this paper, a reactive collision avoidance method is developed using ellipsoid geometry to determine the object size and compute safe waypoint to avoid collision. First, the methodology of the obstacle avoidance for static object is elaborated in the methodology section, also the extension from static to dynamic obstacle avoidance and the adjustments that have been made. Afterward, the method that has been explained is tested in simulation. Finally, results of the simulation for dynamic obstacle avoidance algorithm will also presented and analyzed.

2. Methodology

In this section, the collision avoidance method will be discussed with short introduction to static object avoidance method. Its extension from static to dynamic object will be presented with the adjustment of contact point selection from static to dynamic object avoidance. From the following dynamic object avoidance method, selection of contact point will also be discussed.

2.1. Static Obstacle Avoidance

Suppose a UAV is going from point A to point B (Figure 1). Along the way, an obstacle is detected, so the UAV need to compute a new waypoint to avoid the obstacle. This waypoint can be calculated given that the size and the orientation of the obstacle is known. As in [9], the obstacle is defined as ellipsoid, and using the velocity vector of the UAV a new waypoint that lie in boundary of the ellipsoid can be generated.

A 2-D ellipsoid of the obstacle can be defined as:

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

where $\bar{x} = [x_1, x_2]^T$, $A \in \mathbb{R}^{2 \times 2}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{2 \times 1}$ is the vector of linear term, and $c \in \mathbb{R}$ is a scalar. The eigenvalues of $A$ matrix representing length of major and minor axis of ellipsoid. The ellipse can be rotated to represent obstacle geometry using matrix $V$ which defined as:

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

where $\psi_E$ is the obstacle heading. To find contact point, following formula is used [10]:
\[ \bar{x}_{cp} = -\left[ \sqrt{(V^TA^{-1}Vc^{-1})A} \right]^{-1} V + \bar{x}_{obs} \]  

(3)

where \( \bar{x}_{obs} \) is the obstacle position relative to the UAV and \( \bar{x}_{cp} \) is the contact point for safe waypoint guidance. Since the vector used in contact point calculation is based normal vector of UAV heading, there will be two solutions of contact points, which corresponds to +90- and -90-degree rotation of velocity vector. In static obstacle avoidance, contact point can be selected which have minimum heading change.

2.2. Extension from Static to Dynamic

In general, calculation for contact point between static and dynamic is the same, with moving obstacle size is becoming the parameter for ellipse and heading of the obstacle is heading of the ellipse. From static obstacle avoidance that calculation of contact point will always yield to two contact point solutions. While for static obstacle avoidance the selection of contact point can be simply selected based on difference between current heading and contact point heading reference, in moving obstacle avoidance this is not the case. Since the obstacle is moving, selection of contact point can lead dangerous waypoint crossing the ellipsoid zone. In the next section, proposed methods of selecting contact point will be elaborated.

2.3. Contact Point Selection

2.3.1. Unsafe Heading Requirement

For dynamic obstacle avoidance, since the ellipse is moving, contact points calculated from ellipse algorithm can lead to dangerous position. As shown in Figure 2, because of the velocity vector from the UAV, both contact points calculated from avoidance algorithm are not guiding UAV to safe position. Instead of following calculated contact point, UAV should directly go to final waypoint as the obstacle is already avoided.

To mitigate this behaviour, an unsafe heading calculation for determining which heading reference to follow is proposed. First, limit of obstacle is calculated based on relative vector between UAV and the obstacle. From relative vector of UAV and obstacle, contact points is determined using same formula for finding contact points in avoidance scheme. Relative vector between obsta

\[ \bar{v}_c = \begin{bmatrix} x_{obs} - x_{acf} \\ y_{obs} - y_{acf} \end{bmatrix} \]  

(4)

where \( \bar{v}_c \in \mathbb{R}^{2x1} \) is the relative range. To find unsafe contact point, the following formula same as static avoidance is used:

\[ \bar{x}_{sv} = -\left[ \sqrt{\bar{g}^T M^{-1} \bar{g}} L^{-1} M \right]^{-1} \bar{g}^T \left( \begin{bmatrix} x_c \\ y_c \end{bmatrix} - \bar{x}_{obs} \right) \]  

(6)

\[ \psi_{unsafe} = \tan^{-1}\left( \frac{\bar{x}_{sv} - x_{acf}}{\bar{y}_{sv} - y_{acf}} \right) \]  

(7)
In dynamic avoidance, \( x_{av} \in \mathbb{R}^{2 \times 1} \) is the contact point. After that, the heading for computed contact point from avoidance scheme is compared between both unsafe heading reference. If the heading for avoidance is inside unsafe heading reference, then the heading from avoidance become the heading reference for the guidance system. The following criteria is used to determine if the heading reference is inside unsafe heading:

\[
\min(\psi_{\text{unsafe}}) \leq \psi_{\text{waypoint}} \land \psi_{\text{waypoint}} \leq \max(\psi_{\text{unsafe}})
\]  

(8)

2.3.2. Head-or-tail Selection

In dynamic obstacle, selection of contact points will determine whether UAV will go same or opposite direction with the obstacle. Example of importance in contact point selection can be seen in Figure 2, where the contact point selected by the algorithm is into the head of the obstacle direction, even though it is still safe to use that contact point for waypoint, other contact point which is in tail position of the obstacle is relatively safer.

First, the preference for head or tail selection of contact point is defined. Then, relative heading between UAV and obstacle is calculated using:

\[
\psi_{rel} = w_{360}(\psi_{\text{obs}} - \psi_{\text{acf}})
\]

(9)
where $w_{360}$ is a function to wrap from 0 to 360 degrees. After relative heading is found, then the orientation of the obstacle needs to be checked whether it is in quadrant 1 and 2 or in quadrant 3 and 4, using the following:

$$c_{\text{factor}} = 1 - 2 \cdot (180 < \psi_{\text{rel}} \land \psi_{\text{rel}} < 360)$$  \hspace{1cm} (10)

**Figure 3. Head Case Waypoint**

Value check of relative heading will give true (1) or false (0), so eq. will result in 1 if it is true or -1 if it is false. From the value check of relative heading, to determine normal vector that will be used in contact point calculation, the following transformation is used:

$$g = \begin{cases} 
\cos(c_{\text{factor}} \cdot \pi/2) & - \sin(c_{\text{factor}} \cdot \pi/2) \\
\sin(c_{\text{factor}} \cdot \pi/2) & \cos(c_{\text{factor}} \cdot \pi/2) 
\end{cases} \text{ if tail}$$

$$g = \begin{cases} 
\cos(-c_{\text{factor}} \cdot \pi/2) & - \sin(-c_{\text{factor}} \cdot \pi/2) \\
\sin(-c_{\text{factor}} \cdot \pi/2) & \cos(-c_{\text{factor}} \cdot \pi/2) 
\end{cases} \text{ if head}$$  \hspace{1cm} (11)

In general, the addition of unsafe heading check and head-or-tail selection as flowchart is shown in Figure 4.
3. Results and Discussions

In this section the collision avoidance algorithm is demonstrated with several cases. This simulation is only consider the kinematics of the UAV in 2D X-Y plane, because the focus is in testing the algorithm in various cases. In this simulation, initial speed and heading of the UAV is fixed, while the speed and heading of the obstacle is calculated so that collision will occur at 20 second.

Three cases will be considered, the first case is perpendicular case to show that without unsafe heading check, waypoint from obstacle avoidance calculation will lead to oscillation. Second is head-to-tail case, where the obstacle is in pursuit by the UAV, and last is the head-to-head case.

3.1. Perpendicular Case

In this case, the obstacle is moving perpendicular to the UAV. In the Figure 5, unsafe heading check is turned off. The waypoint generated by static obstacle avoidance algorithm will lead to oscillation. This is caused by the change of sign while calculating normal vector from ellipse contact point. In Figure 6, using unsafe heading check as described in section 2.3.1, UAV can return to its original waypoint.

This has shown that even the generated waypoint from ellipse algorithm can give a safe guidance for the UAV, since the obstacle is moving, there is a chance that the UAV is trapped in oscillation because of switching in normal vector of UAV velocity. Unsafe heading check ensure that the contact point given by ellipsoid algorithm is the actual safe waypoint from obstacle.
3.2. Head-or-Tail Selection

In this simulation, the selection of head or tail contact point is shown in Figure 7 and Figure 8. Simulation parameter of both the UAV and the obstacle are the same. Initial condition in both head and tail simulation are also the same. Figure 7 shows that by selecting head as the contact point preference, UAV will try to avoid the obstacle by going to the same direction with the obstacle.

This will lead to relatively smaller heading rate, but also by going to the same direction with the obstacle, UAV will need to regulate its velocity since at some point UAV will overtake the obstacle to avoid the obstacle.
Figure 6. Parallel Case with Unsafe Heading Check

Figure 7. Head Contact Point Selection
In Figure 8, the UAV is going to opposite direction of the obstacle by selecting tail as the contact point preference. In this case, since the direction is in opposite direction of the obstacle, the UAV is avoiding the obstacle at much faster rate than head case. But depending on the distance of the obstacle and the UAV, larger heading rate might be needed so that the UAV will be able to reach calculated safe waypoint at given time.

4. Conclusions

Collision avoidance system for dynamic obstacle has been developed and simulated. Although the static obstacle avoidance can be readily extended to dynamic avoidance problem, there are adjustments that needs to be done. There are two adjustment that proposed, first is to use the vector between the UAV and obstacle to calculate unsafe heading check so that oscillation is avoided and using head and tail criteria in selection of safe waypoint alternative to accomodate direction of obstacle movement. Next, the effect of tail and head selection for the avoidance will be explored further so that the algorithm can automatically determine in which direction will give the optimal route for the avoidance. Also, implementation of avoidance algorithm to aircraft dynamic model as guidance will be investigated.

Acknowledgements

We gratefully acknowledge the support from Flight Physics Ressearch Group Institut Teknologi Bandung for the P3MI funding.

References

[1] S. M. Adams and C. J. Friedland, “A SURVEY OF UNMANNED AERIAL VEHICLE (UAV) USAGE FOR IMAGERY COLLECTION IN DISASTER RESEARCH AND MANAGEMENT,” p. 9.

[2] J. S. McCarley and C. D. Wickens, “Human Factors Implications of UAVs in the National Airspace,” p. 63.
[3] P. Riseborough, “Automatic Take-Off and Landing Control for Small UAV’s,” p. 9.

[4] H. Alturbeh, “COLLISION AVOIDANCE SYSTEMS FOR UAS OPERATING IN CIVIL AIRSPACE,” Cranfield University, 2014.

[5] A. Mujumdar and R. Padhi, “Evolving Philosophies on Autonomous Obstacle/Collision Avoidance of Unmanned Aerial Vehicles,” Journal of Aerospace Computing, Information, and Communication, vol. 8, no. 2, pp. 17–41, Feb. 2011, doi: 10.2514/1.49985.

[6] S.-C. Han and H. Bang, “Proportional Navigation-Based Optimal Collision Avoidance for UAVs,” Journal of Control, Automation and Systems Engineering, vol. 10, no. 11, pp. 1065–1070, Nov. 2004, doi: 10.5302/J.ICROS.2004.10.11.1065.

[7] H. Alturbeh and J. F. Whidborne, “Real-time obstacle collision avoidance for fixed wing aircraft using B-splines,” in 2014 UKACC International Conference on Control (CONTROL), Loughborough, UK, Jul. 2014, pp. 115–120, doi: 10.1109/CONTROL.2014.6915125.

[8] Y. Zhou and J. S. Baras, “Reachable set approach to collision avoidance for UAVs,” in 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Dec. 2015, pp. 5947–5952, doi: 10.1109/CDC.2015.7403154.

[9] R. A. Sasonkgo and S. S. Rawikara, “Collision Avoidance System for Small Unmanned Aerial Vehicle (UAV),” p. 6.

[10] D. P. Bertsekas, Constrained optimization and Lagrange multiplier methods. Belmont, Mass: Athena Scientific, 1996.