Reactive collision avoidance of multiple realistic UAVs

Joel George Manathara and Debasish Ghose
Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

Abstract

Purpose – Unmanned aerial vehicles (UAVs) have a wide variety of applications such as surveillance and search. Many of these tasks are better executed by multiple UAVs acting as a group. One of the main problems to be tackled in a high-density UAV traffic scenario is that of collision avoidance among UAVs. The purpose of this paper is to give a collision avoidance algorithm to detect and resolve the conflicts of projected path among UAVs.
Design/methodology/approach – The collision avoidance algorithm developed in the paper handles multiple UAV conflicts by considering only the most imminent predicted collision and doing a maneuver to increase the line-of-sight rate to avoid that conflict. After the collision avoidance maneuver, the UAVs fly to their destinations via Dubins shortest path to minimize time to reach destination. The algorithm is tested on realistic six degree of freedom UAV models augmented with proportional-integral controllers to hold altitude, velocity, and commanded bank angles.
Findings – The paper shows, through extensive simulations, that the proposed collision avoidance algorithm gives a good performance in high-density UAV traffic scenarios. The proposed collision avoidance algorithm is simple to implement and is computationally efficient.
Practical implications – The algorithm developed in this paper can be easily implemented on actual UAVs.
Originality/value – There are only a few works in the literature that address multiple UAV collision avoidance in very high-density traffic situations. This paper addresses very high-density multiple UAV conflict resolution with realistic UAV models.

Keywords Multiple unmanned aerial vehicles, Collision avoidance, Six degree of freedom aircraft model, Surveillance, Collisions

Paper type Research paper

Introduction

Owing to its portability, low cost of acquisition, operation, and maintenance, and absence to human risk, the unmanned aerial vehicles (UAVs) find a wide variety of applications. Some missions, like oil pipeline monitoring, are monotonous jobs for humans and they indeed do not require human presence. There are other missions, like nuclear or chemical leakage inspection, that prohibits the presence of humans due to safety reasons. Thus, UAVs find great use in a wide variety of dull, dirty, and dangerous missions. The projected number of UAVs that will be in use in a decade from now is enormous (US Department of Defense, 2005) and many of the tasks will be executed by multiple UAVs acting as a group, coordinating and cooperating with each other. This will lead to a congestion of airspace occupied by UAVs. One of the main problems to be tackled in a high-density UAV traffic scenario is that of collision avoidance. UAVs occupying the same airspace need to avoid each other to ensure a safe completion of respective missions. This paper proposes a computationally efficient and easy to implement collision avoidance algorithm to detect and resolve conflicts among UAVs and evaluate its performance on realistic models of UAVs.

The following problem is considered in this paper. Several UAVs are assumed to fly from different bases to their respective destinations. These UAVs need to avoid mid-air collision with other UAVs on their path. For safety reasons, "miss distance" or the minimum separation between any two UAVs at any time should be greater than a specified value. Any location of two UAVs within the minimum required separation results in a near miss. The objective is to find an algorithm that, when executed by every UAV, results in no collisions and minimum number of "near misses". Although it is desirable to have zero near misses, it might be impossible to achieve this in high-density air traffic scenarios like the ones considered in this paper. Moreover, in the case of UAVs, where there is no human risk, such a high level of safety requirement may not be necessary. Hence the emphasis is on reducing near misses rather than completely eliminating them, in order to achieve better efficiency. The collision avoidance maneuvers have to be efficient in the sense that the deviation of a UAV from its nominal path due to collision avoidance maneuver should be minimal in order to minimize late arrival at its destination. Each UAV needs to find a safe path to its destination when limited information of positions and velocities, of only the neighboring UAVs that are within its sensor range, is available.

Collision avoidance has been an active area of research in the field of robotics. The prominent ones are the dynamic window approach (Fox *et al.*, 1997), the collision cone method (Chakravarthy and Ghose, 1998), the velocity obstacle method (Fiorini and Shiller, 1998; Shiller *et al.*, 2001), and the inevitable collision states approach (Fraichard and Asama, 2004; Gomez and Fraichard, 2009). These methods are developed with robotic application in mind. The collision avoidance techniques employed widely in robotics, as mentioned above, are typically computationally very intensive. This is acceptable for robots that can stop to do the calculations, but not desirable for fixed wing UAVs that require a minimum forward velocity to keep afloat in air. Since these algorithms are designed for robots, it is assumed that the vehicles have braking forces and can stop if required

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1748-8842.htm

Aircraft Engineering and Aerospace Technology: An International Journal
83/6 (2011) 388–396

[DOI 10.1108/00022661111173261]

and can change their direction of motion instantaneously. Although these algorithms can be generalized for multi-vehicle collision avoidance, they are originally designed for one robot navigating through an environment with obstacles. Owing to reasons described above, the algorithms developed for robotic applications cannot be directly applied to UAV problems.

There has been some research over the past decade on aircraft collision avoidance both from the multiple UAV and the air traffic control points of view. Extensive reviews of the collision avoidance algorithms are given from the air traffic control point of view by Kuchar and Yang (2000) and Albaker and Rahim (2009a) from the UAV point of view.

Most of the algorithms developed for air traffic management are those that guarantee safe trajectories in a very low-density traffic involving only two or three aircraft (Tomlin *et al.*, 1998; Frazzoli *et al.*, 2005; Bicchi and Pallottino, 2000). Also, these algorithms require instantaneous changes in position and heading of aircraft which is unrealistic, and they are not applicable in highly dynamic environments considered in this paper.

There are certain algorithms that consider aircraft as agents and deconfliction is achieved by peer-to-peer negotiation (Sislak *et al.*, 2006, 2007; Albaker and Rahim, 2009b). The aircraft share their flight plans, and modify them based on negotiation to find safe trajectories. Such an approach can be used for air traffic management application as well as for UAV collision avoidance. Negotiation requires communication between aircraft. Therefore, these algorithms cannot be used in adversary environments where communication may not be possible. Samek *et al.* (2007) use a multi-party negotiator instead of a peer-to-peer negotiation. Simulations they present involve high-density traffic. The communication and computation overhead associated with this collision avoidance algorithm is high. Moreover, they do not consider realistic aircraft models.

Another approach to collision avoidance is artificial potential-based methods where aircraft or UAVs are treated like "charged particles" of same charge that repel each other; the destination of an aircraft is modeled as a charge of the opposite sign so as to attract or navigate the aircraft toward the destination. The artificial potential methods are susceptible to local minima and require breaking forces, and therefore are not widely used in UAV collision avoidance. However, several variants of potential-based methods have been used for aircraft and UAV collision avoidance (Eby and Kelly, 1999; Roussos *et al.*, 2010; Shim *et al.*, 2003; Rahmani *et al.*, 2008; Panyakeow and Mesbahi, 2010). The potential methods, typically, use only the position information. Ignoring the velocity information can lead to highly conservative solutions that are inefficient. Potential function-based methods are also computationally intensive.

Chakravarthy and Ghose (1998) developed the concept of "collision cone" that can be used to determine whether two objects, of irregular shapes and arbitrary sizes, are on a collision course. The collision cone approach has been the basis for certain collision/obstacle avoidance algorithms (Lalish *et al.*, 2008; Smith and Harmon, 2010). The algorithm in Lalish *et al.* (2008) is not decentralized as a UAV implementing this algorithm requires information of all other UAVs. This may, in general, be not possible and might lead to intensive computational requirements in high-density traffic environments. Also, the "guaranteed" deconfliction can at times be sacrificed for efficiency; an effort to strictly stick to assured separation at times can lead to very inefficient trajectories. Such a situation can occur more frequently in highly cluttered environments which they do not consider. Smith and Harmon (2010) apply their algorithm to scenarios involving two or three UAVs only.

Archibald *et al.* (2008) use satisficing game theory to address the problem of collision avoidance in multiple UAVs. They present simulations involving test cases with high-density air traffic scenarios to test the proposed algorithm. This algorithm, apart from being computationally intensive, requires constant communication between neighboring UAVs which may not always be possible, and considers only kinematic aircraft model.

The literature on collision avoidance among UAVs which consider a realistic model for UAVs is scarce. A notable contribution by Shim *et al.* (2003) employs a potential function method along with nonlinear model predictive control to achieve collision avoidance in a three-dimensional space. They demonstrate, in simulation, collision avoidance between five helicopters which are modeled using six degrees of freedom equations of motion and realistic control inputs. In contrast, the present paper simulates scenarios of 20-80 UAVs.

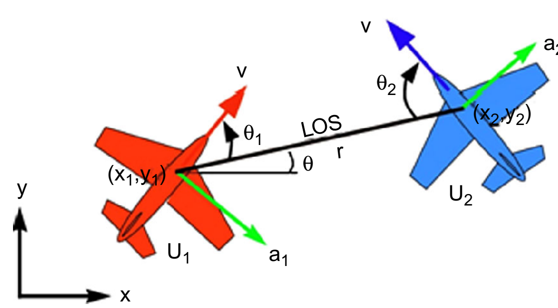
Collision avoidance algorithm

The proposed multiple UAV collision avoidance algorithm consists of applying lateral accelerations to increase the line of sight (LOS) rate between UAVs that are on a collision course. The algorithm is computationally efficient and easy to implement. A preliminary version of this algorithm is given in George and Ghose (2009), where only kinematic models of UAVs are considered.

When a UAV detects multiple conflicts, it selects from among its neighbors, with which the predicted zero effort misses (ZEMs) (miss distance that will result if both the UAVs continue in their current course) are less than the minimum desired separation, the one with which collision is imminent, and pulls out of the collision course with it. This is done by applying a lateral acceleration in a way that will increase the rate of rotation of the LOS connecting the two UAVs (Figure 1). To increase the LOS rate in this scenario ($\theta_2 > \theta_1$), accelerations a_1 and a_2, which makes the UAVs execute the tightest turn, is applied to U_1 and U_2, respectively.

If a UAV is not on a collision course with any other UAV, then it flies to its destination. It is preferable to do this in minimum time. A UAV taking a Dubins (1957) path from the current location to the destination achieves this objective.

Figure 1 Collision avoidance rule for 2D engagement



The collision avoidance algorithm developed in this paper handles the multiple UAV scenarios by considering only that predicted collision which is imminent to a UAV in terms of the minimum LOS rate. This approach results in a good collision avoidance algorithm with good performance as will be demonstrated through simulations. The algorithm can also handle heterogeneous group of UAVs with different velocities and radii of turn by using tuning parameters specific to individual UAVs instead of globally defined parameters. However, to demonstrate the efficacy of the algorithm in a realistic scenario, only a homogeneous group of UAVs are considered for simulations in this paper. The algorithm for multiple UAV collision avoidance is given in Algorithm 1.

Using the kinematic UAV model, it is possible to attain instantaneously the turn rates commanded by the collision avoidance algorithm. However, this may not be possible in case of a realistic UAV leading to the degradation in collision avoidance performance of the algorithm implemented on realistic UAV models. The proposed collision avoidance algorithm is shown to perform well when implemented on a realistic UAV model. Six degree of freedom (DoF) UAV models are used in several simulations involving various conflict geometries.

Algorithm 1. Algorithm for multiple UAV collision avoidance as implemented by UAV U_i:

Find neighbors N_i (the other UAVs within the sensor range of U_i)
Compute ZEM between U_i and each UAV $U_j \in \mathcal{N}_i$
Find $\mathcal{N}_i^* \subseteq \mathcal{N}_i$, the neighbors with which U_i has a predicted near miss
If $\mathcal{N}_i^*$ is not empty
Choose $U_{j^*} \in \mathcal{N}_i^*$ with which U_i has the immediate predicted near miss
Take the tightest turn to increase LOS rate between U_i and U_{j^*}
Else
Take Dubins path to destination

UAV model and controller design

This section discusses the six DoF UAV model and design of a controller that achieves the turn rate demand by the guidance algorithm.

UAV model

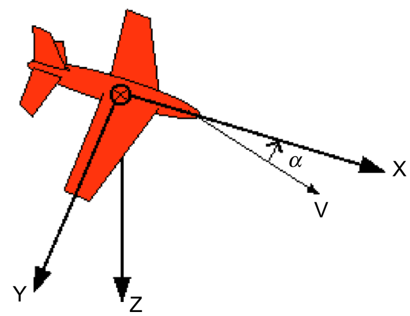
Using the standard notation of Stevens and Lewis (2003), the equations of motion are written with respect to the body axis (Figure 2) of the UAV, where (u, v, w) are velocities along the body axis and (p, q, r) are angular rates about the body axis. The Euler angles which fix the orientation of the UAV relative to an inertial frame is denoted by (ϕ, θ, ψ). The angle of attack is α and side slip is denoted as β. The elevator, aileron, and rudder defections are $(\delta_e, \delta_a, \delta_r)$ and δ_T is the throttle setting.

This paper uses the standard six DoF rigid body equations of motion for UAV models. The forces and moments are modeled linear. For example, the force balance along the x-direction of the body axis is given as:

$$\dot{u} = r v - q w + \frac{1}{m} F_x \tag{1}$$

where, m is the mass (kg) of the UAV. Here, the force (N) F_x is modelled as:

Figure 2 Body axis of a UAV



$$\begin{aligned} F_x = & -mg\sin\theta \\ & + \frac{1}{2}\rho V^2 S\left(C_{X_0} + C_{X_\alpha}\alpha + C_{X_q}\frac{cq}{V} + C_{X_{\delta_e}}\delta_e\right) \\ & + \frac{1}{2}\rho S_{prop}\left(k_{motor}^2 C_{X_{\delta_T}}^2 \delta_T^2 - V^2\right) \end{aligned} \tag{2}$$

where, the stability derivatives have usual definitions and V is the total velocity (m/s). An electric motor-propeller combination is assumed to provide thrust, where S_{prop} is the propeller area (m^2) and k_{motor} and $C_{X_{\delta T}}$ are dimensionless motor parameters.

Similarly, the other force equations and the moment equations, which are standard in the literature, can be obtained. The force and moment equations along with the kinematic and the navigational equations form the entire set of six DoF equations.

For simulations in this section, the UAV model in Aviones flight simulator (MAGICC Lab and BYU, 2010) with a span of 1.4224 m, weight 1.56 kg, and having a flying wing (with mean chord of 0.3302 m) configuration is used. The electric motor parameters are $S_{prop} = 0.0314\,\mathrm{m}^2$, $C_{X_{\delta_T}} = 1$, and $k_{motor} = 20$. A value of $1.2682\,\mathrm{kg/m}^3$ is used for the atmospheric density. The moment of inertias of the UAV model are $\mathcal{J}_x = 0.1147$, $\mathcal{J}_y = 0.0576$, $\mathcal{J}_z = 0.1712$, and $\mathcal{J}_{xz} = 0.0015\,\mathrm{kg\,m}^2$. Table I gives the stability and control derivatives of the model. To reduce the size of the table only those derivatives of the UAV model which has non-zero values are listed. The control surface deflection limits for elevator and aileron are $\pm 40°$.

Since the guidance commands given by the collision avoidance algorithm are in terms of steady level turns, a controller is needed to execute the same. This controller, designed about the steady level flight equilibrium point, should keep the aircraft in a level flight with a constant speed while tracking the turn rate commands given by the collision avoidance algorithm. The procedure to find the control surface deflections, throttle settings, and angle of attack of aircraft required (trim conditions) to maintain a steady level flight at a given altitude and velocity, is given below.

Table I Stability and control derivatives of the UAV model

Longitudinal	C_{L_0}	C_{D_0}	C_{L_α}	C_{D_α}	C_{m_α}	C_{m_q}	$C_{L_{\delta e}}$	$C_{m_{\delta e}}$	
	0.28	0.03	3.45	0.30	−0.38	−3.6	−0.36	0.5	
Lateral	C_{Y_β}	C_{l_β}	C_{n_β}	C_{l_p}	C_{n_p}	C_{l_r}	C_{n_r}	$C_{l_{\delta a}}$	$C_{n_{\delta a}}$
	−0.98	−0.12	0.25	−0.26	0.022	0.14	−0.75	0.08	0.06

Trimming aircraft for a level flight

The nonlinear UAV dynamics, consisting of force, moment, and kinematic equations, can be compactly written as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\delta}) \tag{3}$$

where state $\mathbf{x} = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi]$ and control $\boldsymbol{\delta} = [\delta_e \quad \delta_a \quad \delta_T]$.

To find the control surface deflections, the thrust throttle setting, and the angle of attack for a level flight at a specified speed V, the following optimization problem is solved:

$$\min_{(\alpha,\delta)} |\dot{\mathbf{x}}|_2 \quad \text{subject to} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\delta}) \tag{4}$$

$$\begin{aligned} &p = q = r = 0 \\ &\phi = 0, \theta = \alpha, \psi = 0 \\ &u = V\cos\alpha, \beta = 0, w = V\sin\alpha \end{aligned}$$

This multidimensional optimization problem, with decision variables $(\alpha, \delta_e, \delta_a, \delta_r)$, can be reduced to a one-dimensional search problem as follows.

For an aircraft that is symmetric about the XZ plane, a level flight would mean a zero aileron deflection, that is, $\delta_a = 0$. An explicit expression for elevator deflection in terms of α can be obtained from the $\dot{q}$ equation by setting $\dot{q} = 0$ and imposing other constrains as in equation (4):

$$\delta_e = -\frac{C_{m_0} + C_{m_\alpha}\alpha}{C_{m_{\delta_e}}} \tag{5}$$

Similarly, an explicit expression for the trim throttle setting can be obtained by setting the $\dot{u} = 0$ in equation (1):

$$\begin{aligned} \delta_T = &\frac{1}{k_{motor}} \Big[\big(2mg\sin\alpha - \rho V^2 S(C_{X_0} + C_{X_\alpha}\alpha + C_{X_e}\delta_e) \\ &+\rho V^2 S_{prop}\big)/\rho S_{prop} C_{X_{\delta_T}} \Big]^{\frac{1}{2}} \end{aligned} \tag{6}$$

Equations (5) and (6) along with constraints in equation (4) makes finding the trim condition a problem of solving the $\dot{w}$ equation for α with $\dot{w} = 0$. One way to do it is to solve it as a one-dimensional optimization problem by finding that α which minimizes $\dot{w}^2$.

For the simulations in this section, the nominal flight condition of the UAVs is a cruise flight at an altitude of 1,725 m and a velocity of 12 m/s. The solution of the optimization problem gives a trim angle of attack of 6.49° (0.1133 rad) with an elevator defection of 4.93° (0.0861 rad) and a throttle setting of 0.7423.

Controller design

The objective of the controller is to hold altitude and velocity and track the turn rate command issued by the collision avoidance algorithm. Decoupled controllers for holding altitude, velocity, and attitude (bank angle) are developed. Each of these controllers is designed by successive loop closure technique. In all these controllers, the outer loops control the slower variables and the inner loops control the faster ones.

The outermost loop tries to rectify, through feedback, the error between desired and actual values of the slowest changing variable by computing through proportional-integral blocks what the desired value of faster variable should be. The inner loops handle errors in the faster variables with the innermost loop dealing with the fastest variable.

Altitude hold controller

The altitude hold controller (Figure 3) drives to zero any error between the actual and desired altitudes. The altitude error is fed through proportional-integral blocks which output a desired pitch angle. Since a non-zero pitch angle, $\theta_{\text{ref}} = \alpha$, is required to keep a level flight, it is accounted for when computing the error in pitch. The innermost loop in altitude controller is the one with fastest variable q. The trim elevator setting, $\delta_{e_{\text{ref}}}$ is accounted for in the controller.

The values of the proportional-integral controller gains, obtained after a manual tuning, are $K_{P_h} = 5, K_{I_h} = 0.5, K_{P_\theta} = 5, K_{I_\theta} = 0.5, K_{P_q} = 1$ and $K_{I_q} = 0.1$.

Velocity hold controller

Figure 4 shows the velocity hold controller. As the controller generates commands about the equilibrium point of steady level flight, the trim throttle setting, $\delta_{T_{\text{ref}}}$, corresponding to reference velocity, $V_{\text{ref}} = V$, is accounted for. The controller gain values used for simulations are $K_{P_V} = 5$ and $K_{I_V} = 0.5$.

The throttle command causes an aircraft to climb/descent while the elevator deflection causes a change in its pitch attitude. Therefore, ideally, the throttle control should be used for altitude hold. However, the controller developed in this paper uses the elevator deflection as the control in the altitude hold loop and throttle as the control in the velocity hold loop. The combined effect of the altitude and velocity hold loops will help the aircraft to keep a steady level flight.

If the aircraft is below the required altitude, then the elevator controller in the altitude hold loop pitches the aircraft up. This causes the angle of attack of aircraft, and thus the drag, to increase. The increase in drag will cause a velocity drop, to compensate for which the velocity hold loop will increase the throttle, causing the aircraft to climb the required the altitude.

Attitude hold controller

The UAV banks to turn. Thus, a required turn rate is obtained by rolling the UAV to the corresponding bank angle. Therefore, a controller is designed to hold the commanded bank angle. The controller to hold a commanded bank angle has a structure shown in Figure 5. It has two loops, of which the outer loop corrects the error in the bank angle (slow variable) which when fed through the proportional-integral blocks will demand the required value of the roll rate (fast variable).

The gain values for attitude hold controller that were chosen, by tuning the inner loop first and then the outer loop by successive loop closure, are $K_{P_\varphi} = 5, K_{I_\varphi} = 0.5, K_{P_p} = 2$, and $K_{I_p} = 0.05$.

When a UAV banks, the lift vector tilts causing the UAV to lose altitude. The reference altitude is regained by the combined action of altitude and velocity hold controllers as described earlier.

Testing the response of the UAV

Both the collision avoidance and the Dubins path to destination algorithms demand specific turn rates. A steady level turn rate is achieved by maintaining a bank angle while holding velocity and altitude. The response of the UAV, augmented with controllers designed as described above, to bank angle command is tested with a 3-2-1-1 bank angle command. The 3-2-1-1 is a non-periodic rectangular pulse input which is commonly used during flight tests to excite all the modes of an aircraft. The UAV exhibits a good response

Figure 3 Altitude hold controller

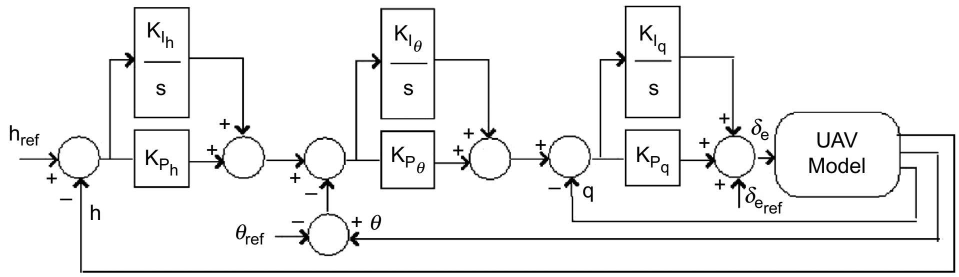


Figure 4 Velocity hold controller

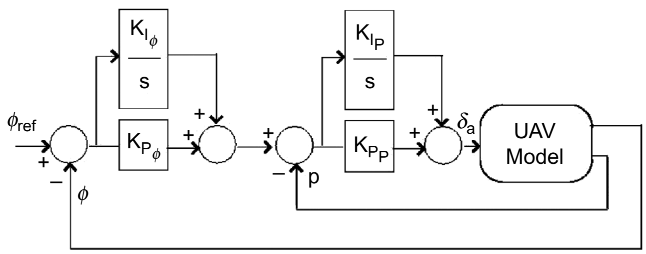
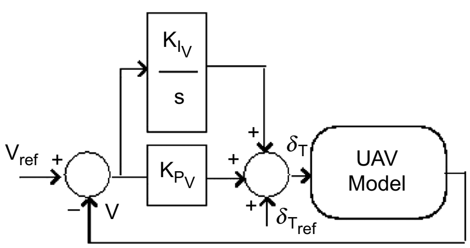


Figure 5 Attitude hold controller



to a 3-2-1-1 bank angle command is shown in Figure 6. The 3-2-1-1 command required the bank angle to be 15° for the first 3 s, −15° for the next 2 s, again 15° for a second and −15° for the subsequent second, and a zero bank angle thereafter. It is to be noted that while the UAV is trying to achieve the demanded bank angles, the velocity and altitude is maintained throughout via appropriate elevator deflection and throttle commands.

The plot of steady state turn rate, $\dot{\psi}$, achieved for different bank angles is shown in Figure 7. It is because the steady state turn rate and bank angle have a linear relationship that a bank angle command is given to obtain a turn rate of $\dot{\psi}$. For a desired turn rate, the corresponding bank angle required can be found from the plot in Figure 7 as:

$$\dot{\psi}_{\text{achieved}} = 0.75\phi \tag{7}$$

The transient of the achieved turn rate for a commanded bank angle of 10° is shown in Figure 7. The transient response has a settling time of around 2 s. It was this transient response that was used to tune the attitude hold proportional-integral controller.

Simulation results for six DoF UAV model

This section presents simulation results for various multiple UAV conflict scenarios. The six DoF UAV equations along with the controller equations are solved simultaneously using a fourth-order Runge-Kutta integration scheme with a time step of 0.01 s.

Tailored test case

A test case involving 5 UAVs is presented. The initial positions and orientations of the UAVs (Table II) are selected so that during flight they encounter multiple conflicts. The destination of each UAV is 200 m ahead of the initial position in the direction of its initial orientation. Figure 8 shows the conflict resolved trajectories of UAVs. Shown in Figure 9 are the important states (bank angle, altitude and velocity) and the corresponding control inputs (aileron deflection, elevator deflection and throttle setting) for one of the UAVs. Note that the altitude and velocity of the UAV were maintained throughout the simulation time.

Random flight test case

A test case of random flights is considered where the UAVs fly from random points on an outer circle, the destination being random points on an inner circle (Figure 10). Each simulation is done for a specified aircraft/UAV density or the number of aircraft in the concerned airspace. When the simulation begins, a new aircraft is introduced at a random point on the outer circle at every 5 s interval, which is assigned a random point on the inner circle as destination, till the required aircraft density is achieved. Whenever an aircraft reaches the destination, a new aircraft is added at randomly selected points on the outer circle to keep the number of aircraft constant in the airspace. Once the airspace achieves a required density, information regarding efficiency and number of near misses is collected for a simulation time of 50 min. Since the test case involves random introduction of UAVs and random assignment of destination points, it is called a "random flights" test case. As there is a randomness involved in this test case, it is expected that if the simulations are carried out sufficiently large number of times, almost all possible conflict geometries would be encountered.

Throughout the test, all UAVs try to maintain a constant speed of 12 m/s with the aid of velocity hold loop of the controller. The maximum turn rate is limited to 10°/s. The performance measures are the number of near misses and efficiency. Any location of two UAVs within 10-m results in a near miss. If T_j^a is the actual flight time of the jth UAV and T_j^i is its ideal flight time (that is, the time taken to reach the destination if the UAV were not to make any deviation for

Figure 6 Response (solid line) of six DoF UAV model to bank angle command (dashed line)

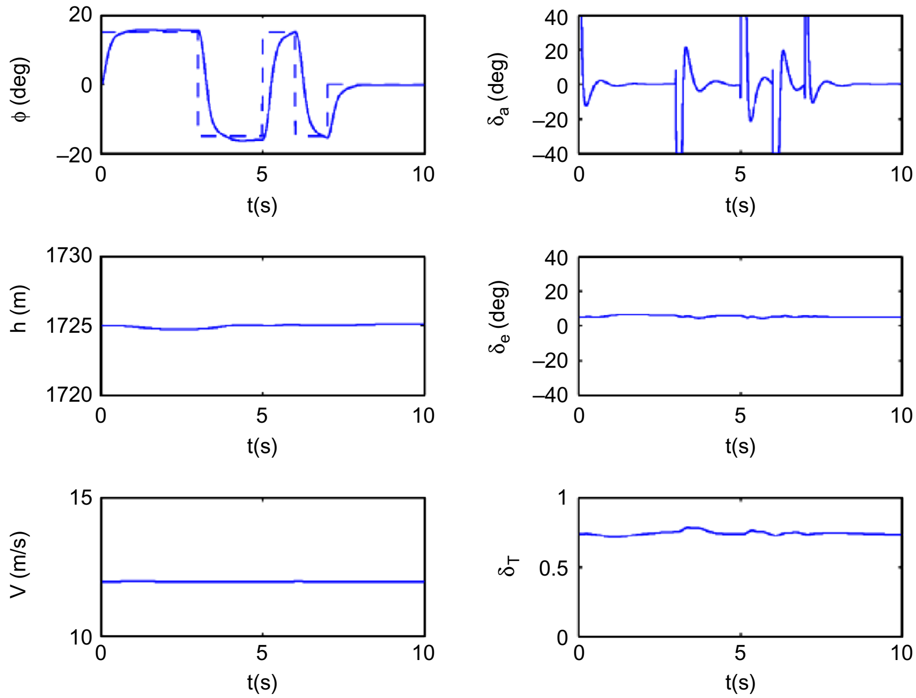
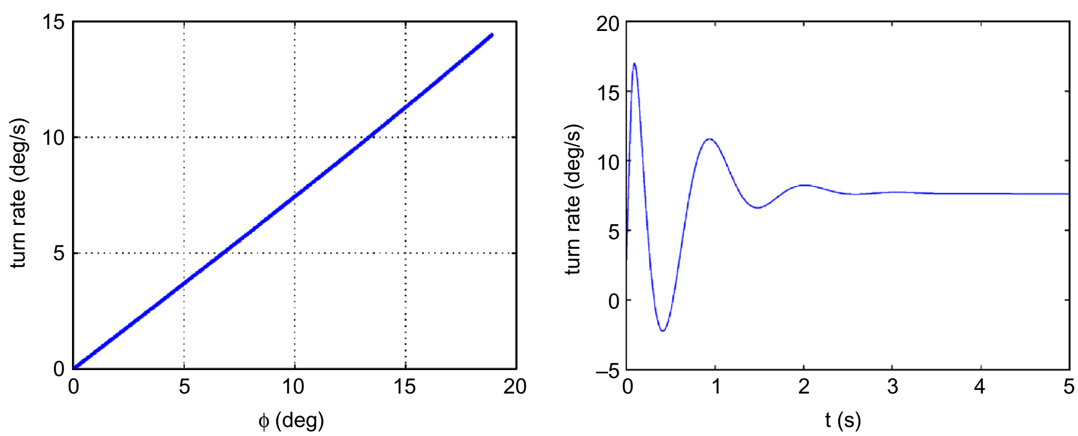


Figure 7 Plots of turn rates for six DoF UAV model



Note: Turn rate achieved for various bank angles and turn rate transient for bank angle of 10°

Table II Initial positions and orientations of UAVs in Figure 8

UAV	1	2	3	4	5
Initial position (*x*, *y*) in meters	(60, − 40)	(− 30, − 70)	(− 90, − 40)	(20,70)	(90,10)
Initial orientation in degrees	75	90	15	245	195

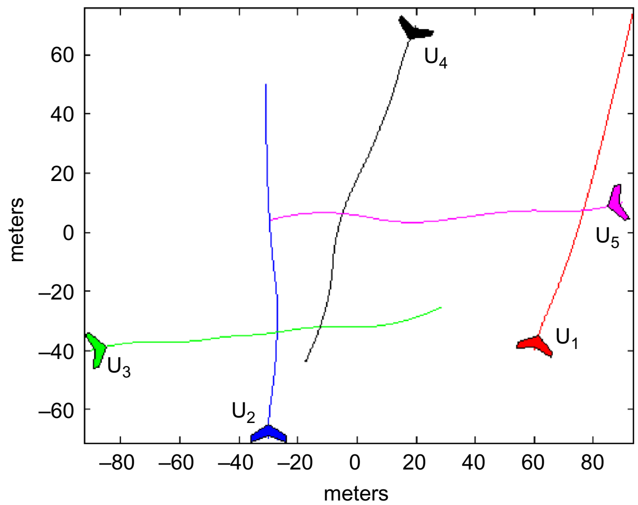
collision avoidance), then the efficiency for N UAVs is:

$$\text{Efficiency} = \frac{1}{N}\sum_{j=1}^{N}\frac{T_j^i}{T_j^a}. \qquad (8)$$

An algorithm with lower efficiency does a more conservative collision avoidance maneuver. Thus, the efficiency metric gives an insight into the optimality of the algorithm in terms of time taken to arrive at the destination while avoiding collisions.

Aircraft Engineering and Aerospace Technology: An International Journal
Volume 83 · Number 6 · 2011 · 388–396

Figure 8 Collision avoidance of five realistic UAVs



The desired separation parameter, which is a tuning parameter, in the proposed collision avoidance algorithm is chosen as 20 m. The UAVs are assumed to have a sensor range of 200 m. The UAVs can detect the position and velocity of other UAVs within this range. A near miss is deemed to occur if two UAVs come within 10 m of each other.

Simulations were done for 20, 40, 60, and 80 UAVs. For each case ten simulations were done. For the random flight test, radii of 4 and 5 km were used for the inner and outer circles. There was no near miss in any of the simulations in this case. The results are shown in Table III. It is to be noted that, for the case of 80 UAVs, the possible occurrences of about 37 near misses per simulation is reduced to zero, without much loss in efficiency, when the proposed collision avoidance algorithm is used.

As seen from Table III, the efficiencies achieved are high. This could be because the area of the test field considered is large. Therefore, another case is considered where the inner and outer radii are selected as 400 and 500 m. The results for this case are given in Table IV. As seen from the table, not applying any collision avoidance maneuver will result in over 2,000 near misses for an aircraft density of 60. The application of the collision avoidance algorithm reduces the average number of near missed to about 2. However, this reduction in the number of near misses comes at the cost of efficiency. The efficiency reduces to about 90 percent which means that some of the UAVs had to take long detours to avoid near misses. Nonetheless, even such efficiency is acceptable considering the small size of the free flight area considered for this study and the achieved reduction in the number of near misses.

Conclusion

This paper proposed a simple algorithm to avoid collisions and reduce near misses among UAVs in free flight in high-density airspace. The algorithm uses concepts from missile guidance theory, like predicted ZEM and time-to-go, to decide the turn rate of a UAV that will lead to deconfliction of a collision course. The algorithm is robust, computationally efficient, and is easy to implement. The collision avoidance algorithm was implemented on realistic six DoF UAV models augmented with controllers designed using successive loop closure. Using these, in a free-flight high-density traffic simulation environment, it was shown that the proposed collision avoidance algorithm performs well.

Figure 9 Control inputs and corresponding states for one of the UAVs in test case shown in Figure 8

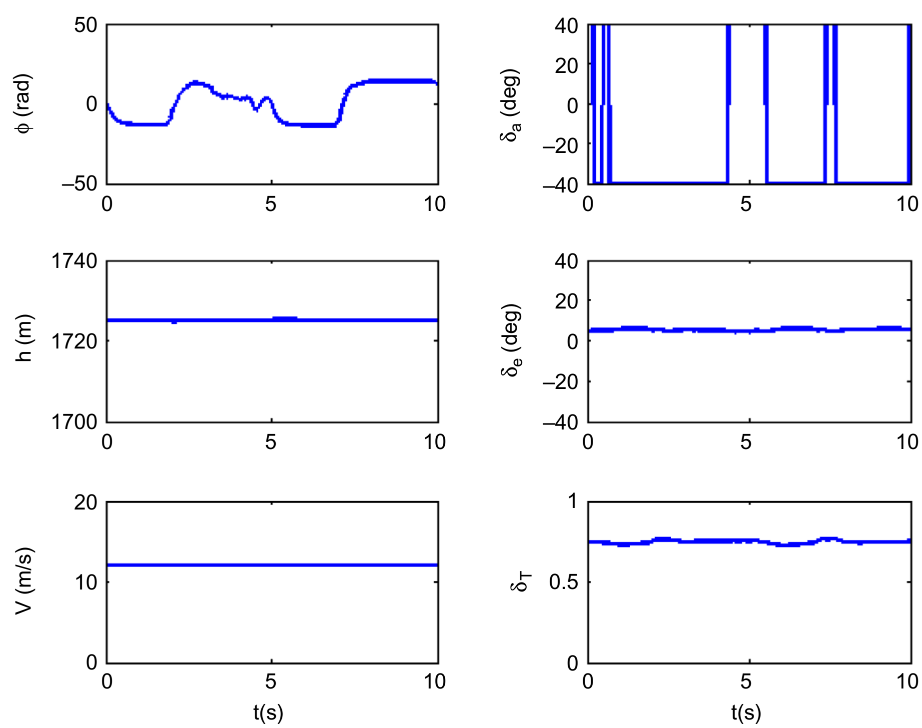


Figure 10 The test case of random flights

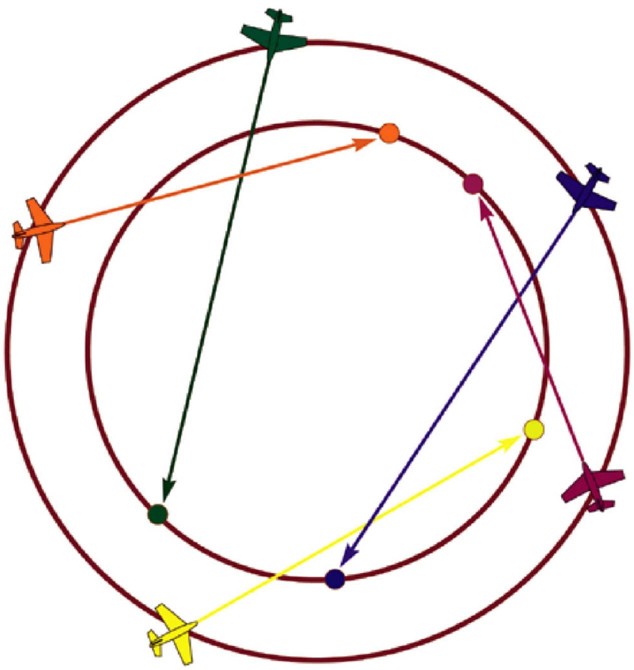


Table III Random flight test case for six DoF UAV models and inner and outer radii of 4 and 5 km

No. of UAVs	Without collision avoidance Near misses	Without collision avoidance Efficiency	With collision avoidance Near misses	With collision avoidance Efficiency
20	2.3	100	0	99.999
40	8.1	100	0	99.994
60	22.4	100	0	99.979
80	36.9	100	0	99.981

Note: Values averaged over ten runs

Table IV Random flight test case for six DoF UAV models and inner and outer radii of 400 and 500 m

No. of UAVs	Without collision avoidance Near misses	Without collision avoidance Efficiency	With collision avoidance Near misses	With collision avoidance Efficiency
20	218.1	100	0.1	96.15
40	899.1	100	1.6	89.17
60	2,027.9	100	1.4	89.11

Note: Values averaged over 10 runs

References

Albaker, B.M. and Rahim, N.A. (2009a), "A survey of collision avoidance approaches for unmanned aerial vehicles", *Proceedings of the International Conference for Technical Postgraduates, Kuala Lumpur, Malaysia*.

Albaker, B.M. and Rahim, N.A. (2009b), "Straight projection conflict detection and cooperative avoidance for autonomous unmanned aircraft systems", *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications, Kuala Lumpur, Malaysia*, pp. 1965-9.

Archibald, J.K., Hill, J.C., Jepsen, N.A., Strirling, W.C. and Frost, R.L. (2008), "A satisficing approach to aircraft conflict resolution", *IEEE Transactions on System, Man, and Cybernetics – Part C: Applications and Reviews*, Vol. 38 No. 4, pp. 510-21.

Bicchi, A. and Pallottino, L. (2000), "An optimal cooperative conflict resolution for air traffic management systems", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1 No. 4, pp. 221-32.

Chakravarthy, A. and Ghose, D. (1998), "Obstacle avoidance in a dynamic environment: a collision cone approach", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 28 No. 5, pp. 562-74.

Dubins, L.E. (1957), "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", *American Journal of Mathematics*, Vol. 79 No. 3, pp. 497-516.

Eby, M. and Kelly, W.E. (1999), "Free flight separation assurance using distributed algorithms", *Proceedings of the IEEE Aerospace Conference, San Francisco, CA, USA*, pp. 429-41.

Fiorini, P. and Shiller, Z. (1998), "Motion planning in dynamic environments using velocity obstacles", *The International Journal of Robotics Research*, Vol. 17 No. 7, pp. 760-72.

Fox, D., Burgard, W. and Thrun, S. (1997), "The dynamic window approach to collision avoidance", *IEEE Robotics and Automation Magazine*, Vol. 4 No. 1, pp. 23-33.

Fraichard, T. and Asama, H. (2004), "Inevitable collision states – a step towards safer robots?", *Advanced Robotics*, Vol. 18 No. 10, pp. 1001-24.

Frazzoli, E., Pallottino, L., Scordio, V. and Bicchi, A. (2005), "Decentralized coopperative conflict resolution for multiple nonholonomic vehicles", paper presented at AIAA Guidance, Navigation and Control Conference and Exhibit, San Francisco, CA, Paper No. AIAA-2005-6048.

George, J. and Ghose, D. (2009), "A reactive inverse PN algorithm for collision avoidance among multiple unmanned aerial vehicles", *Proceedings of the American Control Conference, Missouri, MO, USA*, pp. 3890-5.

Gomez, M.L. and Fraichard, T. (2009), "Benchmarking collision avoidance schemes for dynamic environments", *Proceedings of the ICRA Workshop on Safe Navigation in Open and Dynamic Environments, Kobe, Japan*.

Kuchar, J.K. and Yang, L.C. (2000), "A review of conflict detection and resolution modeling methods", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1 No. 5, pp. 179-89.

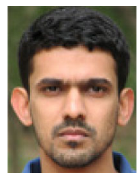
Lalish, E., Morgansen, A. and Tsukamaki, T. (2008), "Decentralized reactive collision avoidance for multiple unicycle-type vehicles", *Proceedings of the American Control Conference, Seattle, WA, USA*, pp. 5055-61.

MAGICC Lab and BYU (2010), *Aviones: UAV Flight Simulator*, available at: http://aviones.sourceforge.net/ (accessed 11 May 2010).

Panyakeow, P. and Mesbahi, M. (2010), "Decentralized deconfliction algorithms for unicycle UAVs", *Proceedings of the American Control Conference, Baltimore, MD, USA*, pp. 794-9.

Rahmani, A., Kosuge, K., Tsukamaki, T. and Mesbahi, M. (2008), "Multiple UAV deconfliction via navigation functions", paper presented at AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI, Paper No. AIAA-2008-6626.
Roussos, G., Dimarogonas, D.V. and Kyriakopoulos, K.J. (2010), "3D navigation and collision avoidance for nonholonomic aircraft-like vehicles", *International Journal of Adaptive Control and Signal Processing*, Vol. 24 No. 10, pp. 900-20.
Samek, J., Sislak, D., Volf, P. and Pechoucek, M. (2007), "Multi-party collision avoidance among unmanned aerial vehicles", *Proceedings of the International Conference on Intelligent Agent Technology, Fremont, CA, USA*, pp. 403-6.
Shiller, Z., Large, F. and Sekhavat, S. (2001), "Motion planning in dynamic environments: obstacles moving along arbitrary trajectories", *Proceedings of IEEE International Conference on Robotics and Automation, Seoul, Korea*, pp. 3716-21.
Shim, D.H., Kim, H.J. and Sastry, S. (2003), "Decentralized nonlinear model predictive control of multiple flying robots in dynamic environments", *Proceedings of the IEEE Conference on Decision and Control, Maui, HI, USA*, Vol. 4, pp. 3621-6.
Sislak, D., Rehak, M., Pechoucek, M., Pavlicek, D. and Uller, M. (2006), "Negotiation-based approach to unmanned aerial vehicles", *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, Prague, Czech Republic*, pp. 279-84.
Smith, A.L. and Harmon, F.G. (2010), "UAS collision avoidance algorithm based on an aggregate collision cone approach", *Journal of Aerospace Engineering*, Vol. 1.
Stevens, B.L. and Lewis, F.L. (2003), *Aircraft Control and Simulation*, Wiley, Hoboken, NJ.
Tomlin, C., Pappas, G.J. and Sastry, S. (1998), "Conflict resolution for air traffic management: a case study in multi-agent hybrid systems", *IEEE Transactions on Automatic Control*, Vol. 43 No. 4, pp. 509-21.
US Department of Defense (2005), *Unmanned Aircraft Systems (UAS) Roadmap 2005 – 2030*, Office of the Secretary of Defense, Arlington, VA.

About the authors



Joel George Manathara is a graduate student with the Department of Aerospace Engineering at Indian Institute of Science, Bangalore. His interests are in multi-agent systems, flight dynamics and controls, and algebraic geometry. Joel George Manathara is the corresponding author and can be contacted at: joel@aero.iisc.ernet.in

Debasish Ghose is a Professor with the Department of Aerospace Engineering, Indian Institute of Science, Bangalore. His research interests are in guidance and control of aerospace vehicles, collective robotics, multiple agent decision making, distributed decision-making systems, and scheduling problems in distributed computing systems.