Autonomous obstacle avoidance for fixed-wing unmanned aerial vehicles

A. H. J. de Ruiter
aderuiter@ryerson.ca
S. Owlia
Ryerson University
Toronto
Canada

ABSTRACT

This paper investigates a method for autonomous obstacle avoidance for fixed-wing unmanned aerial vehicles (UAVs), utilising potential fluid flow theory. The obstacle avoidance algorithm needs only compute the instantaneous local potential velocity vector, which is passed to the flight control laws as a direction command. The approach is reactive, and can readily accommodate real-time changes in obstacle information. UAV manoeuvring constraints on turning or pull-up radii, are accounted for by approximating obstacles by bounding rectangles, with wedges added to their front and back to shape the resulting fluid pathlines. It is shown that the resulting potential flow velocity field is completely determined by the obstacle field geometry, allowing one to determine a non-dimensional relationship between obstacle added wedge-length and the corresponding minimum pathline radius of curvature, which can then be readily scaled in on-board implementation. The efficacy of the proposed approach has been demonstrated numerically with an Aerosonde UAV model.

Paper No. 4239. Manuscript received 26 July 2014, revised version received 14 May 2015, accepted 18 June 2015.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

NOMENCLATURE

a	modelled obstacle width (m)
d	activation/deactivation line distance from obstacle (m)
g	Earth's gravitational acceleration at sea level (m/s^2)
h	modelled obstacle wedge length (m)
ℓ	geometric sink strength (m)
L	aircraft's lift (N)
n	aircraft load factor
$\hat{\mathbf{n}}_i$	unit outward normal of ith panel
$R_{min,P}$	aircraft minimum pull-up radius (m)
$R_{min,T}$	aircraft minimum turning radius (m)
S_i	length of ith panel (m)
V_∞	potential flow free-stream velocity (ms^{-1})
V_a	aircraft's airspeed (ms^{-1})
W	aircraft's weight (N)
$(\bar{x}_i, \bar{y}_i)$	midpoint of ith panel (m)
(x_g, y_g)	sink location (m)
(X_i, Y_i)	starting point of ith panel (m)
β_i	angle of ith panel normal with positive x-axis (rad)
δ	delta region size
δ_y	shortest distance between nominal flight path and side of obstacle (m)
δ_z	shortest distance between nominal flight path and top of obstacle (m)
γ_d	non-dimensional activation/deactivation line distance from obstacle
γ_l	non-dimensional geometric sink strength
ϕ	aircraft's bank angle (rad)

1.0 INTRODUCTION

Unmanned aerial vehicles (UAVs) are being developed in a variety of applications, mainly for military purposes. However, research and demand for civilian applications are increasing. Some examples of such applications are: geophysical surveys[4], forest fire monitoring[7], environmental monitoring[10], border patrol[16], aerial photography[27], monitoring pipelines[17] and even wildlife research[19].

The use of UAVs can be attractive for several reasons. Firstly, UAVs can operate in situations too dangerous for human pilots. Secondly, UAVs can be flown at lower altitudes than manned aircraft[5]. For example, this feature is attractive for geophysical surveys where lower altitude flight increases the resolution of the resulting map. Finally, due to the absence of an on-board pilot, smaller aircraft can typically be used, which decreases the price of the aircraft and yield to savings in fuel consumption.

UAVs have different levels of autonomy. In the absence of autonomy a UAV can only be remotely piloted. At a basic level of autonomy, a flight trajectory is preplanned and the aircraft has the ability to complete a mission. If the system has the ability to avoid obstacles and potential collision scenes, the autonomy level of the aircraft is even higher. In such a UAV, the operator uploads a flight mission to the UAV, and only monitors the flight. However, the level of autonomy of a UAV is not limited to what is mentioned here. For instance, in forest fire monitoring, a UAV can be given the ability to define its flight trajectory during flight, so as to follow the boundaries of the forest fire[7].

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

The motivation for the work in this paper is the development of a fixed-wing UAV, capable of autonomously performing high resolution geophysical surveys at low flight altitudes. The geophysical survey consists of multiple survey lines that are equally spaced over the area of interest. In order to account for terrain, a digital elevation model (DEM) of the survey area is used. This model is provided by the Shuttle Radar Topography Mission (SRTM) and is accessible through The Consultative Group on International Agricultural Research (CGIAR). The survey lines are then offset from the DEM in order to keep the aircraft at a constant height above ground level (AGL)[6]. For the measured magnetic fields to have a high-resolution, the aircraft is desired to fly below 50 metres AGL.

With flight altitudes below 50 metres AGL, the main obstacles the UAV may encounter are powerlines and their towers, telecommunication towers and trees, as well as unmodelled terrain due to the coarse DEM. The UAV is not intended to fly over urban areas, so environments are not expected to be cluttered. However, the occasional buildings, silos and industrial chimneys are also potential obstacles that may not be known prior to flight. Given that the obstacles are initially unknown, the obstacle avoidance method needs to be implementable in real-time within the limited computational resources available on the aircraft.

Obstacle avoidance is a topic that has been researched in different fields that require autonomous operation. The first field where it was researched is robotics[9,11,20,28,29]. With the advent of UAVs, and the increasing desire for autonomous UAVs, robotic obstacle avoidance methods were employed for UAVs[1,18,35]. However, because UAVs and robots have different properties and constraints, the methods being used on UAVs had to be specifically tailored to their new application. For instance, fixed-wing UAVs cannot stop and change direction, whereas robots and rotary-wing UAVs can.

In the literature, obstacle avoidance can be divided into two classes: 1. When obstacle positions and properties are known prior to the start of a mission, the avoiding method is known as trajectory planning (TP). Other terminologies such as, path planning, global guidance and global path planning have been used as well. In this case, trajectories are determined offline, and as such computational time and resources is not a limiting factor. 2. When the existence of obstacles is initially unknown and they 'pop-up' in real-time, the method is called Reactive obstacle avoidance (ROA) or local path planning. In this case, the computation time for an avoidance manoeuvre is of great importance since any delay in evasive commands can lead to collision.

There are many different techniques that have been developed for obstacle avoidance. Road map methods utilise a network of straight lines that do not intersect obstacles and connect the starting point to the goal of a robot. To create such a network, the operational space of the vehicle is divided into an obstacle-free subspace and an obstacle subspace. Then, a network of straight lines within the obstacle-free subspace is created. Using a graph search algorithm, a path is then chosen within that network, which is guaranteed to be obstacle free[39]. Different algorithms such as, Dijkstra and A* have been employed to perform the graph search; some being faster at the price of losing optimality[26]. The road map method is not computationally expensive to implement. However, the trajectories created do not directly take the vehicle's kinematic or dynamic constraints into account. In Ref. 8, Voronoi diagrams were used to generate a path for a UAV with minimal exposure to *a priori* known radars. In this approach, the Voronoi diagrams were fitted with arcs at the diagram's vertices to incorporate the aircraft's minimum turning radius. Nevertheless, in a cluttered environment, the arcs might cross obstacles, and therefore trial and error is required. In Ref. 18, a two-level architecture for UAV OA was proposed that consisted of a higher level trajectory planning and lower level reactive obstacle avoidance. For the trajectory planning, a road map was created by a node expansion technique while considering aircraft kinematic constraints. For reactive obstacle avoidance, a trajectory was created by interconnecting a set of dynamically

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

feasible primitives. These primitives were short manoeuvres that were feasible for the aircraft and were a function of the aircraft's starting state. A graph search method was then used to connect a number of feasible primitives that would avoid the obstacle. A shortcoming of this method is that for different aircraft, a complex set of primitives need to be estimated. In addition, the storage of these motion primitives requires a large memory[31].

Rapidly exploring random trees (RRT)[24,25] have been used for obstacle avoidance. In cluttered environments, this method can become computationally intensive[31]. In Ref. 1, Amin *et al* incorporated the constraints of a rotary-wing aircraft within the RRT method. To achieve this, an extra verification step was added to the RRT algorithm: at each step while expanding the tree, the new segments were checked to satisfy a maximum turn rate. The results were trajectories that could be commanded to the UAV. Unfortunately, they were not tested on the UAV or in simulation. Therefore the method's ability to run in realtime was not assessed. Similarly, in Ref. 35, the RRT method was implemented on a flying wing micro air vehicle, where the the algorithm took up to five seconds to find an obstacle free trajectory through an urban area.

Artificial potential fields (APF)[20] have been proposed for obstacle avoidance. Repulsive potential fields are assigned to obstacles, while an attractive potential field is assigned to a goal point. The gradient of the potential field is then used as an artificial force to be applied to the vehicle by its actuators, to manoeuvre it around the obstacles toward the goal point. A virtue of the APF method is its high computation speed, making it very suitable for real-time implementation. The major weakness the APF method has, is the occurrence of local minima in the overall potential field of an obstacle and a goal[21], which could result in the vehicle becoming stuck at a local minimum. In the previous developments of the APF method, the goal was always assumed to be far from obstacles. However, if the goal is close to an obstacle, the repulsive force generated by the obstacle can be larger than the attractive force of the goal and as a result the robot will miss the goal point. In Ref. 14, the APF was modified to solve this problem by taking the distance between the robot and the goal into account when defining the APFs. In Ref. 15, a new APF was proposed that could be used for moving obstacles and goals. Note that in Ref 14 and Ref 15, the problem of local minima was not addressed.

To overcome the APF local minima problem in the scenario of multiple obstacles, the usage of harmonic potential functions was suggested in Ref. 22, which do not exhibit local extrema. Interestingly, in the potential flow theory from fluid mechanics, a flow potential function is harmonic. In Ref. 41, stream functions of potential flow were used to navigate a mobile robot. Kim *et al*[22] used the panel method from potential flow theory for the obstacle avoidance of a robotic manipulator. Specifically, obstacles were modelled as being in a uniform potential flow together with a sink located at the goal point the manipulator was intended to reach. The resulting flow field was used to guide the manipulator. In Ref. 13, the panel method was employed for multiple robot collision avoidance. Furthermore, in Ref. 12, Fahimi *et al* used the panel method for obstacle avoidance of a hyper-redundant manipulator operating in a three-dimensional space. In this approach, the two dimensional panel method was expanded to three dimensions and as a result, the manipulator could avoid three-dimensional obstacles. In Ref. 30, vortex functions were used to obtain a velocity field for obstacle and collision avoidance among single or multiple UAVs.

Model predictive control (MPC), is a popular method for the control of non-linear dynamic systems[31,34], and have also been proposed for obstacle avoidance. The appealing feature the method has is that realistic constraints such as input saturation and state constraints can be considered[31]. A drawback of MPC is that the non-linear model of the system is required to determine the receding optimal control. The model required may not always be available or might be very complex. Furthermore, high computation power is often required to compute the optimal control sequence

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

to optimise the cost function over each time interval. Shim *et al* employed MPC to a rotary-wing UAV in Ref. 38. An artificial potential function was added to the cost function of the MPC method to penalise the approaching of an obstacle. However, because potential functions were used for OA, the method is vulnerable to local minima[37]. Shim *et al* succeeded to test the MPC method on a helicopter, but the computation was performed on an on-ground PC and the control inputs were transferred via telemetry to the helicopter. Therefore, this method may not be computationally feasible for real-time onboard computing where computing resources are limited[31].

In this paper, a reactive obstacle avoidance method is developed for fixed-wing UAVs flying in uncluttered environments, such as the aforementioned UAV for performing geophysical surveys (which is the motivation for this work). As discussed above, existing obstacle avoidance methods are either too computationally expensive to implement onboard a UAV with limited computational capacity (such as the one motivating this paper), or do not adequately account for UAV kinematic constraints. Unlike other methods in the literature, the objective in this paper (as motivated by performing geophysical surveys) is not to guide the UAV from a starting point to a goal point. Rather, it is to guide the UAV around encountered obstacles, and then return the UAV to a pre-programmed flight path. To overcome these issues, in this paper the obstacle avoidance method based on panel method of potential flow is adapted for fixed-wing UAVs. A simple method for modelling obstacles is presented such that the resulting trajectory satisfies the UAV turning and pull-up constraints, for full three-dimensional flight. In particular, a non-dimensional relationship between minimum turning/pullup radius and the geometry of the modelled obstacle is obtained, which can be stored on-board in a look-up table for use in modelling an encountered obstacle. It should be noted that this relationship is valid only for a single obstacle, and hence the method developed in this paper is applicable to uncluttered environments, where encountered obstacles can be considered one at a time. The presented method requires further development for cluttered environments with several closely spaced obstacles. The method presented here is conceptually similar to the fluid flow method based on vortex functions in Ref. 30. In Ref. 30, UAV turning constraints are accommodated by extending the size of the modelled obstacle by the UAV turning radius. However, it is not demonstrated whether or not this guarantees that the resulting trajectories satisfy the UAV turning radius constraints.

The remainder of the paper is organised as follows. Section 2 presents the theory behind obstacle avoidance using the panel method. Section 3 presents the details of how obstacles are modelled such that UAV kinematic constraints are satisfied. Section 4 shows the results of the method implemented in a high fidelity simulation using the Aerosonde UAV. Finally, Section 5 contains concluding remarks.

2.0 OBSTACLE AVOIDANCE USING POTENTIAL FLOW

An inter-disciplinary OA method stemming from fluid mechanics is explained in this section. When a uniform fluid flow encounters an obstacle, fluid particles deviate from their original course and avoid it. This concept can be used for OA, by commanding the aircraft in a way that it would follow the trajectory of a fluid particle (pathline) encountering an obstacle. Since the fluid particles don't flow through obstacles, the aircraft is able to reach its goal without collision.

In fluid mechanics, the Navier-Stokes equations, which govern fluid flow, are derived from conservation of mass, momentum, and energy[36]. However, the complexity of these equations make their usage difficult for OA. For an irrotational, inviscid and incompressible flow, a scalar potential function can be found[3,36], where its gradient at any point results in the velocity of a fluid particle at that point, namely

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

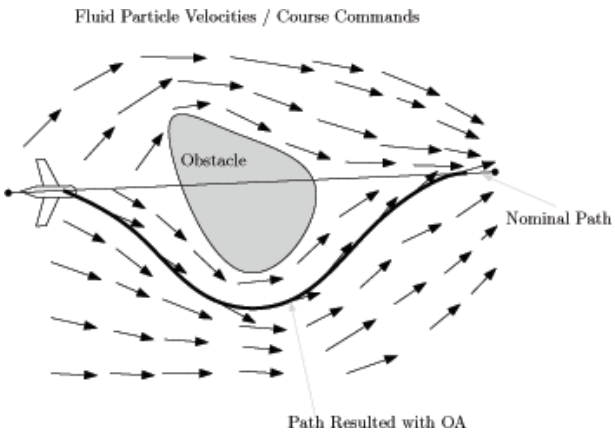


Figure 1. Obstacle avoidance using potential flow.

$$\mathbf{V} = \nabla\Phi \qquad \ldots (1)$$

Furthermore, the potential function is harmonic and does not possess local minima (unless it is constant)[33]. The presence of local minima is a known problem with obstacle avoidance based on the use of potential fields. Since potential field guidance methods guide vehicles toward lower values of the potential function, the existence of local minima can lead to vehicles becoming stuck at those local minima. The use of a harmonic potential function avoids this problem.

Given a flow potential, the resulting velocity field can be employed to guide a UAV around an obstacle. After entering the velocity field, the aircraft is assumed to be a fluid particle and therefore the velocity at each point is commanded as an instantaneous direction for the aircraft to follow. As shown in Section 2.3, provided the aircraft perfectly tracks the commanded direction of the instantaneous flow field velocity vector, the aircraft will be following a trajectory identical to that of a fluid particle in the same flow field, see Fig. 1. Note however, since no aircraft autopilot provides perfect command tracking, the path followed will be not quite the same as a fluid pathline.

A key consideration for reactive obstacle avoidance is ease of on-board implementation in real time. As outlined above, the on-board software needs only compute the flow field velocity vector at the aircraft's instantaneous position, and it is not necessary to compute the entire velocity field, nor is it necessary to determine a pathline. This makes it very suitable for real-time implementation. Additionally, to simplify computations a two-dimensional flow field is utilised, either in the horizontal plane if the aircraft is to fly around the obstacle, or the vertical plane if the aircraft is to fly over it.

In Section 2.1, a method is introduced to obtain the potential flow velocity field in the presence of obstacles.

2.1 Panel method

To obtain a potential flow field around an obstacle, the panel method[3] is used. Specifically, an obstacle is modelled by a series of panels, embedded within a uniform flow field of velocity V_∞, with direction parallel to the nominal desired flight path. For convenience, the co-ordinate system will always be defined such that the desired flight path is parallel to the x-axis. Figure 2 shows an example of a closed obstacle with n panels.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

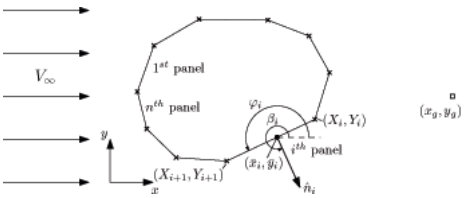


Figure 2. An *n*-panel obstacle.

As shown in Fig. 2, suppose a sink is located at (x_g, y_g), and let the starting point of each panel have co-ordinates (X_i, Y_i), the midpoint of each panel $(\bar{x}_i, \bar{y}_i)$, the outward pointing normal of each panel be $\hat{\mathbf{n}}_i$, the angle that the outward normal makes with the positive x-axis measured counter-clockwise be β_i, and $\varphi_i = \beta_i - \pi/2$. Finally, the length of each panel is given by

$$S_i = \left((X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2\right)^{1/2}. \quad \ldots (2)$$

Then, the total velocity field of an obstacle immersed in a two-dimensional uniform flow with a sink is given by Ref. 33.

$$V_x(x,y) = V_\infty + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} J_{x,j}(x,y) - \frac{V_\infty \ell}{2\pi}\left(\frac{x - x_g}{r_g(x,y)^2}\right) \quad \ldots (3)$$

and

$$V_y(x,y) = \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} J_{y,j}(x,y) - \frac{V_\infty \ell}{2\pi}\left(\frac{y - y_g}{r_g(x,y)^2}\right), \quad \ldots (4)$$

where $r_g(x,y) = \left((x - x_g)^2 + (y - y_g)^2\right)^{1/2}$. The quantities $\lambda_1 \ldots \lambda_n$ are the solutions of

$$\begin{bmatrix} \frac{I_{1,1}}{2\pi} & \cdots & \frac{I_{1,n}}{2\pi} \\ \vdots & \ddots & \vdots \\ \frac{I_{n,1}}{2\pi} & \cdots & \frac{I_{n,n}}{2\pi} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} -V_\infty \mathrm{Cos}\beta_1 + V_\infty \ell \mathrm{Cos}\beta_1 (\bar{x}_1 - x_g)/(2\pi r_g(\bar{x}_1, \bar{y}_1)^2) \\ \vdots \\ -V_\infty \mathrm{Cos}\beta_n + V_\infty \ell \mathrm{Cos}\beta_n (\bar{x}_n - x_g)/(2\pi r_g(\bar{x}_n, \bar{y}_n)^2) \end{bmatrix} \quad \ldots (5)$$

$$+ \begin{bmatrix} V_\infty \ell \mathrm{Sin}\beta_1 (\bar{y}_1 - y_g)/(2\pi r_g(\bar{x}_1, \bar{y}_1)^2) \\ \vdots \\ V_\infty \ell \mathrm{Sin}\beta_n (\bar{y}_n - y_g)/(2\pi r_g(\bar{x}_n, \bar{y}_n)^2) \end{bmatrix}$$

with

$$I_{j,j} = \pi, j = 1, \ldots, n,$$

and for $i \neq j$

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

$$I_{i,j}=\begin{cases}\dfrac{C_{i,j}}{2}\ln\left|\dfrac{S_j^2+2A_{i,j}S_j+B_{i,j}}{B_{i,j}}\right|+\dfrac{D_{i,j}-A_{i,j}C_{i,j}}{E_{i,j}}\left(\mathrm{ArcTan}\dfrac{S_j+A_{i,j}}{E_{i,j}}+\mathrm{ArcTan}\dfrac{A_{i,j}}{E_{i,j}}\right), & B_{i,j}-A_{i,j}^2>0,\\ C_{i,j}\ln\left|\dfrac{S_j+A_{i,j}}{A_{i,j}}\right|+\left(\dfrac{D_{i,j}-A_{i,j}C_{i,j}}{A_{i,j}}\right)\left(\dfrac{S_j}{S_j+A_{i,j}}\right), & B_{i,j}-A_{i,j}^2=0\\ \dfrac{C_{i,j}}{2}\ln\left|\dfrac{S_j^2+2A_{i,j}S_j+B_{i,j}}{B_{i,j}}\right|+\dfrac{D_{i,j}-A_{i,j}C_{i,j}}{2E_{i,j}}\ln\left|\dfrac{(S_j+A_{i,j}-E_{i,j})(A_{i,j}+E_{i,j})}{(S_j+A_{i,j}+E_{i,j})(A_{i,j}-E_{i,j})}\right|, & B_{i,j}-A_{i,j}^2<0,\end{cases} \quad \ldots(6)$$

with

$$\begin{aligned}A_{i,j}&=-(\bar{x}_i-X_j)\mathrm{Cos}\varphi_j-(\bar{y}_i-Y_j)\mathrm{Sin}\varphi_j,\\ B_{i,j}&=(\bar{x}_i-X_j)^2+(\bar{y}_i-Y_j)^2,\\ C_{i,j}&=\mathrm{Sin}(\varphi_i-\varphi_j),\\ D_{i,j}&=(\bar{y}_i-Y_j)\mathrm{Cos}\varphi_i-(\bar{x}_i-X_j)\mathrm{Sin}\varphi_i,\\ E_{i,j}&=\sqrt{B_{i,j}-A_{i,j}^2}.\end{aligned} \quad \ldots(7)$$

The quantities $J_{x,j}(x, y)$ and $J_{y,j}(x, y)$ in Equations (3) and (4) are given by

$$J_{\cdot,j}(x,y)=\begin{cases}\dfrac{C_{\cdot,j}}{2}\ln\left|\dfrac{S_j^2+2A_jS_j+B_j}{B_j}\right|+\dfrac{D_{\cdot,j}-A_jC_{\cdot,j}}{E_j}\left(\mathrm{Arctan}\dfrac{S_j+A_j}{E_j}+\mathrm{Arctan}\dfrac{A_j}{E_j}\right), & B_j-A_j^2>0,\\ C_{\cdot,j}\ln\left|\dfrac{S_j+A_j}{A_j}\right|+\left(\dfrac{D_{\cdot,j}-A_jC_{\cdot,j}}{A_j}\right)\left(\dfrac{S_j}{S_j+A_j}\right), & B_j-A_j^2=0\\ \dfrac{C_{\cdot,j}}{2}\ln\left|\dfrac{S_j^2+2A_jS_j+B_j}{B_j}\right|+\dfrac{D_{\cdot,j}-A_jC_{\cdot,j}}{2E_j}\ln\left|\dfrac{(S_j+A_j-E_j)(A_j+E_j)}{(S_j+A_j+E_j)(A_j-E_j)}\right|, & B_j-A_j^2>0,\end{cases} \quad \ldots(8)$$

where

$$\begin{aligned}A_j&=(X_j-x)\mathrm{Cos}\varphi_j+(Y_j-y)\mathrm{Sin}\varphi_j,\\ B_j&=(x-X_j)^2+(y-Y_j)^2,\\ E_j&=\sqrt{B_j-A_j^2},\end{aligned} \quad \ldots(9)$$

$$\begin{aligned}C_{x,j}&=-\mathrm{Cos}\varphi_j,\\ D_{x,j}&=x-X_j,\end{aligned} \quad \ldots(10)$$

and

$$\begin{aligned}C_{y,j}&=-\mathrm{Sin}\varphi_j,\\ D_{y,j}&=y-Y_j.\end{aligned} \quad \ldots(11)$$

Finally, the quantity $V_\infty \boldsymbol{\ell}$ is the strength of the sink located at (x_g, y_g), such that $\boldsymbol{\ell}$ determines the cross-section of flow upstream whose pathlines will end at the sink. We shall call $\boldsymbol{\ell}$ the geometric sink strength. The purpose of the sink is to draw pathlines around the obstacles back to the nominal flight path. As discussed in Section 3.3, the geometric sink strength $\boldsymbol{\ell}$ is selected in proportion to obstacle size.

Remark The panel method remains valid for multiple obstacles by the enumerating the panels of the additional obstacles such as in Fig. 3.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

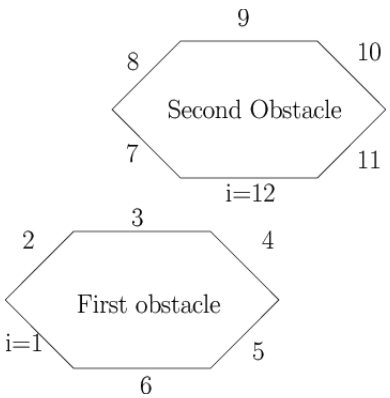


Figure 3. Enumeration of panels for multiple obstacles.

2.2 Scaling of the velocity magnitude

It will now be shown that a pathline is independent of the magnitude of the velocity field, and that the magnitude of the velocity field can in fact be time-varying, as long as the direction is time-invariant.

Let $\mathbf{r}(p):[a,b]\mapsto\mathbb{R}^n$ be a pathline, and parameterise it by some variable $p\in\mathbb{R}$ in some closed interval $a\le p\le b$. In this work, a pathline is generated from the continuous potential flow velocity field $\mathbf{V}(\mathbf{r})$: $\mathcal{D}\mapsto\mathbb{R}^n$, defined on some domain $\mathcal{D}\subset\mathbb{R}^n$. For the velocity fields given in Section 2.1, we may take the domain to be $\mathcal{D}=\mathbb{R}^2/\mathcal{O}$, where $\mathcal{O}\subset\mathbb{R}^2$ is a closed set defined by the obstacle. Then, it is readily seen that the velocity field $\mathbf{V}(\mathbf{r})$ given in Section 2.1 is continuous on the domain $\mathcal{D}$. Furthermore, since $\mathbf{V}(\mathbf{r})\cdot\hat{\mathbf{n}}(\mathbf{r})=0$ on the boundary of the obstacle $\partial\mathcal{O}$†, where $\hat{\mathbf{n}}(\mathbf{r})$ is the unit outward pointing normal at each given point on the obstacle boundary, it must be that any pathline with initial condition $\mathbf{r}(a)\in\mathcal{D}$ satisfies $\mathbf{r}(p)\in\bar{\mathcal{D}}$ for all $p\in[a,b]$ (that is, if it starts in $\mathcal{D}$, it must stay in $\bar{\mathcal{D}}$). Finally, it is to be noted that the velocity fields given in Section 2.1 satisfy $\mathbf{V}(\mathbf{r})\neq\mathbf{0}$ for all $\mathcal{D}$.

Consider any pathline $\mathbf{r}(p):[a,b]\mapsto\mathbb{R}^n$ contained in $\mathcal{D}$. If the pathline parameterisation is time ($p=t$), then the pathline $\mathbf{r}(t)$ is an integral curve of the first order differential equation

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}=\mathrm{V}(\mathbf{r}), \qquad \ldots(12)$$

with initial condition $\mathbf{r}(a)\in\mathcal{D}$ given. Since $\mathrm{d}\mathbf{r}/\mathrm{d}t\neq0$ for all $t\in[a,b]$ (because $\mathbf{V}(\mathbf{r})\neq\mathbf{0}$ on $\mathcal{D}$), the pathline may equivalently be parameterised by its arc-length, s, measured along the pathline from the initial condition $\mathbf{r}(a)$. That is, the pathline may be given by $\bar{\mathbf{r}}(s)$: $[0,s_{max}]\mapsto\mathbb{R}^n$, where $\bar{\mathbf{r}}(s)=\mathbf{r}(t(s))$, and s_{max} is the length of the pathline. It is a property of the arc-length that

$$\frac{\mathrm{d}s}{\mathrm{d}t}=\left|\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right|.$$

Therefore, from Equation (12), it is readily found that the pathline is also an integral curve of

† Note that for the panel method, the condition $\mathrm{V}(\mathrm{r})\cdot\hat{\mathrm{n}}(\mathrm{r})=0$ is only enforced at the panel midpoints, so it is not guaranteed to hold everywhere on the obstacle boundary

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s} = \frac{1}{|\mathbf{V}(\mathbf{r})|}\mathbf{V}(\mathbf{r}), \qquad \ldots (13)$$

with initial condition $\bar{\mathbf{r}}(0) = \mathbf{r}(a)$.

We are free to parameterise s by any other parameterisation τ (which can be thought of as a different time-scale). In particular, let $g(s,\tau) \geq \varepsilon > 0$ be a strictly positive function, continuous in s and piecewise continuous in τ, and suppose that s is the solution of the first order differential equation

$$\frac{\mathrm{d}s}{\mathrm{d}\tau} = g(s,\tau), \qquad \ldots (14)$$

on an appropriate interval $\tau \in [c,d]$ with initial condition $s(c) = 0$. Note that $s(\tau)$ is a strictly increasing continuous function, and the upper limit of the interval d is set such that $s(d) = s_{max}$ (which can be done since $\mathrm{d}s/\mathrm{d}\tau \geq \varepsilon > 0$). Then, the pathline is equivalently represented by $\hat{\mathbf{r}}(\tau) : [c,d] \mapsto \mathbb{R}^n$, where $\hat{\mathbf{r}}(\tau) = \bar{\mathbf{r}}(s(\tau))$, and has velocity $\mathrm{d}\hat{\mathbf{r}}/\mathrm{d}\tau = g(s,\tau)\,\mathrm{d}\hat{\mathbf{r}}/\mathrm{d}s$.

Now, let $f(\mathbf{r},\tau) > 0$ be a strictly positive function on $\mathcal{D}\times\mathbb{R}$, continuous in r and piecewise continuous in τ, with the property that one can find a strictly positive function $\partial(\mathbf{r}) > 0$, continuous on $\mathcal{D}$ such that $f(\mathbf{r},\tau) \geq \partial(\mathbf{r})$ for all $t \in \mathbb{R}$ and all $\mathbf{r} \in \mathcal{D}$. Now, for a given pathline $\bar{\mathbf{r}}(s)$, let $g(s,\tau)$ in Equation (14) be given by

$$g(s,\tau) = f(\bar{\mathbf{r}}(s),\tau)\,|\mathbf{V}(\bar{\mathbf{r}}(s))|. \qquad \ldots (15)$$

Clearly, $g(s,\tau) > 0$ for $s \in [0, s_{max}]$, and is continuous in s and piecewise continuous in τ. Note that since the interval $[0, s_{max}]$ is closed and bounded, we can readily find an $\varepsilon_1 > 0$ such that $|\mathbf{V}(\bar{r}(s))| \geq \varepsilon_1$ and $\partial(\bar{\mathbf{r}}(s)) \geq \varepsilon_2$ for $s \in [0, s_{max}]$ (just take $\varepsilon_1 = min_{s \in [0, s_{max}]}\{|\mathbf{V}(\bar{\mathbf{r}}(s))| > 0$, and $\varepsilon_2 = min_{s \in [0, s_{max}]}\{\partial(\bar{\mathbf{r}}(s))\} > 0$. Setting $\varepsilon = \varepsilon_1\varepsilon_2$, it follows from Equation (15) that $g(s,\tau) > \varepsilon$ for $s \in [0, s_{max}]$. Finally, by combining Equations (14) and (15) with Equation (13), it is readily found that the pathline is also an integral curve of

$$\frac{\mathrm{d}\bar{\mathbf{r}}}{\mathrm{d}\tau} = f(\bar{\mathbf{r}},\tau)\mathbf{V}(\bar{\mathbf{r}}), \qquad \ldots (16)$$

with initial condition $\hat{\mathbf{r}}(c) = \mathbf{r}(a)$.

To conclude, it has been shown that pathlines are integral curves of the differential Equations (12), (13) and (16). The variables used to parameterise the pathlines (t, s, τ), can simply be viewed as variables of integration in Equations (12), (13) and (16), and may be interpreted as different time-scales. Therefore, the right-hand sides of Equations (12), (13) and (16) may all be viewed as velocity fields leading to the same pathlines. As such it may be concluded that a pathline is completely determined by the direction of the velocity field at each point. The magnitude simply changes the temporal nature with which the pathline is traversed. This justifies why, at each point in an aircraft's trajectory, the direction of the local velocity field may be used to provide either a commanded course or commanded path angle (without consideration of the aircraft's speed). With perfect tracking, the aircraft will then follow a fluid pathline.

Choosing V_∞

As a consequence of the preceding analysis, the magnitude of the free-stream velocity V_∞ does not affect the pathlines. Indeed, let

$$V_\infty' = \beta V_\infty,$$

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

for some $\beta > 0$. Noting from Equation (6) that $I_{i,j}$ is independent of V_∞, Equation (5) results in

$$\lambda_i' = \beta\lambda_i,$$

Equation (8) shows that $J_{x,j}$ and $J_{y,j}$ are also independent of V_∞. Therefore, from Equations (3), (4) it follows that

$$\begin{aligned} V_x' &= \beta V_x, \\ V_y' &= \beta V_y. \end{aligned} \quad \ldots (17)$$

That is, scaling the free-stream velocity simply results in a scaling of the magnitude of the resulting velocity field, and not its direction. Hence, the pathlines do not change, and the magnitude of the free-stream velocity may be freely chosen.

2.3 Scaling of the velocity field

Consider a velocity field $\mathbf{V}(\mathbf{r})$, and a scaled velocity field $\mathbf{V}_s(\mathbf{r}) = \mathbf{V}(\mathbf{r}/\alpha)$, where $\alpha > 0$. As shown in the previous section, the resulting pathlines are integral curves of Equation (12). Let $\mathbf{r}(t)$ be a pathline corresponding to the velocity field $\mathbf{V}(\mathbf{r})$, with initial condition $\mathbf{r}(a)$, and consider the curve $\mathbf{r}_s(t) = \alpha\mathbf{r}(t)$. Then, it is readily found that $\mathbf{r}_s(t)$ is an integral curve of

$$\frac{d\mathbf{r}}{dt} = \alpha\mathbf{V}(\mathbf{r}/\alpha), \quad \ldots (18)$$

with initial condition $\mathbf{r}_s(a) = \alpha\mathbf{r}(a)$. As seen in the previous section, the pathlines are unaffected by scaling of the velocity field magnitude. Hence, $\mathbf{r}_s(t)$ is equivalently an integral curve of

$$\frac{d\mathbf{r}}{dt} = \mathbf{V}(\mathbf{r}/\alpha), \quad \ldots (19)$$

with initial condition $\mathbf{r}_s(a) = \alpha\mathbf{r}(a)$, and therefore is a pathline generated by the scaled velocity field $\mathbf{V}(\mathbf{r}_s/\alpha)$. That is, scaling of the velocity field results in the same scaling of the pathlines.

Scaling an obstacle field

As a consequence of the previous analysis, scaling of an obstacle field simply results in a scaling of the pathlines. That is, the shapes of pathlines do not change.

Indeed, consider an obstacle field scaled by a factor of $\alpha > 0$. The scaled panel midpoints and starting points are $(\bar{x}_{i,s}, \bar{y}_{i,s}) = \alpha(\bar{x}_{i,s}, \bar{y}_{i,s})$ and $(X_{i,s}, Y_{i,s}) = \alpha(X_{i,s}, Y_{i,s})$, respectively‡. In addition, the scaled sink location is $(x_{g,s}, y_{g,s}) = \alpha(x_g, y_g)$, and the scaled geometric sink strength is $l_s = \alpha l$. From Equation (7), it follows that

$$\begin{aligned} A_{i,j,s} &= \alpha A_{i,j}, \\ B_{i,j,s} &= \alpha^2 B_{i,j}, \\ C_{i,j,s} &= C_{i,j}, \\ D_{i,j,s} &= \alpha D_{i,j}, \\ E_{i,j,s} &= \alpha E_{i,j}, \end{aligned} \quad \ldots (20)$$

while from Equation (2),

‡ *The s subscript is used for the scaled obstacle*

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

$$S_{j,s} = \alpha S_j \quad \ldots (21)$$

Substitution of Equations (20) and (21) into Equation (6), shows that $I_{i,j,s} = I_{i,j}$. Consequently, from Equation (5) it is concluded that

$$\lambda_{i,s} = \lambda_i \quad \ldots (22)$$

Next, from Equations (9) to (11), it follows that

$$\begin{aligned} A_{j,s}(x,y) &= \alpha A_j(x/\alpha, y/\alpha), \\ B_{j,s}(x,y) &= \alpha^2 B_j(x/\alpha, y/\alpha), \\ C_{\cdot,j,s} &= C_{\cdot,j}, \\ D_{\cdot,j,s}(x,y) &= \alpha D_{\cdot,j}(x/\alpha, y/\alpha), \\ E_{j,s}(x,y) &= \alpha E_j(x/\alpha, y/\alpha). \end{aligned} \quad \ldots (23)$$

Therefore, from Equation (8) it can be concluded that

$$\begin{aligned} J_{x,j,s}(x,y) &= J_{x,j}(x/\alpha, y/\alpha) \\ J_{y,j,s}(x,y) &= J_{y,j}(x/\alpha, y/\alpha). \end{aligned} \quad \ldots (24)$$

Finally from Equations (3), (4), (22) and (24) it is concluded that the velocity field with scaled obstacles is given by

$$\begin{aligned} V_{x,s}(x,y) &= V_x(x/\alpha, y/\alpha) \\ V_{y,s}(x,y) &= V_y(x/\alpha, y/\alpha). \end{aligned} \quad \ldots (25)$$

Thus, by the previous analysis, this means that if an obstacle field is scaled by a factor of α, all pathlines will be scaled by α as well.

3.0 OBSTACLE MODELLING AND IMPLEMENTATION

Since the UAV is expected to fly in uncluttered environments, a typical flight scenario will be the need to avoid a single obstacle at a time. Therefore, the following discussion will be limited to the modelling and avoidance of a single obstacle. This will also allow the discussion to remain focused on the main ideas being presented in this paper, without overcomplicating details. For a discussion of how the ideas in this section may be extended to multiple obstacles under simultaneous consideration, the reader is referred to Ref. 33.

3.1 Obstacle modelling

In order to maintain computational simplicity, obstacles are modelled as rectangles, either in the horizontal plane or vertical plane depending on whether a fly-around or fly-over manoeuvre is required. These rectangles are defined such that they bound the obstacles. Initially, the only information available about a given obstacle is what the aircraft can physically see. In particular, this means that the details of the far side of the obstacle (such as its depth) are initially unknown. For this reason, the obstacle is initially assumed to have depth equal to its width (see Fig. 4 for an example). As the aircraft starts the avoidance manoeuvre and newer obstacle data becomes available, the depth of the obstacle can be updated. Additionally, for safety purposes the bounding rectangles are taken to be larger than the actual obstacle.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

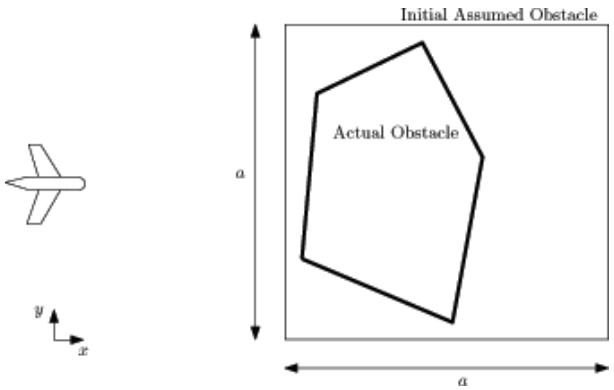


Figure 4. Exact and initially assumed obstacles.

3.2 Incorporating aircraft constraints

The velocity field obtained in Section 2.1 does not take into account the aircraft manoeuvring capabilities. In this section, aircraft constraints are introduced from Ref. 2, and a method to incorporate these constraints in the potential flow method is proposed.

A limitation fixed-wing aircraft have is the tightest turn they can make while maintaining level flight. The minimum turning radius of an aircraft is

$$R_{min,T} = \frac{V_a^2}{g\sqrt{n^2-1}}, \quad \ldots (26)$$

where V_a is the aircraft's airspeed, $g = 9{\cdot}81\text{m/s}^2$ is the Earth's gravitational acceleration at sea level, and n is the load factor of the aircraft defined as

$$n = \frac{L}{W}, \quad \ldots (27)$$

where L and W are the aircraft's lift and weight, respectively.

In turning flight, it can be shown that

$$n = \frac{1}{\text{Cos}\phi}, \quad \ldots (28)$$

where ϕ is the aircraft's bank angle. From Equation (26) in order to get the tightest turn at a given speed the highest load factor should be used.

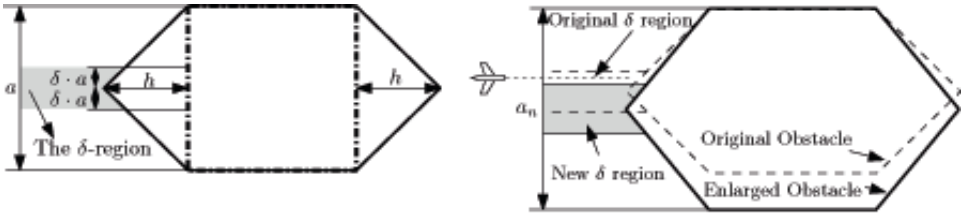
Similarly, the aircraft has a limitation on the tightest pull-up turn it can make. The minimum pull-up radius can be calculated from

$$R_{min,P} = \frac{V_a^2}{g(n-1)} \quad \ldots (29)$$

where n is given by Equation (27). Again, to obtain the tightest pull-up turn, the load factor should be maximised.

It should be noted that the constraints on turning and pull-up radii introduced in this section are for

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press



(a) Modelled obstacle (a) Enlarged obstacle

Figure 5. Modelled obstacle.

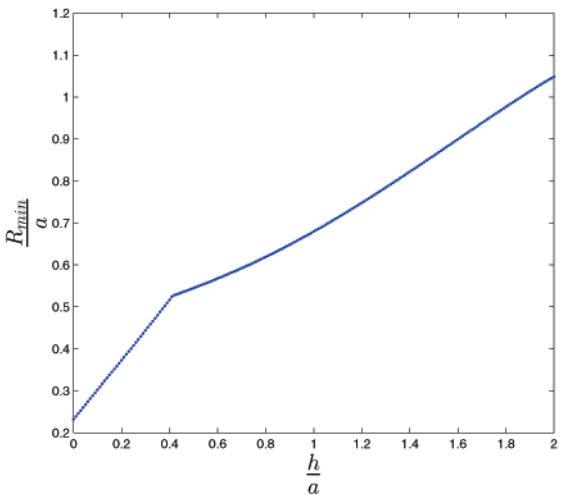


Figure 6. Minimum pathline radius of curvature for different wedge lengths (δ = 0·1).

steady flight, and are therefore somewhat conservative. In an emergency, tighter turns could likely be executed for short periods of time (a scenario that would arise if the obstacle was detected late).

In order that the obstacle avoidance manoeuvres be feasible, it is important that the pathlines of the velocity field do not have radii of curvature which are smaller than the aircraft's minimum turning radius. To accomplish this, wedges with lengths h are added at the front and back ends of each modelled obstacle, as shown in Fig. 5(a). In addition, as shown in Fig. 5(a), a δ-region is defined about the modelled obstacle's axis of symmetry. As shown in Fig. 5(b), if the aircraft's nominal flight-path lies within the δ-region, then the modelled obstacle is enlarged to one side such that the nominal flight path lies outside the δ-region. The size of the δ-region is a user-defined parameter.

To guide the selection of wedge length, h, for a given obstacle, the minimum radius of curvature can be found across all pathlines starting outside the δ region of an obstacle. This is a procedure that must be performed numerically. Given that obstacles are initially assumed to have the same depth as width (as explained in Section 3.1), this procedure is performed for square obstacles. As shown in Section 2.3, pathlines scale geometrically with obstacle size, so it is sufficient to consider a unit square obstacle (effectively non-dimensionalising the results by obstacle width, a). Figure 6 shows the results for a square obstacle with δ region size selected to be δ = 0·1.

The results shown in Fig. 6 can be stored on-board as a look-up table to determine the required

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

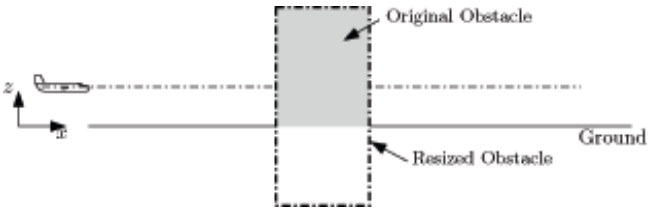


Figure 7. Resizing an obstacle to guarantee flying over.

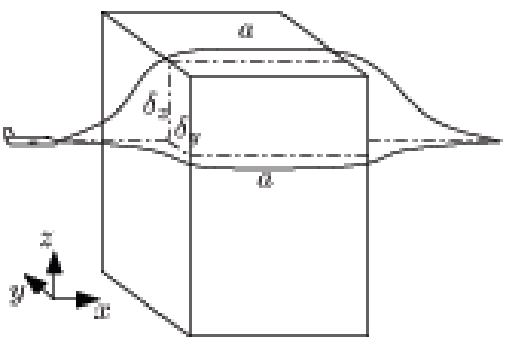


Figure 8. Projection of avoidance manoeuver on obstacle.

wedge-length for a given obstacle, given constraints on the minimum turning or pull-up radii as given in Equations (26) and (29), respectively.

An additional consideration is required for fly-over manoeuvres. In this case, it must be ensured that the aircraft flies over an obstacle, and not under it. To accomplish this, modelled obstacles are enlarged as shown in Fig. 7 such that the aircraft is above the obstacle's axis of symmetry.

As mentioned earlier, either a fly around or fly-over manoeuvre can be performed. A criterion that can be used to select is simply the shortest expected path that the aircraft will follow. Figure 8 shows the projections of fly-around and fly-over manoeuvres onto a modelled obstacle. Based on the projections, a simple condition for selecting fly-around or fly-over manoeuvres is given as the shorted projected path. Namely,

$$a + 2\delta_y < a + 2\delta_z \quad \ldots (30)$$

If the condition in Equation (30) is satisfied, then a fly-around manoeuvre is performed. Otherwise a fly-over manoeuvre is performed.

3.3 Obstacle consideration and sink location and strength

To avoid undue deviations from the nominal flight path, a decision needs to be made as to whether an avoidance manoeuvre is needed for a given obstacle, or if it can be ignored. There are two factors that are used to make this decision as shown in Fig. 9. First, an avoidance manoeuvre is considered only if the nominal flight path intersects the modelled obstacle. Second, if an avoidance manoeuvre is needed, it is activated once the aircraft passes the activation line, located at a distance $d = \gamma_d a$ ahead of the modelled obstacle (as shown in Fig. 9), where $\gamma_d > 0$ is a user-specified parameter.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

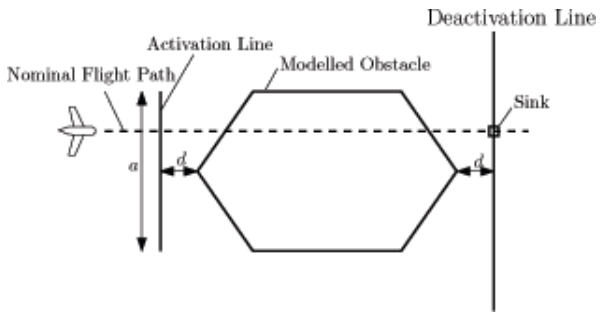


Figure 9. Obstacle avoidance activation and sink location.

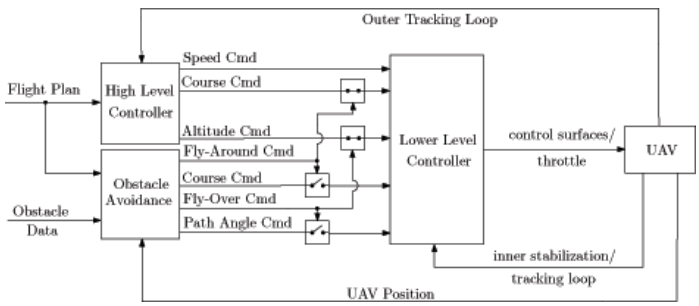


Figure 10. Integration of obstacle avoidance with autopilot.

The avoidance manoeuvre is deactivated once the aircraft crosses the deactivation line, located at a distance $d = \gamma_d a$ behind the modelled obstacle.

As discussed in Section 2.2, a sink is added to draw the fluid pathlines (and hence the aircraft) back to the nominal flight path. Therefore, the sink is placed at the intersection of the nominal flight path and the deactivation line, as shown in Fig. 9. However, the sink adds a singularity to the resulting velocity field. To ensure that this singularity is not encountered during the avoidance manoeuvre, in practise the sink should be located on the nominal flight path, slightly beyond the deactivation line.

The geometric sink strength, ℓ, introduced in Section 2.2 determines the cross-section of flow upstream from the obstacle, whose pathlines end at the sink. As such, the geometric sink strength is taken to be $\ell = \gamma_\ell a$, where $\gamma_\ell > 0$ is a user specified parameter. That is, the geometric sink strength is taken to be proportional to the obstacle width.

3.4 Integration with flight control laws

The obstacle avoidance method detailed in the previous sections is designed to be integrated with a flight control architecture as shown in Fig. 10. Nominally, the flight control architecture consists of an inner and outer loop. In the outer loop, a high level controller takes as input the nominal flight path from the mission plan as well as the measured aircraft position and velocity, and outputs airspeed, altitude and course (the direction of travel in the horizontal plane from a reference direction) commands, which are sent to the low level controller in the inner loop. The inner loop is responsible for stabilising the aircraft, and tracking the airspeed, altitude and course

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

Table 1
Aerosonde UAV specifications

Weight	27-30lb (12·2-13·6kg)
Length	6ft (1·8m)
Height	2ft (0·6m)
Wing span	10ft (3·0m)
Wing area	6ft (0·6m)
Engine	24cc, 1·6hp (1·2kW)
Cruise Speed	51mph (82·1kph)
Range	2,044miles (3,289·5km)
Altitude range	Up to 20,000ft (6,096km)
Payload	Maximum 5lb (2·3kg) with full fuel

commands, as well as possibly an additional path angle (the angle of the flight path above the horizontal plane) command from the obstacle avoidance algorithm. As shown in Fig. 10, the obstacle avoidance algorithm is implemented in parallel to the high level controller, where it can override outputs of the high level controller depending on whether or not obstacle avoidance is required. Specifically, if a fly-around manoeuvre is required, the course command from the obstacle avoidance algorithm overrides the course command from the high level controller. If a fly-over manoeuvre is required, the altitude command from the high level controller is disabled, and is replaced by a path angle command from the obstacle avoidance algorithm. The obstacle avoidance algorithm does not affect the airspeed command.

4.0 NUMERICAL DEMONSTRATION

For the purposes of numerical demonstration of the proposed obstacle avoidance algorithm, a full non-linear six-degree of freedom UAV simulation was created in the MATLAB Simulink framework. The Aerosonde UAV model from within the AeroSim aeronautical simulation blockset (developed by Unmanned Dynamics) was used as the aircraft model, and custom flight control laws were created, capable of tracking course and path-angle commands, as shown in Fig. 10. Full details of the flight control laws may be found in Ref. 33.

The Aerosonde UAV is a small UAV developed for weather-reconnaissance and remote-sensing missions, and it was the first UAV to cross the north-Atlantic in 1998[32,40]. A summary of the aircraft's specifications are listed in Table 1[23,32].

For the purposes of the obstacle avoidance manoeuvres to be demonstrated here, the Aerosonde is set to fly at a cruise speed of 23ms^{-1} (approximately 51mph as in Table 1). Furthermore, in obstacle avoidance manoeuvres, its bank angle is limited to 60 degrees. Note that for the cruise speed mentioned, the aircraft was tested in simulation and was able to maintain altitude while keeping a 60 degree bank angle. Therefore, from Equation (28)

$$n = 2$$

and from Equation (26) the minimum turning radius is

$$R_{min,T} = 31{\cdot}13\text{m} \quad \ldots (31)$$

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

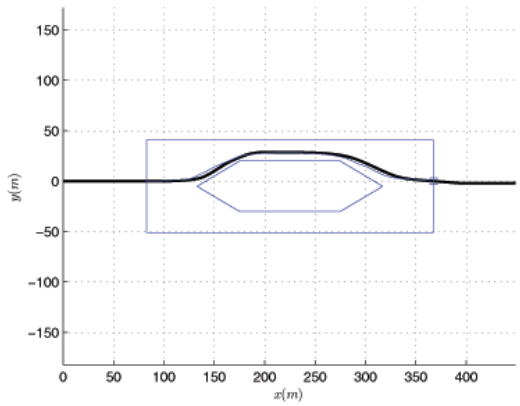


Figure 11. Fly-around avoidance manoeuver.

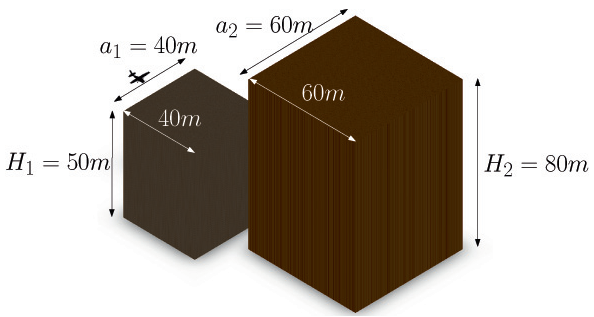


Figure 12. Two obstacle scenario.

while from (29), the minimum pull-up radius is

$$R_{min,P} = 53{\cdot}92\text{m} \quad \ldots (32)$$

The numerical parameters used in the obstacle avoidance algorithm for the activation/deactivation area and geometric sink strength are $\gamma_d = \gamma_\ell = 1$, respectively.

Figure 11 shows a fly around manoeuvre for an obstacle of width a = 50m and depth $2a$ = 100m. The flight path in the figure is from left to right. It is initially assumed that the obstacle has equal depth and width as in Fig. 4. However, at x = 200m, the obstacle's real depth of $2a$ = 100m is discovered and the obstacle avoidance velocity field is recalculated. In Fig. 11, both the fluid pathline (thin blue line) and actual UAV (thick black line) trajectory are shown. The left and right sides of the bounding box represent the activation and deactivation lines respectively, and the sink location is shown by the small square on the deactivation line. It can be seen that the fluid path line is followed quite closely. Note that even though the obstacle's dimension changes while an avoiding manoeuvre is in process, the avoiding manoeuvre corrects itself and successfully avoids

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

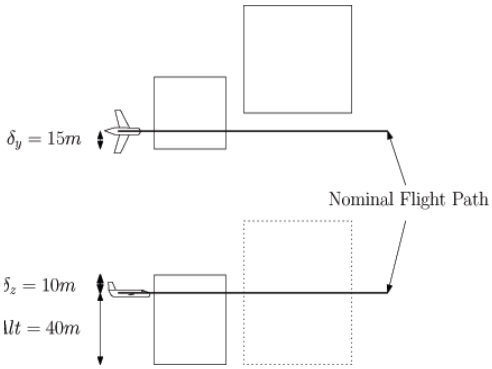


Figure 13: Two obstacle scenario — Flight Case I.

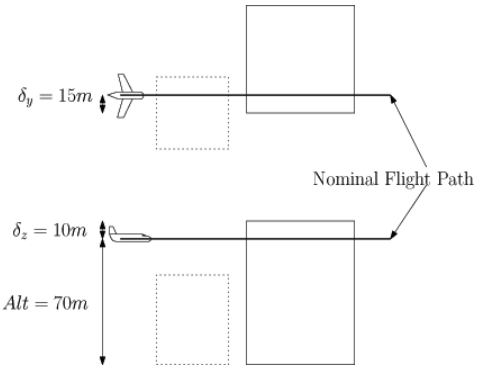


Figure 14. Two obstacle scenario — Flight Case II.

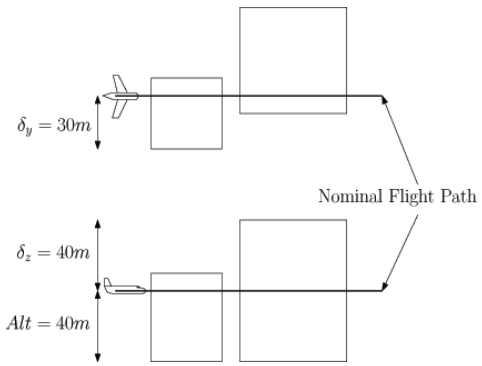


Figure 15. Two obstacle scenario — Flight case III.

the obstacle, returning the UAV to the nominal flight path. Some difference between the actual UAV trajectory and the fluid pathline is to be expected. As demonstrated in Section 2.2, if the UAV's velocity vector is always parallel to the local flow field velocity vector, the UAV trajectory will coincide with the fluid pathline. However, given that the UAV autopilot only has access to the direction of the local fluid flow velocity vector (which depends on the UAV's instantaneous position), the difference between the pathline and the UAV trajectory connot be eliminated, and depends on the speed of response of the autopilot (the faster the autopilot response, the closer the UAV trajectory to the fluid pathline).

The next demonstration includes a pair of three-dimensional obstacles as shown in Fig. 12. Three different nominal flight paths are considered as shown in Figs 13, 14 and 15. Figure 16 shows the resulting avoidance manoeuvres. In Case I, a fly-over manoeuvre is chosen for the first obstacle, while the second obstacle is ignored. In Case II, a fly-over manoeuvre is chosen for the second obstacle, while the first obstacle is ignored. In Case III, both obstacles are combined into a single obstacle and a fly-around manoeuvre is commanded.

In all three cases, the UAV successfully avoids the obstacles, and returns to the nominal flight path.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

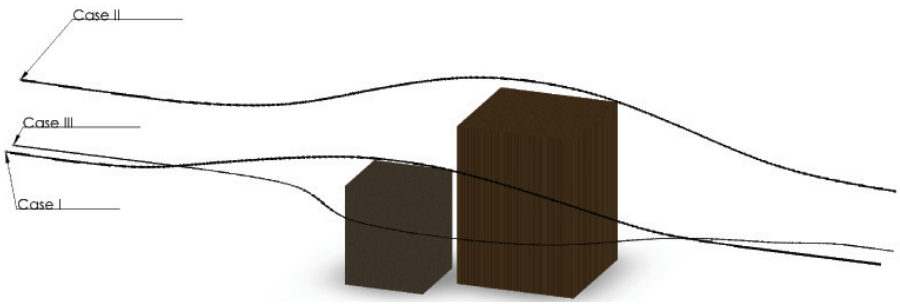


Figure 16. OA manoeuvers in a scenario with two obstacles for three different cases.

5.0 CONCLUDING REMARKS

A method for autonomous obstacle avoidance for fixed-wing UAVs, utilising potential fluid flow theory has been investigated, with the idea that fluid flow pathlines around an obstacle represent a suitable trajectory for a UAV to avoid it. It has been rigorously demonstrated that a pathline is completely determined by the direction of the potential flow velocity field, independent of its magnitude (which could even be time-varying). This justifies using the direction of the instantaneous local potential velocity vector as a commanded flight direction for the aircraft. In particular, if the aircraft's flight control laws provide perfect tracking, the flight path will coincide exactly with a fluid pathline. Thus, this method is highly suited to real-time on-board implementation, since at any point in time, the obstacle avoidance algorithm needs only compute the instantaneous local potential velocity vector. In particular, new obstacle information is readily accommodated in the obstacle avoidance scheme, making it a truly reactive approach (as opposed to an approach based on path planning).

In order to maintain computational simplicity, two-dimensional potential flow is utilised, and obstacles are either modelled in the horizontal or vertical planes depending on whether a fly-around or fly-over manoeuvre is required. To accommodate UAV manoeuvring constraints on turning or pull-up radii, obstacles are approximated by bounding rectangles, with wedges added to their front and back to shape the resulting fluid pathlines. It has been shown in this paper that the resulting potential flow velocity field is completely determined by the obstacle field geometry. This allows one to determine a non-dimensional relationship between obstacle added wedge-length and the corresponding minimum pathline radius of curvature, which can then be readily scaled in on-board implementation.

The efficacy of the proposed approach has been demonstrated numerically with an Aerosonde UAV model.

Future work will include the investigation of full three dimensional obstacle modelling, removing the need for decision making between a fly-around and fly-over manoeuvre.

ACKNOWLEDGEMENTS

This work was supported in part by Sander Geophysics Ltd and in part by the Natural Sciences and Engineering Research Council of Canada.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

REFERENCES

1. Amin, J., Boskovic, J. and Mehra, R. A fast and efficient approach to path planning for unmanned vehicles, 2006, AIAA Guidance, Navigation, and Control Conference and Exhibition, Keystone, Colorado, USA, AIAA 2006-6103.
2. Anderson, J.D. *Introduction to Flight*, Third edition, 1989, McGraw-Hill, New York, USA.
3. Anderson, J.D. *Fundamentals of Aerodynamics*, Fourth edition, 2007, McGraw-Hill, Boston, MA, USA.
4. Barnard, J.A. The use of unmanned aircraft in oil, gas and mineral E+P activities, 2008, Society of Exploration Geophysicists, p 1132, Las Vegas, Nevada, USA.
5. Barnard, J.A. Use of unmanned air vehicles in oil, gas and mineral exploration activities, 2010, AUVSI Unmanned Systems North America.
6. Caron, R.M. Aeromagnetic Surveying Using a Simulated Unmanned Aircraft System, Master's thesis, 2011, Carleton University.
7. Casbeer, D.W., Li, S.-M., Beard, R.W., Mehra, R.K. and McLain, T.W. Forest fire monitoring with multiple small UAVs, 2005 American Control Conference, IEEE, pp 3530-3535.
8. Chandler, P., Rasmussen, S. and Pachter, M. UAV cooperative path planning, 2000, AIAA Guidance, Navigation and Control Conference, American Institute of Aeronautics and Astronautics.
9. Chapman, B.L. and Perreira, N. Algorithm for intelligent control of a robot manipulator, 1983, International MOTORCON Conference, Orlando, FL, USA, Intertec Communications, pp 334-344.
10. De Biasio, M., Arnold, T., Leitner, R. and Meestert, R. UAV based multi-spectral imaging system for environmental monitoring, 2011, 20th IMEKO TC2 Symposium on Photonics in Measurement, 20th ISPM 2011, Shaker Verlag, pp 69-73.
11. Doty, K.L. and Govindaraj, S. Robot obstacle detection and avoidance determined by actuator torques and joint positions, 1982, IIEEE Southeastcon '82, IEEE, pp 470-473.
12. Fahimi, F., Ashrafiuon, H. and Nataraj, C. Obstacle avoidance for spatial hyper-redundant manipulators using harmonic potential functions and the mode shape technique, *J Robotic Systems,* 2003, **20**, (1), pp 23-33.
13. Fahimi, F., Nataraj, C. and Ashrafiuon, H. Real-time obstacle avoidance for multiple mobile robots, *Robotica*, 2008, **27**, (2), pp 189-198.
14. Ge, S. and Cui, Y. New potential functions for mobile robot path planning, *IEEE Transactions on Robotics and Automation*, 2000, **16**, (5), pp 615-620.
15. Ge, S. and Cui, Y. Dynamic motion planning for mobile robots using potential field method, *Autonomous Robots*, 2002, **13**, (3), pp 207-222.
16. Girard, A.R., Howell, A.S. and Hedrick, J.K. Border patrol and surveillance missions using multiple unmanned air vehicles, 2004, 43rd IEEE Conference on Decision and Control (CDC), **1**, IEEE, pp 620-625.
17. Hausamann, D., Zirnig, W., Schreier, G. and Strobl, P. Monitoring of gas pipelines — a civil UAV application, *Aircraft Eng and Aerospace Tech*, 2005, **77**, (5), pp 352-360.
18. Hwangbo, M., Kuffner, J. and Kanade, T. Efficient two-phase 3D motion planning for small fixed-wing UAVs, 2007, IEEE International Conference on Robotics and Automation, Rome, Italy, pp 1035-1041.
19. Jones, G.P., Pearlstine, L.G. and Percival, H.F. An assessment of small unmanned aerial vehicles for wildlife research, *Wildlife Society Bulletin*, 2006, **34**, (3), pp 750-758.
20. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots, *Int J of Robotics Research*, March 1986, **5**, (1), pp 90-98.
21. Khosla, P. and Volpe, R. Superquadric artificial potentials for obstacle avoidance and approach, 1988 IEEE International Conference on Robotics and Automation, IEEE Comput Soc Press, pp 1778-1784.
22. Kim, J.-O. and Khosla, P. Real-time obstacle avoidance using harmonic potential functions, *IEEE Transactions on Robotics and Automation*, 1992, **8**, (3), pp 338-349.
23. Kurnaz, S., Cetin, O. and Kaynak, O. Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles, *J Intelligent and Robotic Systems,* Oct 2008 **54**, (1-3), pp 229-244.
24. LaValle, S. and Kuffner, J. Randomized kinodynamic planning, *Int J of Robotics Research*, 2001, **20**, (5), pp 378-400.
25. Lavalle, S.M. Rapidly-exploring random trees: A new tool for path planning, 1998, Tech Report, Department of Computer Science, Iowa State University, Ames, IA, USA.
26. LaValle, S.M. *Planning Algorithms*, 2006, Cambridge University Press, UK.
27. Li, X. and Yang, L. Design and implementation of UAV intelligent aerial photography system BT, 2012, Foutth International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMS, **2**, IEEE Computer Society, pp 200-203.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press

28. LOEFF, L.A. and SONI, A.H. An algorithm for computer guidance of a manipulator in between obstacles, Transactions of the ASME, Series B, *J Eng for Industry,* 1975, **97**, (3), pp 836-842.
29. MARCE, L., JULLIERE, M. and PLACE, H. Strategy of obstacle avoidance for a mobile robot, *RAIRO Automatique*, 1981, **15**, (1), pp 5-18.
30. MCINNES, C. Velocity field path-planning for single and multiple unmanned aerial vehicles, *Aeronaut J*, 2003, **107**, pp 419-426.
31. MUJUMDAR, A. and PADHI, R. Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles, *J Aerospace Computing*, *Information, and Communication,* 2011, **8**, pp 17-41.
32. Museum of Flight. Insitu Aerosonde Laima, http://www.museumofflight.org/aircraft/insitu-areosonde-laima, accessed: 12 November 2012.
33. OWLIA, S. Real-time Autonomous Obstacle Avoidance for Low-altitude Fixed-wing Aircraft, Master's thesis, 2013, Carleton University.
34. RICHALET, J. Industrial applications of model based predictive control, *Automatica*, 1993, **29**, (5), pp 1251-1274.
35. SAUNDERS, J., CALL, B. and CURTIS, A. Static and dynamic obstacle avoidance in miniature air vehicles, 2005, Infotech@Aerospace, Arlington, VA, USA, *AIAA*, 2005, pp 2005-6950.
36. SHAMES, I.H. *Mechanics of Fluids*, Second edition, 1982, McGraw-Hill, New York, USA.
37. SHIM, D., CHUNG, H., KIM, H. and SASTRY, S. Autonomous exploration in unknown urban environments for unmanned aerial vehicles, 2005, AIAA GN&C Conference.
38. SHIM, D. and SASTRY, S. An evasive maneuvering algorithm for UAVs in see-and-avoid situations, 2007 American Control Conference, New York City, USA, *IEEE*, July 2007, pp 3886-3891.
39. TSOURDOS, A., WHITE, B. and SHANMUGAVEL, M. Cooperative path planning of unmanned aerial vehicles, 2011, American Institute of Aeronautics and Astronautics, Wiley, West Sussex, UK.
40. Unmanned dynamics, AeroSim aeronautical simulation blockset Version 1.2 Users's Guide. Hood River, OR, USA.
41. WAYDO, S. and MURRAY, R. Vehicle motion planning using stream functions, 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, *IEEE*, 2003, pp 2484-2491.

https://doi.org/10.1017/S0001924000011325 Published online by Cambridge University Press