# Engineering Notes

## Reactive Collision Avoidance Using Nonlinear Geometric and Differential Geometric Guidance

Anusha Mujumdar[*] and Radhakant Padhi[†]
*Indian Institute of Science, Bangalore 560012, India*

DOI: 10.2514/1.50923

## I. Introduction

UNMANNED Aerial Vehicles (UAVs) hold good promise for autonomously carrying out complex civilian and military operations. However, many of these missions require them to fly at low altitudes, making them vulnerable to collision with both stationary as well as moving obstacles. Hence, it is vital that UAVs are equipped with autonomous capability to sense and avoid collisions, especially for the pop-up threats. When such a threat is sensed and a collision is predicted within a short time ahead, the UAV should be able to react and maneuver away quickly so that the collision is avoided. An algorithm which can assure such a maneuver is called a "reactive collision avoidance algorithm." Since the available reaction time in such a scenario is usually is small and UAVs are usually limited by computational resources, such an algorithm should also be computationally very efficient (it should preferably be noniterative). It is also required that while maneuvering away, it should not maneuver too much away from the obstacle either. This is both to avoid collision from other nearby obstacles and not to compromise on the overall mission objective.

There are various attempts in the literature to develop algorithms for collision avoidance purpose, many of which are inspired from global path planning algorithms. The artificial potential field method is such an approach where the motion of the vehicle is guided under the influence of a potential field. The potential field (which is essentially a cost function) is designed in such a way that obstacles have repulsive fields while the destination has an attractive field. The safe path of the UAV is then found by optimizing the carefully selected cost function. To tune this basic philosophy for reactive collision avoidance, a model predictive control-based algorithm has been proposed in the literature. This algorithm essentially assures path following under safe conditions (i.e. if no collision is predicted in the near future) and invokes the potential field function when new collisions are sensed. However, in potential field based techniques the associated optimization process is typically done in an iterative manner. Because of this they are usually computationally intensive and hence are not suitable for reactive collision avoidance of airborne UAVs in general.

A promising algorithm in collision avoidance and global path planning is the philosophy of rapidly-exploring random tree (RRT) [1], which has also been used for reactive collision avoidance. However, there are many concerns about the RRT approach, which can largely be attributed to the random nature of the algorithm. For example, the path predicted by RRT is usually a sting of connected straight lines that does not reflect the path followed by a vehicle with nonholonomic constraints. More important, it is a probabilistic approach and hence there is no guarantee of finding a feasible path within a limited finite time. Other graph search algorithms such as best-first search are also implemented for reactive collision avoidance [2]. However, this is not systematic approach and could result in the algorithm searching far too many nodes under some conditions. Moreover, precomputing motion primitives and saving them in a lookup table is infeasible for UAVs, which are usually resource-limited.

An interesting perspective to collision avoidance problem is the minimum effort guidance [3], where an optimal control-based approach has been proposed after applying the collision cone philosophy to detect collisions. This method is computationally nonintensive as a closed form solution has been proposed. Even though this is an interesting idea, by minimizing the lateral acceleration, perhaps it imposes unwanted extra constraint on the problem formulation as reactive collision avoidance problems do not necessarily have to be carried out with minimum lateral acceleration. More important, one can observe that this formulation only assures position guarantee and no constraint is imposed on the velocity vector. Hence, even though it guides the vehicle to a carefully selected target point on the safety boundary (we call it the "aiming point"), it causes the vehicle to maneuver until this point. This can be risky as the vehicle may enter the safety ball before reaching the aiming point.

Even though the collision cone based aiming point philosophy is a very good idea, the authors of this Note strongly believe that instead of only position guarantee, rather the velocity vector should be aligned towards the aiming point as soon as a collision is detected (which will automatically lead to position guarantee as well). Towards this objective, two new nonlinear guidance laws are proposed in this Note, which are named as nonlinear geometric guidance (NGG) and differential geometric guidance (DGG). These guidance laws are inspired by the philosophy of "aiming point guidance" (APG) [4], which has been proposed in missile guidance literature. It turns out that the APG is a simplified case of the NGG where the associated *since* function is replaced by its linear approximation (hence, for a systematic discussion, it is renamed as linear geometric guidance (LGG) in this Note). Both of the guidance algorithms proposed in this Note quickly align the velocity vector of the UAV along the aiming point within a part of the available time-to-go, which ensures quick reaction and hence safety of the vehicle. The main feature of this philosophy is that they effect high maneuvering at the beginning, causing the velocity vector of the UAV to align with the aiming point direction quickly and then settling along it. Therefore there is no need to maneuver all the way until the aiming point is reached and hence the chance of the UAV entering into the safety ball is minimized.

Using the point of closest approach (PCA) [5], the proposed NGG and DGG algorithms have also been extended for collision avoidance with moving obstacles in both cooperative as well as ignorant scenarios. Mathematical correlations between the guidance laws have also been established, which show that the NGG and DGG are exactly correlated to each other with appropriate gain selections, while the LGG is an approximation of DGG. A "sphere-tracking algorithm" is also proposed in this Note where the UAV is guided to track the surface of the safety sphere whenever a brief violation of the safety boundary occurs after reaching the aiming point because of the location of the next aiming point (which may include the target in

Presented as Paper 2010-8315 at the AIAA Guidance, Navigation and Control, Toronto, 2–5 August 2010; received 26 May 2010; revision received 4 October 2010; accepted for publication 6 October 2010.

[*] Former Project Assistant, Department of Aerospace Engineering; anushamujumdar87@gmail.com.

[†] Associate Professor, Department of Aerospace Engineering; padhi@aero.iisc.ernet.in.

absence of obstacles). This ensures additional safety. An autopilot lag compensation logic is also proposed in this Note to make the proposed guidance laws more effective.

## II. Reactive Collision Avoidance Problem

### A. Collision Detection with Stationary Obstacles

The collision cone approach is an effective tool for both detecting a collision as well as for finding an alternate direction of motion to avoid it. The construction of the collision cone is shown in Fig. 1.

$X_r$ is relative distance between the UAV and the obstacle, and $V$ is total velocity. A collision cone is constructed by dropping tangents from the UAV to the circle $\beta$. If the velocity vector $V$ lies within the collision cone, the UAV will violate $\beta$ in due course and result in collision.

$V$ can be expressed in terms of the tangents $r_1$ and $r_2$ as

$$V = ar_1 + br_2 \tag{1}$$

The collision criterion is stated as, If $a > 0$ AND $b > 0$ then the obstacle under consideration is said to be critical. The aiming point is then found from the collision cone. First, the tangents are given as $r_1 = X_r + du_1$ and $r_2 = X_r + du_2$, where $u_1$ and $u_2$ are the unit vectors perpendicular to the tangents. Then, the aiming point is determined as follows:

$$\text{If } a > b, \quad X_{\text{ap}} = X_v + r_1 \qquad \text{If } b > a, \quad X_{\text{ap}} = X_v + r_2 \tag{2}$$

Since the velocity in $x$-direction is assumed constant, the $t_{\text{go}}$ can be found as follows:

$$t_{\text{go}} = \frac{1}{u}[(X_{\text{ap}})_x - (X_v)_x] \tag{3}$$

The expressions for $a$ and $b$ are derived next. From Fig. 1, note that $r_i$ and $u_i$ are perpendicular:

$$r_i \cdot u_i = (X_r + du_i) \cdot u_i = 0 \qquad i = 1, 2 \tag{4}$$

Since $u_i \cdot u_i = 1$,

$$X_r \cdot u_i + d = 0 \tag{5}$$

The vector $u_i$ is defined in a two-dimensional plane containing $X_r$ and $V$ and can therefore be expressed as

$$u_i = \alpha_i X_r + \beta_i V \tag{6}$$

Using this in Eq. (5),

$$\alpha_i = -\frac{\beta_i(X_r \cdot V) + d}{\|X_r\|^2} \tag{7}$$

Further, $u_i \cdot u_i = 1$. Therefore, from Eq. (6),

$$\alpha_i^2\|X_r\|^2 + 2\alpha_i\beta_i(X_r \cdot V) + \beta_i^2\|V\|^2 = 1 \tag{8}$$

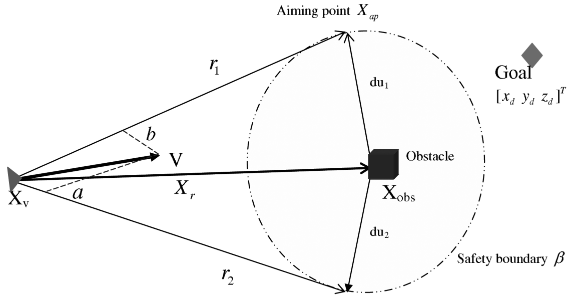

**Fig. 1 Construction and analysis of the collision cone.**

Substituting Eq. (7) in and solving for $\beta_i$

$$\beta_i = \pm\sqrt{\frac{\|X_r\|^2 - d^2}{\|X_r\|^2\|V\|^2 - (X_r \cdot V)^2}} = \pm c \tag{9}$$

Let $\beta_1 = +c$ and $\beta_2 = -c$. Then, from Eq. (6) and (7)

$$u_1 = -\frac{1}{\|X_r^2\|}(c(X_r \cdot V) + d)X_r + cV \tag{10}$$

$$u_2 = \frac{1}{\|X_r^2\|}(c(X_r \cdot V) - d)X_r - cV \tag{11}$$

The vectors $r_i$ can be expressed as

$$r_1 = X_r - \frac{d}{\|X_r^2\|}(c(X_r \cdot V) + d)X_r + cdV \tag{12}$$

$$r_2 = X_r + \frac{d}{\|X_r^2\|}(c(X_r \cdot V) - d)X_r - cdV \tag{13}$$

Adding $r_1$ and $r_2$ and solving for $X_r$ yields $X_r = \frac{1}{2}\frac{X_r^2}{\|X_r\|^2 - d^2}(r_1 + r_2)$. After substituting $X_r$ back in Eq. (12)

$$r_1 = \frac{1}{2}\left(1 - \frac{cd(X_r \cdot V)}{\|X_r\|^2 - d^2}\right)(r_1 + r_2) + cdV \tag{14}$$

$$V = \frac{1}{2}\left(\frac{X_r \cdot V}{\|X_r\|^2 - d^2} + \frac{1}{cd}\right)r_1 + \frac{1}{2}\left(\frac{X_r \cdot V}{\|X_r\|^2 - d^2} - \frac{1}{cd}\right)r_2 \tag{15}$$

Comparing Eq. (1) and (15)

$$a = \frac{1}{2}\left(\frac{X_r \cdot V}{\|X_r\|^2 - d^2} + \frac{1}{cd}\right) \tag{16}$$

$$b = \frac{1}{2}\left(\frac{X_r \cdot V}{\|X_r\|^2 - d^2} - \frac{1}{cd}\right) \tag{17}$$

Two practical issues are expected to arise in the implementation of the collision cone approach:

1) Obstacle safety-bound violation after UAV reached the aiming point: once the UAV is past an aiming point, it maneuvers towards the next aiming point. This may result in a brief violation of the first obstacle's safety bound if the direction of the new aiming point lies in the opposite side of the safety sphere (Fig. 2). To resolve this, a sphere-tracking algorithm is activated when $X_r < d$. The sphere-tracking algorithm computes a new aiming point, called the virtual aiming point. This is calculated as Eq. (19) by radially extending the original relative distance line $X_r$ until it meets the surface of the safety sphere. The UAV then aims for the virtual aiming point until $X_r > d$. There are two solution sets for $X_{\text{vap}}$ depending on the sign of $k$. The solution chosen is the point closer to the vehicle's position $X_v$:

$$k = \pm\frac{d}{\sqrt{(x_v - x_{\text{ob}})^2 + (y_v - y_{\text{ob}})^2 + (z_v - z_{\text{ob}})^2}} \tag{18}$$

$$x_{\text{vap}} = k(x_v - x_{\text{ob}}) + x_{\text{ob}} \qquad z_{\text{vap}} = k(y_v - y_{\text{ob}}) + z_{\text{ob}}$$
$$z_{\text{vap}} = k(z_v - y_{\text{ob}}) + z_{\text{ob}} \tag{19}$$

2) Intersecting obstacle safety boundaries: Fig. 3 illustrates the problem of safety boundary intersection. It is apparent from the vehicle's orientation that obstacle1 is critical, and the UAV must aim towards $X_{p_2}$. However, $X_{p_2}$ is clearly illegal as it lies within the safety

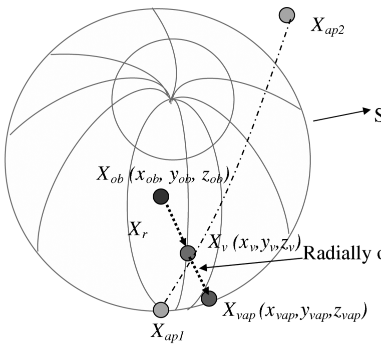

**Fig. 2 Safety boundary violation after aiming point is reached.**

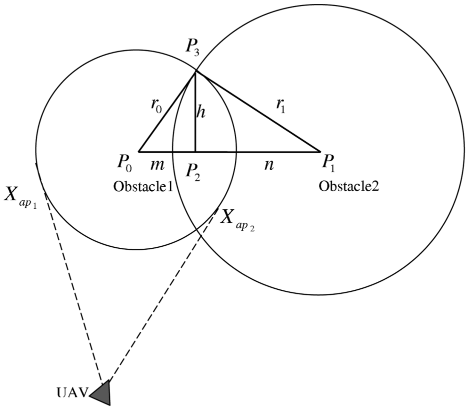

**Fig. 3 Intersection of safety boundaries of two obstacles.**

circle of obstacle 2. This is solved by determining the center $P_2$ of the intersecting area of the two circles:

$$P_2 = P_0 + m(P_1 - P_0)/l \tag{20}$$

where $l = \|P_1 - P_0\|$ and $m = \frac{l^2 + r_0^2 - r_1^2}{2l}$. Then, we choose the aiming point which is farther away from $P_2$, i.e.,

$$X_{\text{ap}} = \begin{cases} X_{p_1}, & \text{if } \|X_{p_1} - P_2\| \geq \|X_{p_2} - P_2\| \\ X_{p_2}, & \text{if } \|X_{p_1} - P_2\| < \|X_{p_2} - P_2\| \end{cases} \tag{21}$$

### B. Collision Detection with Moving Obstacles

The PCA algorithm finds the instant of time $t_c$ at which the two vehicles are closest and the predicted miss distance is $r_m$. The principle of PCA is shown in Fig. 4. If the miss distance $r_m$ is lower than a prespecified safety measure $r_{\text{safe}}$, conflict is said to be detected. The motion of the two UAVs is modeled in Eqs. (22) and (23):

$$Xv_1 = Xv_{1_0} + tV_1 \tag{22}$$

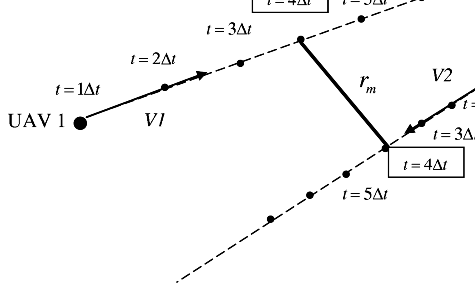

**Fig. 4 PCA method to find the miss distance and the time of closest approach.**

$$Xv_2 = Xv_{2_0} + tV_2 \tag{23}$$

The relative distance between the two UAVs is given by

$$R(t) = X_{v_1}(t) - X_{v_2}(t) \tag{24}$$

$$R(t) = R_0 + t(V_1 - V_2) \tag{25}$$

where $R_0 = Xv_{1_0} - X_{v_{2_0}}$. The point of miss distance occurs when $d(t) = \|R(t)\|$ is minimum, i.e., $D(t) = d(t)^2$ is minimum:

$$D(t) = (V_1 - V_2)\cdot(V_1 - V_2)t^2 + 2R_0\cdot(V_1 - V_2)t + R_0\cdot R_0 \tag{26}$$

For minimum distance $\frac{\mathrm{d}(D(t))}{\mathrm{d}t} = 0$. Therefore, the time instant at which UAV 1 and UAV 2 are closest $t_c$ is given by Eq. (27) and the miss distance is $r_m = X_1(t_c) - X_2(t_c)$:

$$t_c = -\frac{R_0\cdot(V_1 - V_2)}{\|V_1 - V_2\|^2} \tag{27}$$

The conditions required for a potential conflict to occur are:

1) $t_c > 0$: if $t_c$ is negative, the UAVs are moving away from each other.

2) $r_m < r_{\text{safe}}$: the $r_{\text{safe}}$ is the safety distance that must be maintained between the two UAVs. If both conditions (1) and (2) are simultaneously satisfied, a collision will occur at time $t_c$. In an ignorant scenario, the conflict resolution is carried out by only one UAV, assuming complete sharing of information between the UAVs. In a cooperative scenario both the UAVs share the conflict resolution maneuver to avoid collision. The conflict resolution strategy implemented in this Note is vector sharing resolution [5], in which each UAV maneuvers along the miss distance vector until a distance of $r_{\text{safe}}$ is maintained between them. The principle of vector sharing resolution is illustrated in Fig. 5. $V_1$ and $V_2$ are the velocities of UAV 1 and UAV 2, respectively, and $r_m$ is the distance between them. $V_{\text{des}_1}$ and $V_{\text{des}_2}$ are the calculated desired velocities of UAV 1 and UAV 2, respectively, that would ensure a separation of $r_{\text{safe}}$ between them. The distance that must be moved by the UAVs at the PCA is $r_{\text{res}} = r_{\text{safe}} - \|r_m\|$.

Case 1: ignorant scenario: in this case only UAV 1 is assumed to be the maneuverable. The desired velocity $V_{1_{\text{des}}}$ of UAV 1 to resolve this conflict is given by the vector sum of the original position $V_1 t_c$ and the distance $r_{vs_1}$ that results in a safety distance of $r_{\text{safe}}$ between the UAVs. Since we require knowledge of the alternate direction for safe maneuver, we consider the unit vector along the resultant:

$$V_{1_{\text{des}}} = (V_1 t_c + r_{vs_1})/\|V_1 t_c + r_{vs_1}\| \tag{28}$$

where $r_{vs_1}$ is the distance $r_{\text{res}}$ in the direction of $-r_m$. This causes the UAV 1 to move away from UAV 2 by a distance of $r_{\text{res}}$:

$$r_{vs_1} = \frac{r_{\text{res}}(-r_m)}{r_m} \tag{29}$$

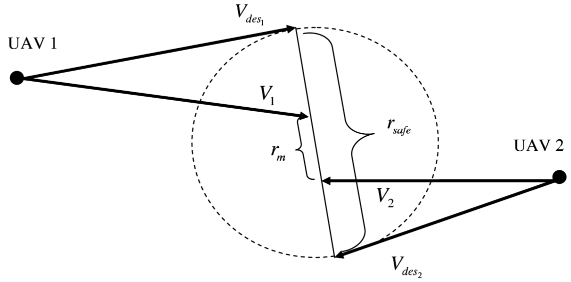

**Fig. 5 Cooperative conflict resolution using vector sharing resolution.**

Case 2: cooperative scenario: in a cooperative scenario, both UAV 1 and UAV 2 share the conflict resolving maneuver in following way:

$$r_{vs_1} = \frac{\|V_2\|}{\|V_1\| + \|V_2\|} \frac{-r_m}{|r_m|} r_{\text{res}} \tag{30}$$

$$r_{vs_2} = \frac{\|V_1\|}{\|V_1\| + \|V_2\|} \frac{r_m}{|r_m|} r_{\text{res}} \tag{31}$$

The desired velocities of UAV 1 and UAV 2 are

$$V_{1_{\text{des}}} = \frac{V_1 t_c + r_{vs_1}}{\|V_1 t_c + r_{vs_1}\|} \tag{32}$$

$$V_{2_{\text{des}}} = \frac{V_2 t_c + r_{vs_2}}{\|V_2 t_c + r_{vs_2}\|} \tag{33}$$

Note that the geometry for guidance is the same as that of stationary obstacles (see Fig. 6), except that the aiming line now represents the desired velocity vector.

### C. Geometry of the Guidance Problem

The guidance objective is to align the velocity vector to the aiming line so the collision is avoided. The problem geometry is shown in Fig. 6. $u$, $v$ and $w$ are the velocity components along the $x$, $y$ and $z$ directions, respectively, and $a_y = \dot{v}$, $a_z = \dot{w}$ are the accelerations applied in the $y$ and $z$ directions, respectively. Note that the velocity $u$ is assumed to be a constant.

The angle $\theta$ between the total velocity vector and the aiming line is to be eliminated. To formulate a guidance that computes the controls $a_y$ and $a_z$ that would eliminate $\theta$, we consider the three-dimensional problem as a combination of two separate two-dimensional problems in the $XY$ and $XZ$ planes. $(X_{\text{ap}})_{XY}$ and $(X_{\text{ap}})_{XZ}$ are the projections of the aiming line on to the $XY$ and $XZ$ planes, respectively, while $V_{XY}$ and $V_{XZ}$ are projections of the velocity vector. $\theta_y$ is the angle between $(X_{\text{ap}})_{XY}$ and $V_{XY}$ and $\theta_z$ is the angle between $(X_{\text{ap}})_{XZ}$ and $V_{XZ}$. The guidance objective is to compute the controls $a_y$ and $a_z$ to eliminate $\theta_y$ and $\theta_z$ respectively in the two independent $XY$ and $XZ$ planes.

### D. Two-Dimensional Decomposition of the Three-Dimensional Guidance Problem

Figure 7 shows the geometry of the guidance problem in the $XY$ plane. The geometry of the problem in $XZ$ plane is the same.

When the velocity vector $V_{XY}$ is perfectly aligned with the aiming line $X_{\text{ap}}$, the $y$-component of the velocity vector changes to a new value, $v^*$ ($w^*$ in the $XZ$ plane). $v^*$ is found by the intersection of the lines $l_1$ and $l_2$. From Fig. 7

$$l_1\colon\ x = u \tag{34}$$

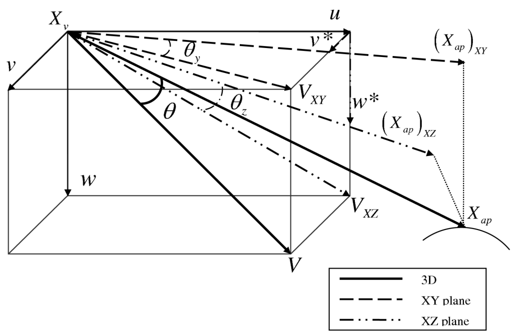

Fig. 6 Geometry of the guidance problem in three-dimensional.

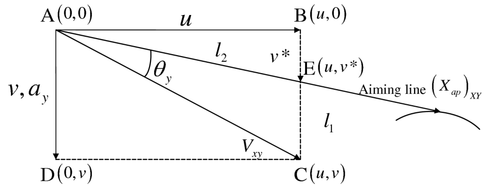

Fig. 7 Geometry of the guidance problem in two-dimensional.

$$l_2\colon\ \frac{y}{y_{\text{ap}}} = \frac{x}{x_{\text{ap}}} \tag{35}$$

$$v^* = \left(\frac{y_{\text{ap}}}{x_{\text{ap}}}\right) u \tag{36}$$

## III. Geometric and DGG for Collision Avoidance

### A. LGG

The velocity vector $V$ must be aligned with the aiming line $X_{\text{ap}}$. From Fig. 7, it is seen that this amounts to taking the aiming angle $\theta$ to zero in time $t_{\text{go}}$. One way to achieve this is by using a purely Geometric Guidance, described as the APG [4].

The APG law aims to achieve it by explicitly designing the accelerations $a_y$ and $a_z$, linearly proportional to the angles $\theta_y$ and $\theta_z$, with proper dimensions for the gains $\hat{k}_v$ and $\hat{k}_w$ respectively:

$$\begin{bmatrix} a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \hat{k}_v \theta_y \\ \hat{k}_w \theta_z \end{bmatrix} \tag{37}$$

Such a guidance law results in a high turning rate at the beginning and quickly settles along the aiming line, instead of maneuvering all the way to impact. Such a maneuver is an advantage when compared with a conventional proportional navigation guidance law. As discussed in the APG law works very well when accurate predictions of $t_{\text{go}}$ and $X_{\text{ap}}$ are possible. Here $X_{\text{ap}}$ and $t_{\text{go}}$ are found accurately using the collision cone approach. For stationary targets $t_{\text{go}}$ prediction is accurate. It is possible to extend this to moving obstacles via simple prediction algorithms. Only in the case of a highly maneuverable target, a high precision prediction algorithm is needed to find the terminal conditions. However, since UAV flight occurs in urban terrain, the targets (or obstacles) are in general not highly maneuverable.

### B. Nonlinear Geometric Guidance

We propose a new NGG law as follows:

$$\begin{bmatrix} a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \hat{k}_v \sin\theta_y \\ \hat{k}_w \sin\theta_z \end{bmatrix} \tag{38}$$

The control is a nonlinear function of the aiming angle $\theta$. The immediate advantage is that the acceleration in NGG is always bounded since the range of the sine function is $[-1, 1]$, provided $\hat{k}_v$ is bounded. Even if we bound $\theta$ to a realistic range of $[-\pi, \pi]$, the NGG control will only be one third as high as the LGG control for the same value of $\hat{k}_v$.

### C. Nonlinear DGG

We design a guidance law based on dynamic inversion (DI) [6], a control strategy used for output tracking of nonlinear systems. The principle of DI is to drive a stabilizing error dynamics (chosen by the designer) to zero. The main advantage of DI is that it essentially guarantees global asymptotic stability with respect to the tracking error. We now describe the DI-based guidance design. Let the error be

$$e_v = v - v^* \tag{39}$$

Imposing the first order error dynamics

$$\dot{e}_v + k_v e_v = 0 \tag{40}$$

$$(\dot{v} - \dot{v}^*) + k_v(v - v^*) = 0 \tag{41}$$

With quasi-steady approximation (i.e. assuming $v^*$ is a constant at every instant of time), $v^* = 0$. In addition, $\dot{v} = a_y$ from the system dynamics. From Eq. (41), the DI-based guidance law can be derived as

$$a_y = -k_v(v - v^*) \tag{42}$$

The constant $k_v$ is designed such that the settling time (i.e. the time taken to align the velocity vector with the aiming line) is inversely proportional to the time-to-go, i.e. $k_v = \frac{1}{\tau_v}$ where $\tau_v$ is the desired time constant of the error dynamics. Note that $\tau_v$ can be selected by choosing an appropriate settling time $T_{s_v}$, since for linear systems theory, $\tau_v = \frac{T_{s_v}}{4}$.

Furthermore, one can choose $T_{s_v}$ as $T_{s_v} = \alpha t_{\mathrm{go}}$, $0 < \alpha < 1$. Such a guidance strategy ensures that a larger control is generated for an obstacle that is nearer (i.e. the time-to-go is smaller). The guidance strategy in Eq. (42) is proportional to the error in the $y$-velocity, and thus produces a large control input at the beginning which effects quick settlement along the aiming line. Two factors influence the control $a_y$:

1) $t_{\mathrm{go}}$: the gain $k_v$ is designed to be inversely proportional to $t_{\mathrm{go}}$ so that larger control is demanded for obstacles that are closer.

2) $\alpha$: the choice of $\alpha$ determines the speed of settling of the control. If the value of $\alpha$ is close to one, the settling is slow and consequently, the peak in control is low. However, if fast settling is desired, a low value of $\alpha$ will effect it. However, faster settling results in a higher peak in control.

If both $t_{\mathrm{go}}$ and $\alpha$ are small, the peak in control may become impractically large and results in control saturation. Therefore, the value of $\alpha$ must be chosen judiciously based on the requirement of speed of alignment. Since reactive collision avoidance requires quick maneuvering, a value of $\alpha$ between 0.3 and 0.8 is suitable. Slower maneuvers with $\alpha > 0.8$ may be suitable for flying to the destination. In an analogous manner, the expression for $a_z$ in $XZ$ plane can be derived.

The stability of the DI-based guidance strategy is examined with respect to the error. The error is defined as

$$\begin{bmatrix} e_v \\ e_w \end{bmatrix} = \begin{bmatrix} v - v^* \\ w - w^* \end{bmatrix} \tag{43}$$

The error dynamics are described as follows:

$$\begin{bmatrix} \dot{e}_v \\ \dot{e}_w \end{bmatrix} = \begin{bmatrix} -k_v & 0 \\ 0 & -k_w \end{bmatrix} \begin{bmatrix} e_v \\ e_w \end{bmatrix} \tag{44}$$

Equation (44) constitutes a linear time-varying system. Consider a linear time-varying system of the form $\dot{x} = A(t)x$. The system is determined to be asymptotically stable if the symmetric matrix $A(t) + A^T(t)$ has all eigenvalues lying the left half of the complex plane. In the system in Eq. (44), the eigenvalues of $A(t) + A^T(t)$ are $-2k_v$ and $-2k_w$. Since $k_v$ and $k_w$ are inversely proportional to the $t_{\mathrm{go}}$, they are strictly positive. Therefore, both the eigenvalues lie strictly on the left-half of the complex plane and the system in Eq. (44) is asymptotically stable.

## IV. Correlation of LGG, NGG, and DGG

*Theorem 1*: the LGG is approximately equivalent to DGG with $\hat{k}_v = \sqrt{2}k_v(\sqrt[4]{(u^2 + v^2)(u^2 + v^{*2})})$

*Proof*: the DGG law is

$$a_y = -k_v(v - v^*) \tag{45}$$

The LGG law is

$$a_y = -\hat{k}_v\theta_y \tag{46}$$

Dividing Eq. (45) by Eq. (46)

$$\frac{\hat{k}_v}{k_v} = \frac{v - v^*}{\theta_y} \tag{47}$$

From Fig. 7, $\theta_y$ is found as follows:

$$\theta_y = \cos^{-1}\left(\frac{AC \cdot AE}{\|AC\|\|AE\|}\right) \tag{48}$$

$$\cos\theta_y = \frac{u^2 + vv^*}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}} \tag{49}$$

The Taylor series approximation for $cos\theta_y$ is $\cos\theta_y = 1 - \frac{\theta_y^2}{2!} + \frac{\theta_y^4}{4!} - \ldots$. After neglecting powers of $\theta_y$ greater than or equal to four:

$$\cos\theta_y = 1 - \frac{\theta_y^2}{2} = \frac{u^2 + vv^*}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}} \tag{50}$$

$$\theta_y = \sqrt{2}\left(\frac{u^2\sqrt{1 + \frac{v^2}{u^2}}\sqrt{1 + \frac{v^{*2}}{u^2}} - (u^2 + vv^*)}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}}\right)^{\frac{1}{2}} \tag{51}$$

Using the binomial approximation $\sqrt{1 + x} = 1 + \frac{x}{2}$,

$$\theta_y = \sqrt{2}\left(\frac{u^2(1 + \frac{v^2}{2u^2})(1 + \frac{v^{*2}}{2u^2}) - (u^2 + vv^*)}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}}\right)^{\frac{1}{2}} \tag{52}$$

Neglecting the higher order term $\frac{v^2v^{*2}}{4u^4}$ and simplifying, we get

$$\theta_y = \frac{1}{\sqrt{2}}\left(\frac{(v - v^*)}{\sqrt{[4](u^2 + v^2)(u^2 + v^{*2})}}\right) \tag{53}$$

Substituting this in the LGG guidance law [Eq. (46)]

$$a_y = -\hat{k}_v\theta_y = \frac{-\hat{k}_v}{\sqrt{2}}\left(\frac{(v - v^*)}{\sqrt{[4](u^2 + v^2)(u^2 + v^{*2})}}\right) \tag{54}$$

Equating Eq. (54) with Eq. (45)

$$\hat{k}_v = \sqrt{2}k_v(\sqrt[4]{(u^2 + v^2)(u^2 + v^{*2})}) \tag{55}$$

Therefore, we see that with relevant approximations, the control in the LGG becomes equal to the control in DGG with the preceding derived appropriate gain selection. □

*Theorem 2*: the LGG is approximately equivalent to DGG with $\hat{k}_v = k_v\left(\frac{\sqrt{u^2+v^2}\sqrt{u^2+v^{*2}}}{u}\right)$

*Proof*: with relation $\sin\theta_y = \sqrt{1 - \cos^2\theta_y}$. From Eq. (49) we have

$$\sin\theta_y = \left(\frac{u}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}}\right)(v - v^*) \tag{56}$$

With approximation $\sin\theta_y \approx \theta_y$

$$\theta_y = \left(\frac{u}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}}\right)(v - v^*) \tag{57}$$

Substituting this in the LGG law [Eq. (46)]

$$a_y = -\hat{k}_v\left(\frac{u}{\sqrt{u^2 + v^2}\sqrt{u^2 + v^{*2}}}\right)(v - v^*) \tag{58}$$

Equating Eq. (58) with the DGG law [Eq. (45)]

$$\hat{k}_v = k_v\left(\frac{\sqrt{u^2+v^2}\sqrt{u^2+v^{*2}}}{u}\right) \tag{59}$$

□

*Theorem 3*: the NGG is equivalent to DGG with $\hat{k}_v = k_v(\frac{\sqrt{u^2+v^2}\sqrt{u^2+v^{*2}}}{u})$

*Proof*: instead of making the approximation $\sin\theta \approx \theta$ in Eq. (57), we represent the guidance directly as

$$a_y = \hat{k}_v \sin\theta_y \tag{60}$$

This is the NGG. Note that the derivation of this correlation involves no approximations. □

Thus, the NGG is directly correlated to the DGG, and the LGG is an approximation of the DGG when the gains are selected appropriately. Therefore, the NGG will possess the advantages of DGG discussed in Sec. III.C. Additionally, the LGG approximations in both Case 1 and Case 2 hold true only while $|\theta|$ is small. If $|\theta|$ becomes large, the approximation $\sin\theta \approx \theta$ no longer holds true. The NGG will still give precise values, while the control in LGG will be inaccurate.

## V. Numerical Results

### A. Collision Avoidance with Stationary Obstacles

The experiments with stationary obstacles involve a finite space with two obstacles in various positions, and the DGG, LGG and NGG algorithms are used to find a collision-free path to the destination. The gains $k_v$ and $\hat{k}_v$ are chosen in accordance with the correlation proposed in Sec. IV and hence the results of both DGG and NGG are found to be same.

*1. Collision Avoidance with Ideal Autopilot*

Figure 8 shows the path found by DGG, LGG and NGG in three dimensions when both the obstacles are critical. The safety boundaries of the obstacles are spheres. The plot of the accelerations $a_y$ and $a_z$ are shown in Fig. 9. Each peak in the control represents each change in the aiming point.

The sphere-tracking algorithm is invoked in cases where an obstacle's safety boundary is violated due to the positioning of the aiming points (discussed in section). Figure 10 shows the controls $a_y$ and $a_z$. The sphere-tracking algorithm effects the control to align the velocity vector toward the next aiming point. Figure 11 shows the trajectory of the UAV.

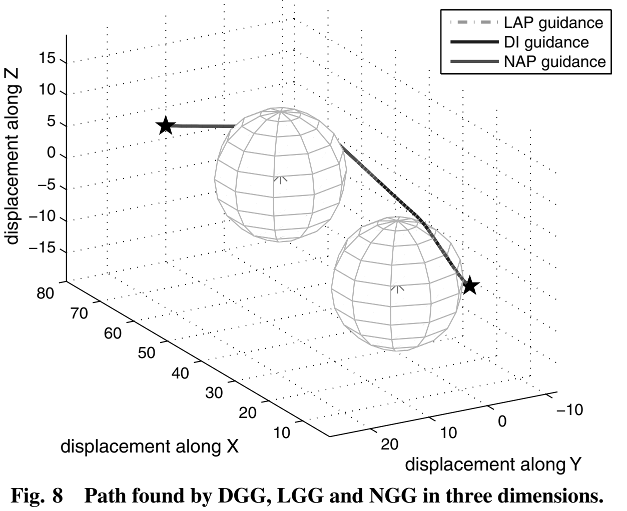

Fig. 8 Path found by DGG, LGG and NGG in three dimensions.

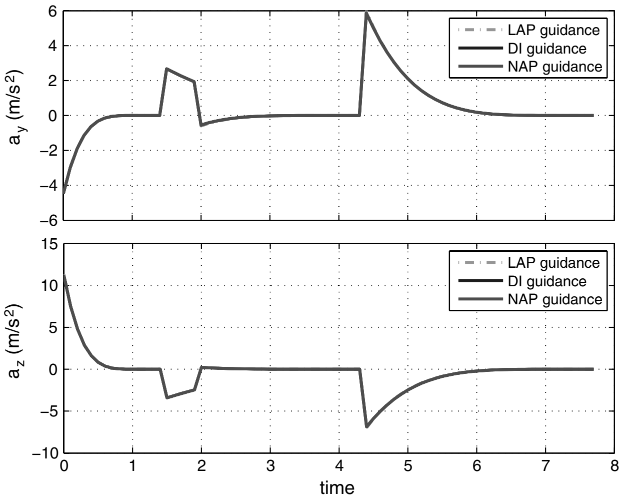

Fig. 9 Control for DGG, LGG and NGG for three-dimensional case.

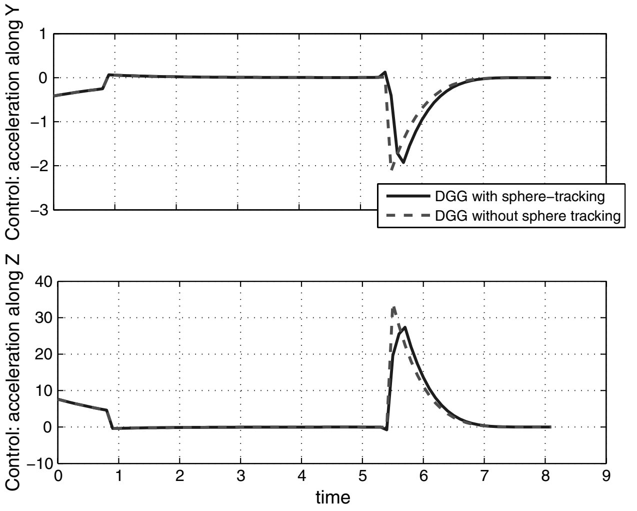

Fig. 10 DGG Control in *y*- and *z*- with and without sphere-tracking algorithm.

*2. Collision Avoidance with Autopilot Lag*

The guidance laws are also tested with an autopilot lag of 0.2 s. The results of these simulation studies for the DGG are shown in Figs. 12 and 13. Figure 13 shows the comparison between the controls with and without autopilot delay. In addition to the autopilot lag, a control saturation of 20 $\mathrm{m/s^2}$ is also imposed, since practical vehicles cannot

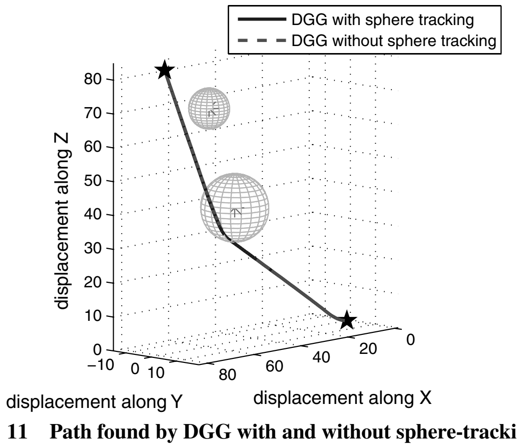

Fig. 11 Path found by DGG with and without sphere-tracking.

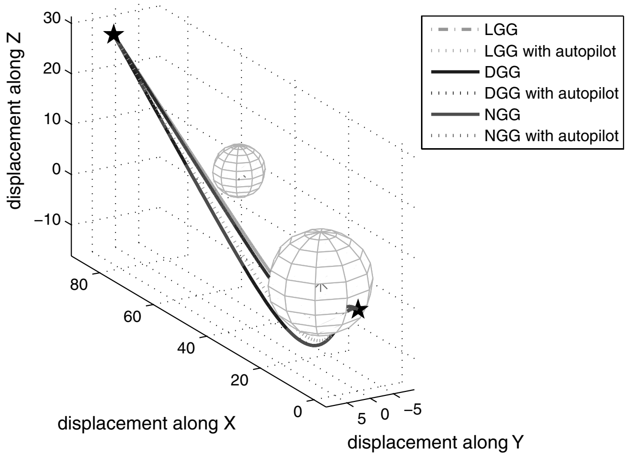

**Fig. 12 Path found by DGG, LGG and NGG with and without autopilot lag; lag = 0.2 s.**

carry out very large maneuvers. Even though the results are satisfactory in this case, for higher autopilot lags large deviations from the original trajectory are observed. Hence an autopilot compensation loop is proposed next to improve the performance in those cases.

### *3. Collision Avoidance with Autopilot Lag Compensation*

To improve the performance in higher autopilot lag cases, we incorporate a lag compensation strategy that modifies the demanded control such that the generated control approaches the desired value. The autopilot lag is modeled as before:

$$\dot{U} = \frac{1}{\tau}(U - U_c) \tag{61}$$

where $U = [a_y; a_z]$ represents the realized control, and $U_c$ is the commanded value of control. Our objective is $U \to U^*$, where $U^*$ is the desired control, computed by the guidance laws. This gives us the error in control $\Delta U = U - U^*$. Imposing the first order error dynamics we have

$$\Delta\dot{U} + k_a \Delta U = 0 \tag{62}$$

$$\text{i .e., } (\dot{U} - \dot{U}^*) + k_a(U - U^*) = 0 \tag{63}$$

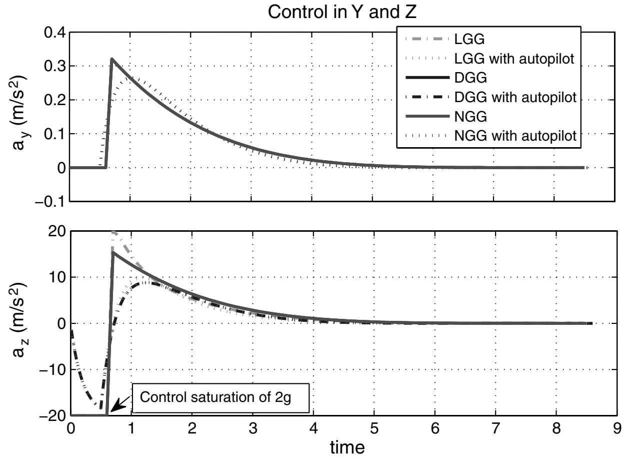

**Fig. 13 Control for DGG, LGG and NGG with and without autopilot lag of 0.2 s case.**

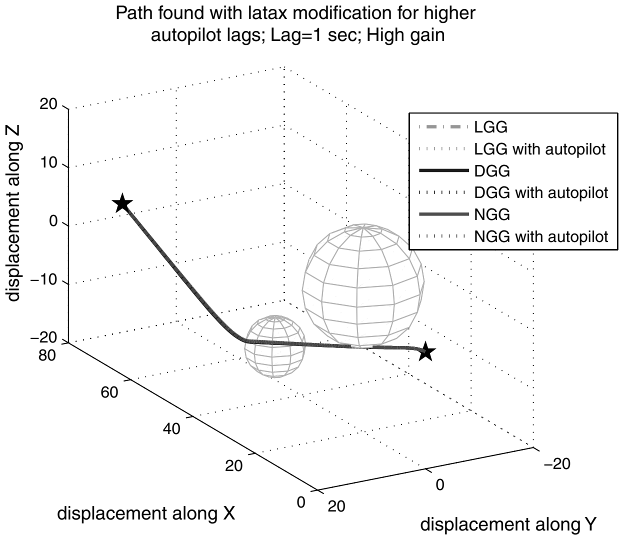

**Fig. 14 Path found by DGG, LGG and NGG without and with autopilot lag (with lateral acceleration modification); lag = 1 s.**

Substituting from Eq. (61),

$$\left(\frac{U_c - U}{\tau} - \dot{U}^*\right) + k_a(U - U^*) = 0 \tag{64}$$

The desired rate of change of control $\dot{U}^* = 0$. Therefore,

$$U_c = -k_a \tau (U - U^*) + U \tag{65}$$

This yields the expression for the control to be commanded such that the control actually generated by the system after autopilot lag is $U^*$. $k_a$ is the gain to be selected appropriately by the designer and $\tau$ is the autopilot lag of the system. As in Sec. III.C, we design the gain $k_a = \frac{4}{T_s}$. The smaller the time allowed for settling, the better is the performance of the autopilot compensation system in terms of tracking the desired acceleration.

Figures 14 and 15 give the trajectory and controls with autopilot compensation. The chosen gain is $k_a$ is for a settling time of 0.0025 s. Such a step results in almost complete tracking of the autopilot values with the desired values, with the maximum deviation from the original trajectory being 0.14 m. For higher autopilot lags, therefore, it is recommended to apply autopilot lag compensation for good performance of the guidance.

## B. Collision Avoidance with Moving Obstacles

The simulations were carried out in both ignorant as well as cooperative scenarios. In ignorant conflict resolution, maneuver is

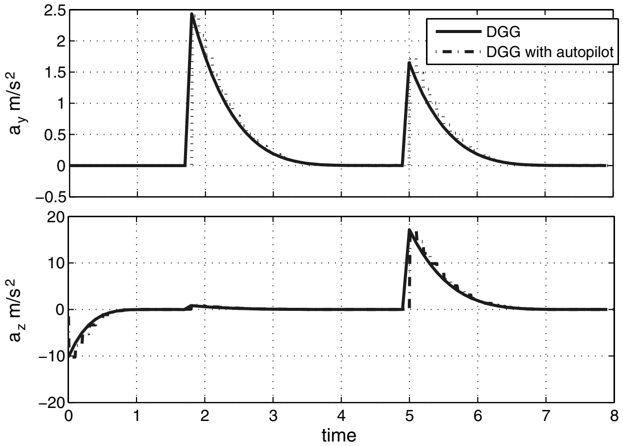

**Fig. 15 Control for DGG, LGG and NGG without and with autopilot lag (with lateral acceleration modification); lag = 1 s.**

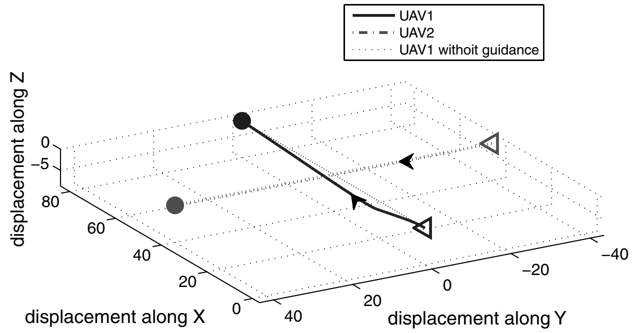

**Fig. 16 Path found by DGG for avoiding collision with a nonmaneuvering obstacle in a noncooperative scenario.**

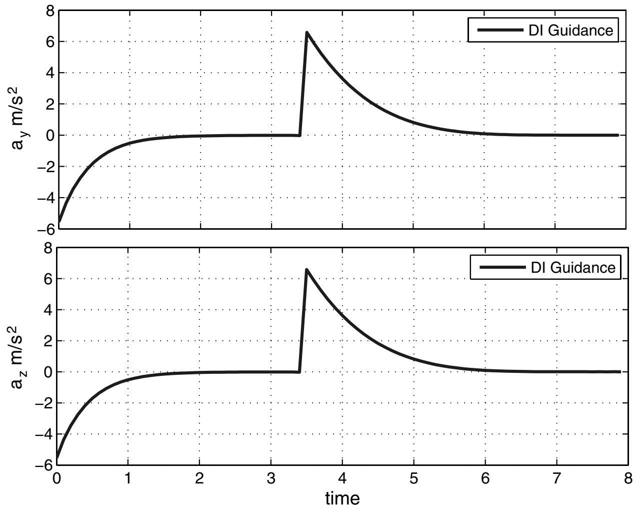

**Fig. 17 Control in *Y* and *Z* directions for noncooperative collision avoidance with a moving obstacle.**

carried out by the maneuverable UAV only (i.e., UAV 1). The obstacle UAV (i.e., UAV 2) is considered to have a constant velocity and no acceleration. This scenario can also be interpreted as a "ignorant scenario" as the UAV 2 is assumed to be ignorant about the motion of UAV 1. The initial and destination points of UAV 1 are [0,0,0] and [80,0,0], respectively, while the initial and destination points of UAV 2 are $[40, -40, 0]$ and [40,40,0], respectively. The velocities are $V_1 = [10, 0, 0]$ and $V_2 = [0, 10, 0]$. These conditions are chosen such that the trajectories of the two UAVs will intersect at [40,0,0] without guidance. Figure 16 shows the trajectories of UAV 1 and UAV 2 with and without guidance. The controls $a_y$ and $a_z$ are shown in Fig. 17. Results are equally promising in cooperative conflict resolution scenarios. However, results are omitted here for the sake of limited space.

## VI. Conclusions

Using the philosophy of "collision cone" and "DI," two novel guidance strategies, named as NGG and DGG, are proposed in this Note for reactive collision avoidance of UAVs. Appropriate correlation among them has also been derived. For additional safety, a sphere-tracking algorithm is also proposed in this Note which guides the vehicle along the surface of an obstacle's safety sphere when a brief violation occurs after reaching the aiming point on its surface. Incorporating the philosophy of "PCA," the guidance laws have also been made applicable for moving obstacles as well. The numerical simulation results are quite promising. To be more realistic, simulation studies have also been carried out in presence of autopilot lags and, if the lag is large, an innovative lag compensation logic has also been proposed. The numerical simulation results are quite promising.

Even though the main aim of this Note is collision avoidance of UAVs, the core idea is inspired from the missile guidance literature. Hence, other potential applications of the proposed guidance laws would include guidance of missiles. In such applications, the predicted target position can serve as the aiming point towards which the velocity vector of the missile should be aligned.

## Acknowledgment

This research was supported by the U.S. Air Force Office of Scientific Research, Air Force Research Laboratories, USA, under the contract number FA-23860814102, which was operated at the Indian Institute of Science, Bangalore, India with the project code SID/PC99148.

## References

[1] Amin, J. N., Boskovic, J. D., and Mehra, R. K., "A Fast and Efficient Approach to Path Planning for Unmanned Vehicles," AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA Paper 2006-6103, Keystone, CO, Aug. 2006.

[2] Hwangbo, M., Kuffner, J., and Kanade, T., "Efficient Two-Phase 3D Motion Planning for Small Fixed-Wing UAVs," IEEE International Conference on Robotics and Automation, Rome, 10–14 April 2007.

[3] Watanabe, Y., Calise, A. J., and Johnson, E. N., "Minimum Effort Guidance for Vision-Based Collision Avoidance," AIAA Atmospheric Flight Mechanics Conference and Exhibit, AIAA Paper 2006-6641, Keystone, CO, 21–24 Aug. 2006.

[4] Lu-Ping, Tsao, Ching-Lain, Chou, Chuen-Ming, Chen, and Chi-Teh, Chen, "Aiming Point Guidance Law for Air-to-Air Missiles," *International Journal of Systems Science*, Vol. 29, No. 2, 1998, pp. 95–102.
doi:10.1080/00207729808929501

[5] Park, J. W., Oh, H. D., and Tahk, M. J., "UAV Collision Avoidance Based on Geometric Approach," *SICE Annual Conference*, 2008, pp. 2122–2126.

[6] Enns, D., Bugajski, D., Hendrick, R., and Stein, G., "Dynamic Inversion: An Evolving Methodology for Flight Control Design," *International Journal of Control*, Vol. 59, No. 1, 1994, pp. 71–91.
doi:10.1080/00207179408923070