Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

Title: Evaluation of UAS Separation Maneuvers and their Automated Execution

Author: Xavier Feixas Güell

Directors: Enric Pastor Llorens, Marc Pérez Batlle

Date: 13 July 2012

Title: Evaluation of UAS Separation Maneuvers and their Automated Execution

Author: Xavier Feixas Güell

Directors: Enric Pastor Llorens, Marc Pérez Batlle

Date: 13 July 2012

Abstract

This project is focussed on the evaluation of the separation maneuvers done by the Unmanned Aerial Systems. For this evaluation the behaviour of variables such as speed of the UAS, heading changes done by the UAS and the different times to conflict are analysed. The evolution of the separation distance with these variables will be plotted. The project will focus on the oblique maneuver, for the forward case and the backward case. The maneuver is described and the ranges of angles that will define it are computed. This angles determine if the maneuver can be considered oblique or not, which means that if the conflict geometry is over the maximum or below the minimum angle the maneuver will not be considered as oblique. Finally an interpolation is done in order to determine which speeds, change of heading and time to conflict are necessary in order to execute the maneuver. Then the dependence with the minimum distance of every variable is explained. All these analysis are done using Matlab where different functions are used to compute all the distances, and this way its evolution with the other variables can be described.

ÍNDEX

INDEX OF ACRONYMS

UAS: Unmanned Aerial System.

d_{min}: Minimum distance achieved by the UAS when the separation maneuver is executed.

β: Relative angle between the UAS and the intruder.

Δh: Change of heading done by the UAS in order to perform the separation maneuver.

d_{req}: Required distance, desired distance that the UAS will have to achieve to perform the separation maneuver.

t_c: Time to conflict.

t_{extra}: Extra time the UAS will have to wait in order to execute the separation maneuver

d_c: Distance to conflict.

ICARUS: Research group that aims to improve the Air Transportation efficiency and the development of ground systems and on-board avionics for UAS.

ATC: Airt Traffic Control

INTRODUCTION

Nowadays the interest on using Unmanned Aerial Systems (UAS) in civil applications has increased. But the lack of regulation basis as its certification, its airworthiness and operations ban them from non-segregated airspace [1]. Their missions would normally require non-conventional flight plans, they will not fly from point to point, they will possibly make scan flights, perimeter loops, etc. These UAS will work at a similar altitude than the conventional airliners, but they will have poorer performance. When this happens there is a possibility of having a collision conflict [2]. So at this point two functionalities appear: separation assurance and collision avoidance. Separation assurance aims at keeping minimum distances between aircraft and the potential intruders, and collision avoidance can prevent an imminent collision in case of a loss of separation as a last resort maneuver [3]. For this purpose separation maneuvers have to be analysed.

This is project focused on the analysis of the separation maneuvers, supposing a possible collision with an airliner, which is a faster plane than the UAS [2]. So the objective of this project is the analysis of the behaviour of the different variables that must be taken into account in a separation maneuver and the characterization of the oblique maneuver.

For this purpose in the first chapter the geometry of the conflict, and the geometry of the maneuver is explained. There are also introduced the variables that must be taken into account in order to realize the maneuver. The information used on this chapter is from two papers written by the ICARUS group.

The second chapter is focused on distance calculations. Two calculations are done: the minimum distance, and the angle related to a given minimum distance. In order to make this calculus Matlab is used, that will simulate several situations obtaining the resulting plots describing every one. The previous work of this project was a distance calculus and angle calculus for only one case of speed of the UAS, so in this project the distance is computed for a range of UAS speeds.

The third chapter focuses on the oblique maneuver. The minimum and the maximum angle that allow the realisation of this type of maneuver are calculated. Finally the time that the UAS takes to arrive at the point where the separation must start has been computed. In order to obtain this time and the angles Matlab is used.

In the last chapter an interpolation is done. This allows determining the required speeds and angles for a given value of the separation needed. This would be useful because the UAS can be informed of the speed and the angle that must be used in a situation of a collision conflict. Finally the influence of every variable in the separation maneuver is showed and it is possible to know which is the variable that most affects the separation. These calculations are also done using Matlab.

CHAPTER 1. DESCRIPTION OF THE GEOMETRY

In order to evaluate the separation conflict it is very important to analyse its geometry, taking into account the differences in the performance of the aircraft, for example: in a conflict between the UAS and an airliner, the airliner has a higher speed than the UAS. One of the possible solutions for the conflict is changing the flight level of the UAS, but due to its poor climbing or descending performance it makes this solution not possible. A better solution would be a change of the heading of the UAS or the airliner. If the UAS makes the heading change it has to be executed well in advance if its speed is slower than the speed of the intruder.

1.1. General geometry

The conflict can be described with the scheme on figure 1.1:

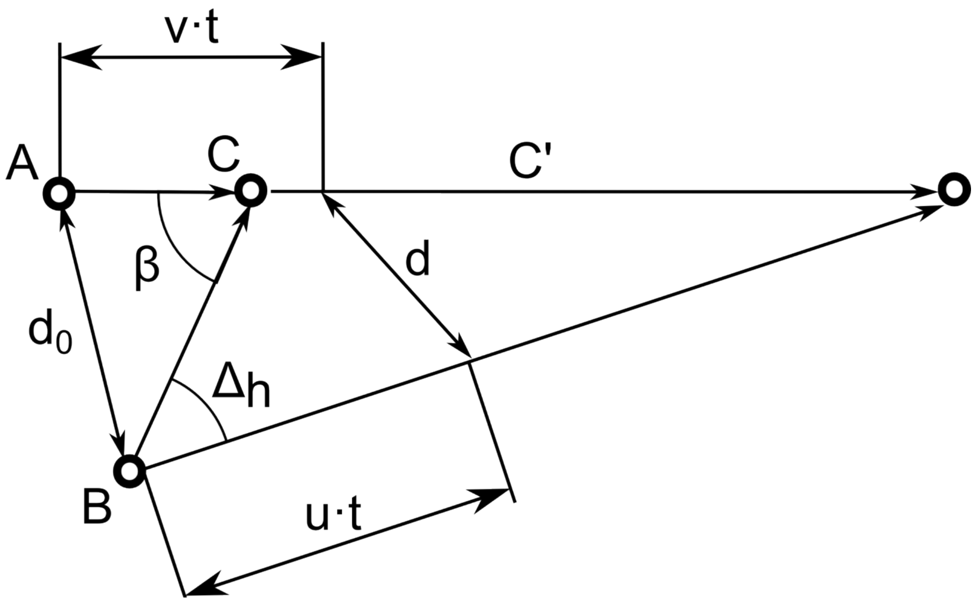


Figure 1.1: Geometry of the conflict

The points A and B represent both aircraft, where A is the airliner and B is the UAS. Both A and B are flying at the same altitude and at constant speed, A is flying at constant speed v and B at constant speed u. Both aircraft are flying to the point C and they will arrive at the same time, so C is the collision point. β is the angle between the heading of aircraft A and aircraft B, and depending on this angle the separation maneuver will change. In order to avoid the collision aircraft B, the UAS, will change its heading. This change of heading is represented by Δh and maintains a minimum separation represented by d. The

time to conflict (t_c) must be taken into account, because it will determine the point where A and B are placed. It also determines where B has to start its maneuver for the purpose to avoid the collision at point C.
We want to evaluate the dependence of d_{sep}, the separation distance, with Δh, v, u, β and t_c , and find a solution that best fits to the minimum distance we need between both aircraft [2].

1.2. Oblique Maneuver

The simplest separation maneuver will be the oblique maneuver, and it is explained below:

When the conflict is detected at a distance d_c UAS will change its heading to an equivalent angle to β. This will provide a parallel track to the heading of the aircraft A, the airliner, at a distance d_{min}. When the conflict finishes, the UAS will turn back to the initial heading.
The procedure explained before is represented in figure 1.2:

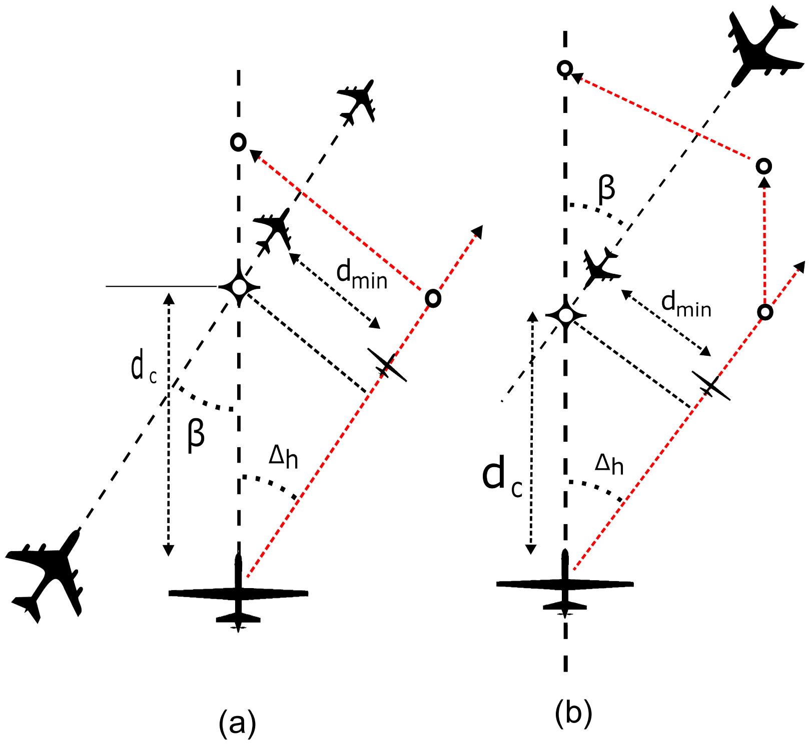


Figure 1.2: Separation maneuver.
(a) Backard oblique maneuver.
(b) Forward oblique maneuver.

As showed on figure 1.2 once the conflict is cleared the initial trajectory is recovered in both cases.

There is the possibility that when the conflict is detected the UAS is far from the point where the maneuver must be done. If the UAS performs the maneuver at this point the resultant separation distance will be higher than the required one. So at this moment the UAS will advance to the next point in order to guarantee the distance we wanted at first. So the detection time will not be at the same time the UAS executes the separation maneuver, there is going to be an added time that the UAS spends going to the next point to start the maneuver.

Figure 1.3 shows the explained before:

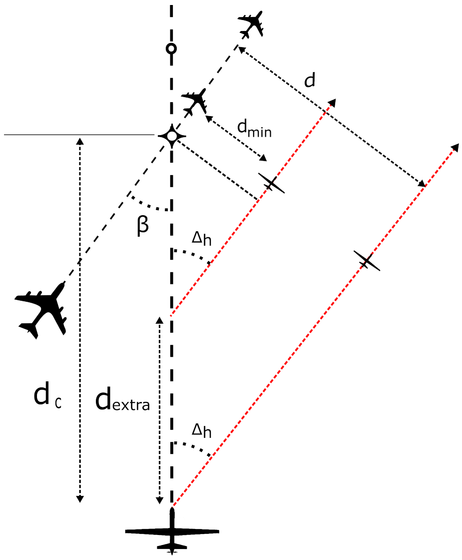


Figure 1.3: Separation maneuver when the conflict is detected before the the minimum distance

The point where the UAS is situated is where the conflict is detected. The resulting distance if the aircraft makes the separation maneuver will be *d*. If the minimum distance d_{min} must be maintained, the UAS must advance a distance d_{extra} where the maneuver must be done.

After evaluating all the angles, some similar maneuvers can be determined for a specific range of angles.

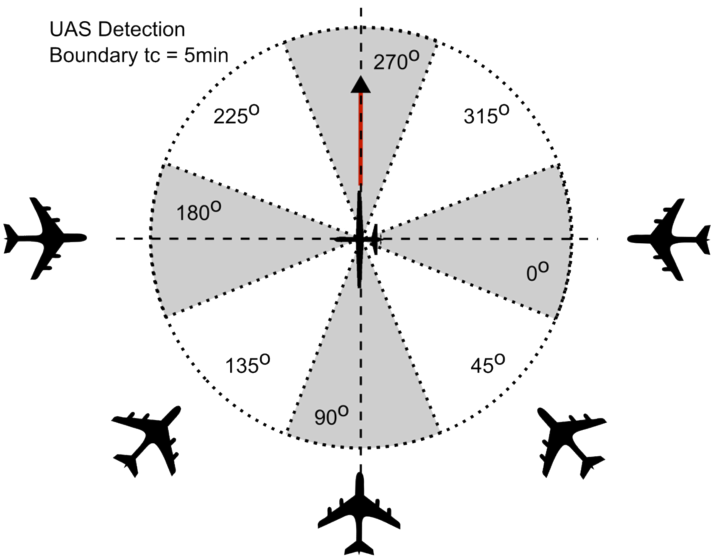


Figure 1.4: Angles range where the same maneuver is going to be applied

For example, for angles similar to 0º the same type of separation maneuver will be done, or for angles similar to 45º a different type of maneuver will be done. So we could differ the type of maneuver for the situation when the intruder comes from different headings [4].

CHAPTER 2. DISTANCE CALCULATIONS

There are different ways to evaluate the geometries described in the previous chapter. The evaluation in this project is done using Matlab and simulating various situations taking into account the performance of both planes involved in the conflict.

2.1. Angle (β) VS. Minimum distance (d_{min})

The first step on these calculations is to compute the angle that can be achieved in the maneuver for a given speed and for a given d_{min}. This calculus is useful in order to see if d_{min} can be achieved with a given β, which is equal to Δh. This calculus will be done according to the geometry on figure 1.2.

2.1.1. UAS at constant speed

For this purpose a Matlab code is used to compute the angle to guarantee a d_{min} for a given speed and a range of times to conflict. This would be useful in order to check if the angle that the UAS will use in order to avoid the collision will achieve a secure separation. This code plots the results as in figure 2.1:

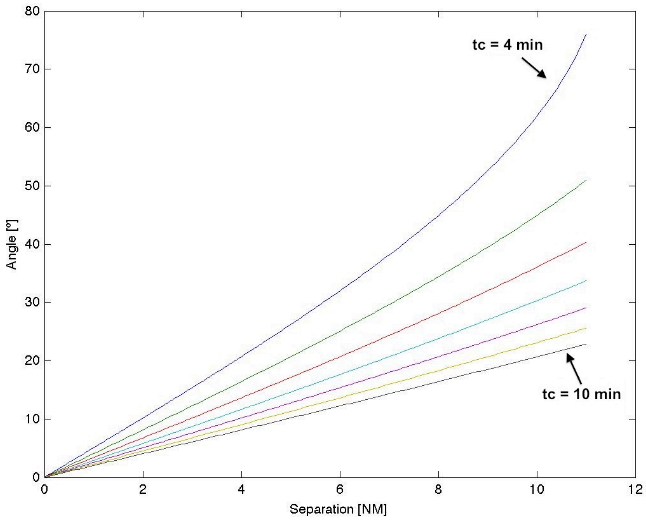


Figure 2.1: Result of the angle as a funtion of distance for a speed of 170 kt. The diferent lines represent the diferent times to conflict

The different lines represent the different times to conflict, from 4 minutes to 10 minutes by steps of 1 minute. The x-axis is d_{min} in NM and the y-axis is the angle β in degrees. So the angles that the UAS can achieve will depend on the time to conflict. So for a selected time to conflict, for example 5 minutes, all the angles will correspond to the ones on the 5 minutes line.

2.1.2. Speed of the UAS as a variable

Apart from the time to conflict, speed is a variable that must be taken into account. So on the next paragraphs speed is going to be treated as a variable and not like a constant. This way several situations with different speeds can be evaluated.

The function mentioned before has been adapted in order to compute β for different speeds. The geometry used is on figure 1.2.

This code calculates the angles for:

- Time to conflict (*tc*): from 240 seconds to 600 seconds.
- Speed of the UAS (*v1*): from 120 kt to 300 kt. These are the speeds of a slow UAS like the General Atomics Predator and the speed of the fastest UAS like the Global Hawk. This way all the possibilities can be showed, but for a specific model of UAS the maximum and minimum speeds can be determined by the performance of the UAS.
- Minimum separation distance (*dmin*): from 1 NM to 10 NM.

The code used to compute the angle is showed on figure 2.2:

```
function [] = ls16_vector()

% Calculations of beta

tc = linspace(240,600, 10);
dmin = linspace(0, 11, 10);
v1=linspace(120,300, 10);
dc=zeros(length(tc),length(v1));

for i=1: 1: length(tc)
    for j=1: 1: length(v1)
        dc(i,j)=v1(j)*tc(i)/3600;
    end
end

beta = zeros(length(tc), length(dmin), length(v1));

for i = 1: 1:length(tc)
    for j=1:1:length(dmin)
        for k=1: 1: length(v1)
            if dc(i,k)>=dmin(j)
                beta(i,:,k) = asin(dmin/dc(i,k));
            else
                beta(i,j,k)=pi/2;
            end
        end
    end
end
beta = 180/pi*alpha;
```

```
% Plot

hold off

for k=1: 1: length(dmin)
    dmin1=linspace(dmin(k), dmin(k), length(beta));
    plot3(v1(:), dmin1(:), beta(:,:,k));
    hold on
end

ylabel('Separation [NM]');
zlabel('Beta [º]');
xlabel('Speed [kt]');
```

Figure 2.2: Code used to compute β

The angle defined as `beta` corresponds to Δh. It computes `beta` using `dmin` and `dc`, where `dc` is the distance to the conflict at a time `tc`. There are some cases when the result of `beta` is a complex number. This means that `dmin` is higher than `dc`, so this maneuver cannot be realized. To solve this, the function adds a `pi/2` to the corresponding position of the matrix `beta`. This way the unfeasible maneuvers because of the geometry can be identified if the value of `beta` is 90º.

The function plots the angle as a function of the speed and the time to conflict for a given value of d_{min} and this way the influence of the speed on β and the separation can be analysed. The result of this is plotted on figure 2.3:

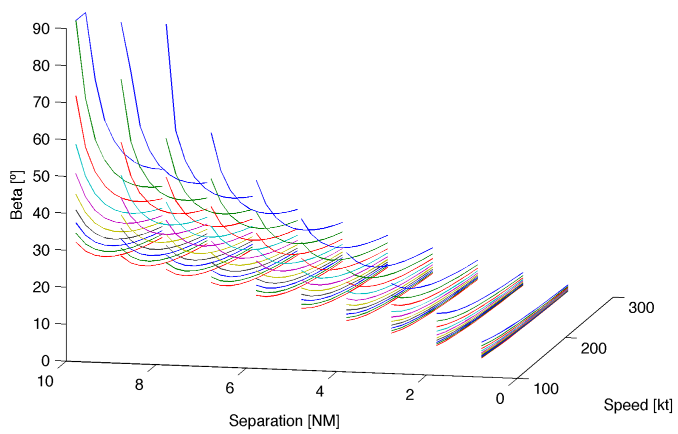


Figure 2.3: Minfimum separation related to β and t_c

As it is a bit difficult to take information from this plot, values of 10 NM, 5 NM, 3 NM and 1 NM are taken in order to evaluate them separately.

First 10 NM case is going to be analysed. Figure 2.4 shows the angle as a function of the speed for 10 NM for a range of times from 4 minutes to 10 minutes with steps of 20 seconds:

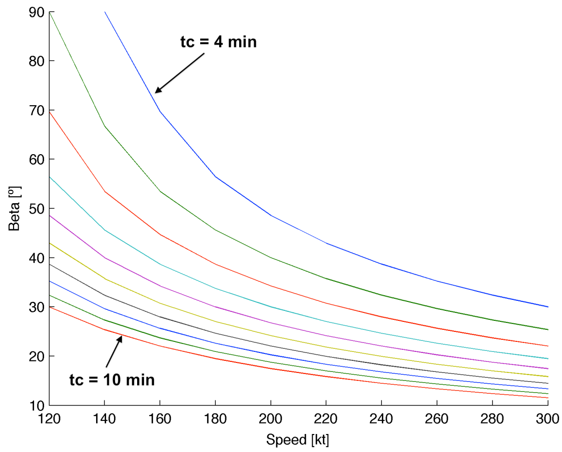


Figure 2.4: Angle as a function of speed for 10 NM. The different lines represent the times to conflict.

These times to conflict will be used in all the cases that are studied.

As the figure shows, for a given d_{min} of 10 NM, when the speed is increased the angle β is decreased. There are also some cases where the separation of 10 NM is not possible. These cases are showed when the angle β is equal to 90º, for example for a time to conflict of 4 minutes and speeds of 120 kt and 140 kt. So the extreme cases will be analysed:

10 NM	Speed	Time to conflict	β
	120 kt	4 minutes	90º
		10 minutes	30º
	300 kt	4 minutes	30º
		10 minutes	11.54º

Table 2.1: β for 10 NM when the extreme cases are analysed.

The angle is showed on figure 2.5 in order to remember its meaning. The angle computed is the angle β, which is the same as Δh.

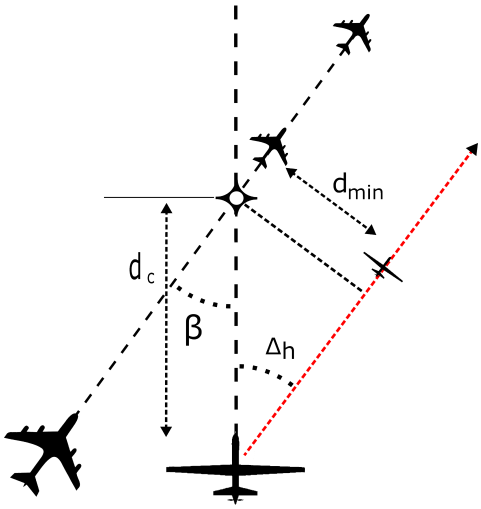


Figure 2.5: Geometry of the calculus

In order to evaluate the dependence of the angle with d_{min}, a 5 NM d_{min} has been analysed.For the case of 5 NM the plot is showed on figure 2.6:

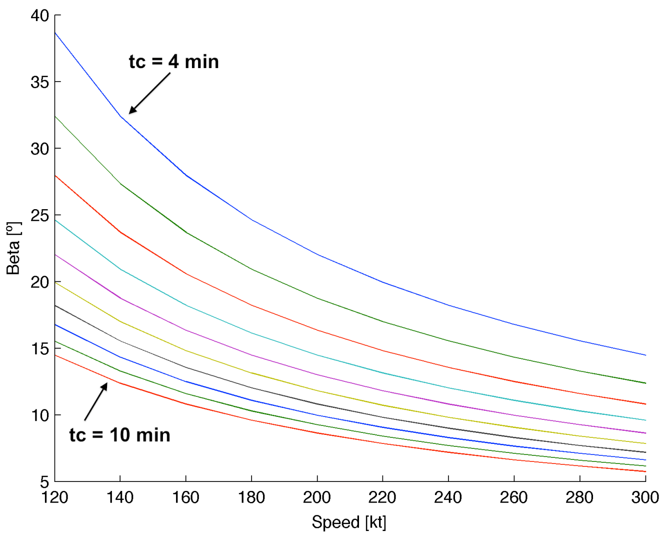


Figure 2.6: β as a function of speed for 5 NM The different lines represent the times to conflict.

The same cases are analysed:

5 NM		Speed	Time to conflict	β
	120 kt	4 minutes	38.68º	38.68º
		10 minutes	14.48º	14.48º
	300 kt	4 minutes	14.48º	14.48º
		10 minutes	5.739º	5.739º

Table 2.1: β for 5 NM when the extreme cases are analysed.

The next case that will be analysed is a d_{min} of 3 NM in order to show its behaviour .Figure 2.7 shows the plot for a minimum distance of 3 NM:

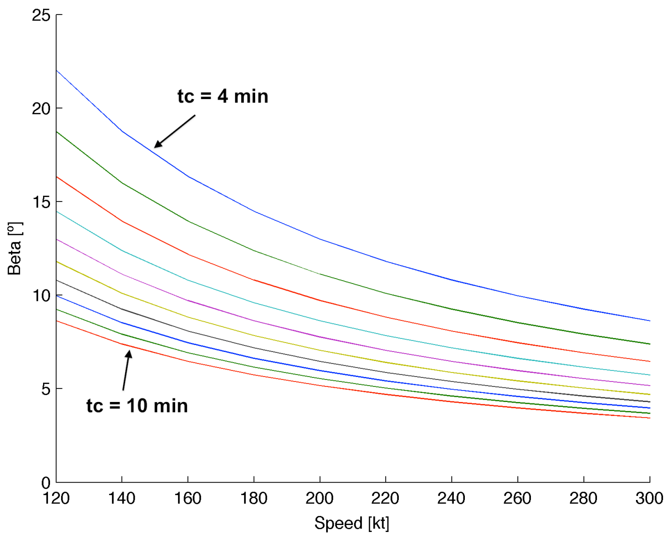


Figure 2.7: β as a function of speed for 3 NM. The different lines represent the times to conflict.

For the case of 3 NM the analysis is the same:

3 NM	Speed	Time to conflict	β
	120 kt	4 minutes	22.02º
		10 minutes	8.627º
	300 kt	4 minutes	8.627º
		10 minutes	3.440º

Table 2.2: β for 3 NM when the extreme cases are analysed.

In this case β has decreased respect to the other cases, when d_{min} is higher.

Finally the case of 1NM is plotted on figure 2.8:

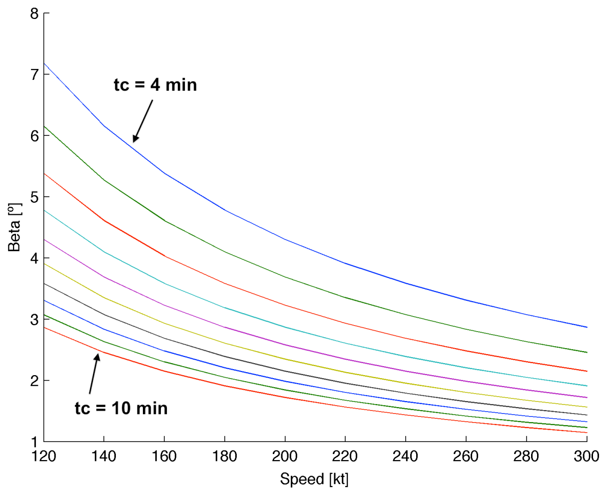


Figure 2.8: β as a function of speed for 1 NM. The different lines represent the times to conflict.

For 1 NM the results are:

1 NM	Speed	Time to conflict	β
	120 kt	4 minutes	7.181º
		10 minutes	2.866º
	300 kt	4 minutes	2.866º
		10 minutes	1.146º

Table 2.3: β for 1 NM when the extreme cases are analysed.

Finally it can be concluded that β decreases when d_{min} is decreased and also when the speed is increased. When the time to conflict is increased, β is decreased.

2.2. Minimum distance calculations (d_{min})

The second step will be computing the separation distance that can be achieved in order to know if the separation maneuver will achieve d_{min}. The geometry used to make this calculus is showed on figure 1.1.

2.2.1. Speed of the UAS and the intruder as constants

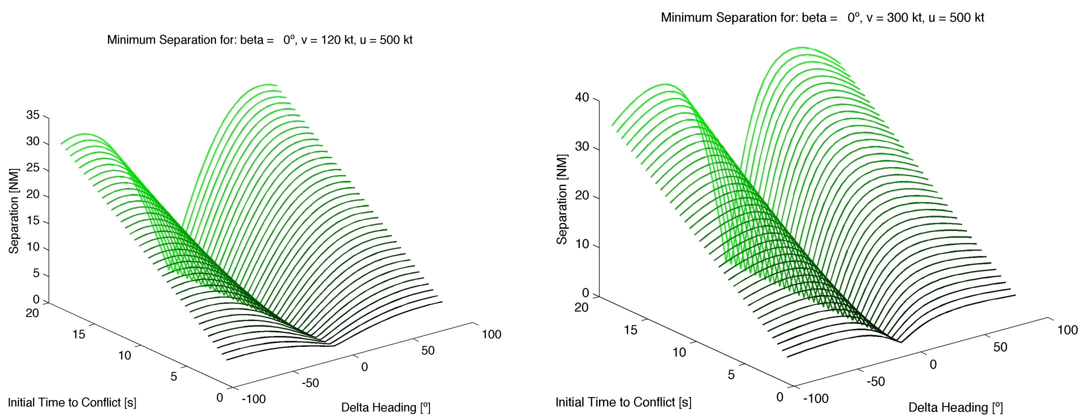
Using a Matlab function the minimum distance achieved for a change of heading is computed. The function computes the minimum distance for a range of Δh from -90º to 90º, and also for different times to conflict ranging from 2 minutes to 20 minutes with steps of 30 seconds. Speeds and heading of every aircraft are the inputs of the function and have to be introduced manually.

In order to show the results of the function the following speeds are taken:

Aircraft Model	**Speed [kt]**
Airbus A320	500 kt
General Atomics Predator	120 kt
Global Hawk	300 kt

Table 2.4: Aircraft models and their speeds

The resulting plots of the code are shown on figure 2.9:

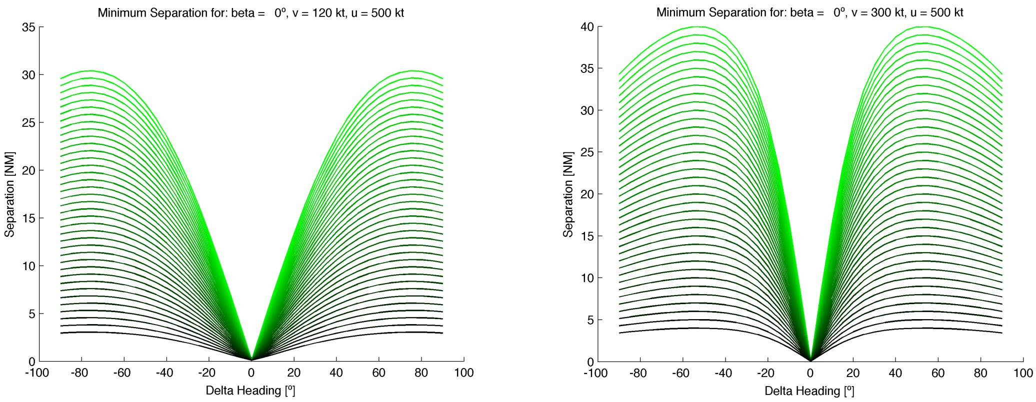


a) b)

Figure 2.9: Resulting plot of the calculus for:
a) intruder at 500 kt and UAS at 120 kt
b) intruder at 500 kt and UAS at 300 kt
The different lines respresent the different times to conflicte from 2 minutes to 20 minutes

The plots show the evolution of the minimum separation with the change of heading (∆h) for different values of the time to conflict.

So for the speeds commented above and a heading of 0º for every aircraft the resulting plots are showed on figure 2.10. They only show the heading change and the minimum separation.



a) b)

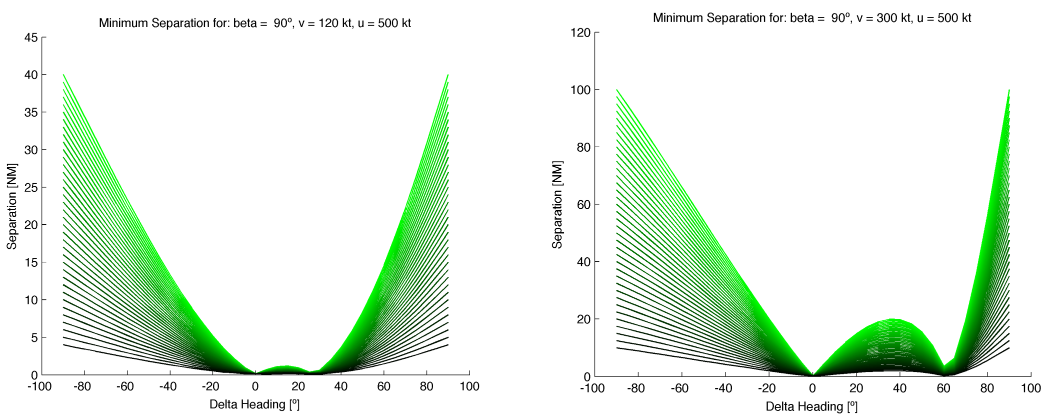
Figure 2.10: Front result of the plot for β = 0º for:

a) intruder at 500 kt and UAS at 120 kt
b) intruder at 500 kt and UAS at 300 kt

The different lines respresent the different times to conflicte from 2 minutes to 20 minutes

Looking at these plots, if there is no heading change the minimum distance the aircraft will achieve will be 0, they will collide. The lower lines are for lower times to conflict starting at 2 minutes and finishing at 20 minutes with steps of 30 seconds, so the lower the time to conflict or heading change are, the lower the separation is.

Analysing other cases, for example the case where β is 90º (The UAS heading is 0º and the airliner heading is 90º). Figure 2.11 shows the results:



a) b)

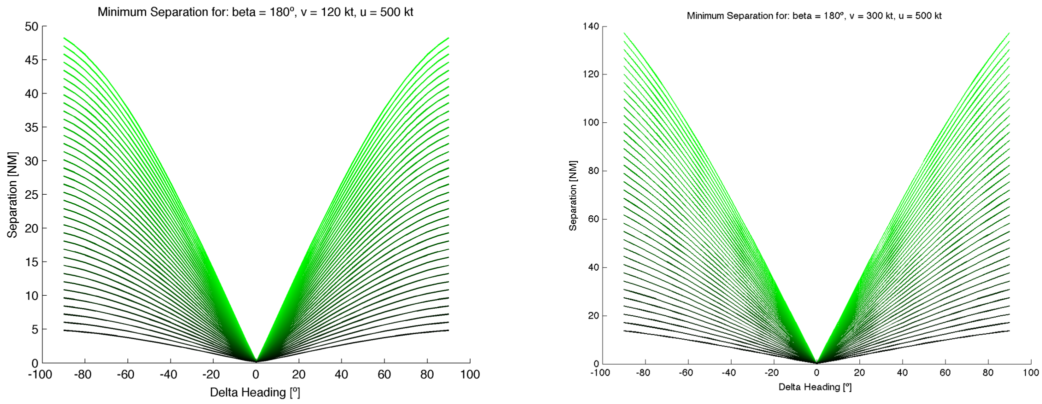
Figure 2.11: Minimum separation VS heading change for β = 90º for:

a) intruder at 500 kt and UAS at 120 kt
b) intruder at 500 kt and UAS at 300 kt

The different lines respresent the different times to conflicte, from 2 minutes to 20 minutes

For this case there are two points where the aircraft would collide when ∆h is 0º, as normal, and for a heading change of 20º and 60º approx. for the 120 kt and 300 kt cases. So in order to guarantee the minimum distance for this case higher values of ∆h are needed.

Finally if the aircraft have a parallel track but they have opposite directions the resulting separation would not be the same like β = 0º case. Figure 2.12 shows the difference.



a) b)

Figure 2.12: Minimum separation VS heading change for β = 180º for:

a) intruder at 500 kt and UAS at 120 kt
b) intruder at 500 kt and UAS at 300 kt

The different lines respresent the different times to conflicte from 2 minutes to 20 minutes

As we can see d_{min} is not the same for both cases, so β = 0º and β = 180º are taken as different cases.

2.2.2. Speed of the UAS and the intruder as variables

In order to evaluate the behaviour of the distance for every speed, a new Matlab function is used. It will take into account that the speeds of the UAS and the intruder are variables. So different cases of the conflict for several speeds can be studied. A matrix with the different values of the minimum distance is used. This code will also be used in chapter 3 for the purpose of analysing the oblique maneuver.

As explained in the previous paragraph, the speeds of both aircraft are taken as variables. The speed of the UAS will range from 120 knots, taking the speed of a slow UAS, to 300 knots, taking the speed of a fast UAS. The speed of the intruder aircraft is supposed to be the speed of a common airliner and it will range from 500 knots to 800 knots.

The code used to compute the distance is shown below:

```
v1=linspace(120,300,20);
v2=linspace(500, 800, length(v1));
v1 = v1 * 1852/3600;
v2 = v2 * 1852/3600;
tsep = linspace(120,1200,20);
deltaH = linspace(-90,90,20);


for k=1:1:length(v1)
    for m=1:1:length(v2)

        [d_min] = ls6_proves (v1(k), v2(m), 0, 0);
        for i=1:1:length(tsep)
            for e=1:1:length(deltaH)

                d(i,e,k,m)=d_min(i,e);

            end
        end
    end
end
```

Figure 2.13: Code used to compute d_{min} for the speed of the UAS and the speed of the intruder as variables

Where the function "ls6_proves" is the following:

```
function [d_min] = ls6_proves (v1, v2, h1, h2)


MAX_T = 5000;
v1_kt = v1;
v2_kt = v2;
v1 = v1 * 1852/3600;
v2 = v2 * 1852/3600;

h1 = h1 * pi/180;
h2 = h2 * pi/180;

tsep = linspace(120,1200,20);
deltaH = linspace(-90,90,20);
deltaH = deltaH * pi/180;


d_min = zeros(length(tsep), length(deltaH));
```

```
d1 = zeros(1,2);
d2 = zeros(1,2);

dprev = 0;

for i = 1: 1: length(tsep)

    d1(1) = v1 * tsep(i) * cos(pi/2 - (h1-pi));
    d1(2) = v1 * tsep(i) * sin(pi/2 - (h1-pi));


    d2(1) = v2 * tsep(i) * cos(pi/2 - (h2-pi));
    d2(2) = v2 * tsep(i) * sin(pi/2 - (h2-pi));

    d1_0 = d1;
    d2_0 = d2;

    for j = 1 : 1 : length(deltaH)
        d1 = d1_0;
        d2 = d2_0;

        for t = 1 : 1 : MAX_T
            d1(1)  =  d1(1)  +  v1  *  cos(pi/2  -  (h1  +
deltaH(j)));
            d1(2)  =  d1(2)  +  v1  *  sin(pi/2  -  (h1  +
deltaH(j)));

            d2(1) = d2(1) + v2 * cos(pi/2 - h2);
            d2(2) = d2(2) + v2 * sin(pi/2 - h2);

            d = norm(d1-d2)/1852;

            if t ~= 1
                if d > dprev
                    break;
                end
            end
            dprev = d;
        end
        d_min(i,j) = d;
    end
end
```

Figure 2.14: Function to compute d_{min} for the speeds of the UAS and the intruder as constants.

In order to make sure the code is working correctly a plot can be helpful. The following plot shows the resulting evolution of the minimum distance as a function of the change of heading. The speed of the UAS is 120 kt and the speed of the intruder is 547 kt.

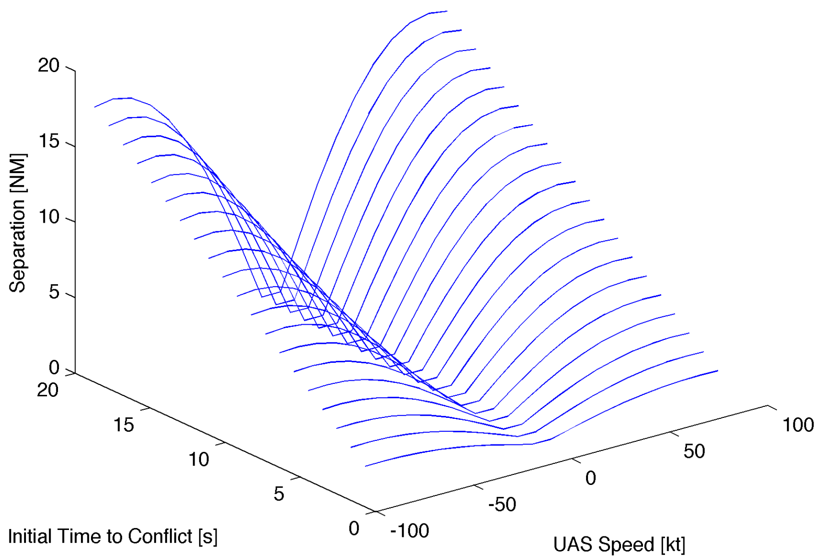


Figure 2.15: Minimum distance as a function of the heading change, for diferent times to conflicte. The different lines represent the times to conflicte, from 2 minutes to 20 minutes.

As we can see, the tendency of every line is similar to the plots shown before.

It will be interesting to analyse the influence of the speed of the UAS, and the speed of the intruder on the minimum distance calculus. For this purpose a new plot is done, with a constant heading change and a constant speed of the intruder. This way the influence of the UAS speed is analysed.

The plots are made for a constant ∆h of 10º and a constant intruder speed of 600 kt. Every line represents a different time to conflict from 2 minutes to 20 minutes with steps of 2 minutes.

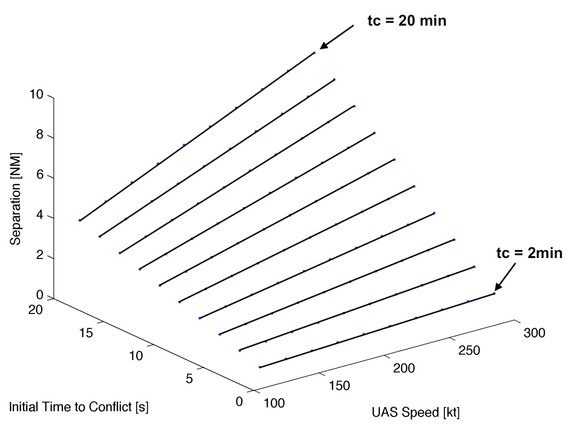


Figure 2.16: d_{min} as a function of speed of the UAS with a constant ∆h and speed of the intruder. Each line is a diferent time to conflict

For a closer look only the speed and d_{min} are shown as in figure 2.17.

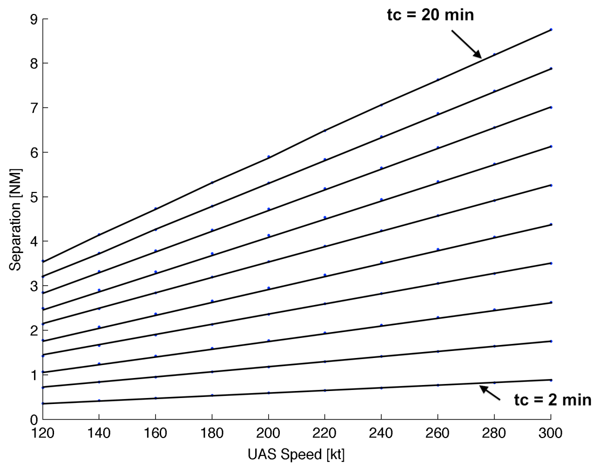


Figure 2.17: d_{min} as a function of speed with constant Δh and speed of the intruder. Each line represents a diferent time to conflict

Every line in the plot represents a different time to conflict, where the upper line is for 20 minutes, and the lower line is for 2 minutes. When the speed of the intruder is increased d_{min} is also increased. The next figure shows the same but for a higher Δh of 50º.

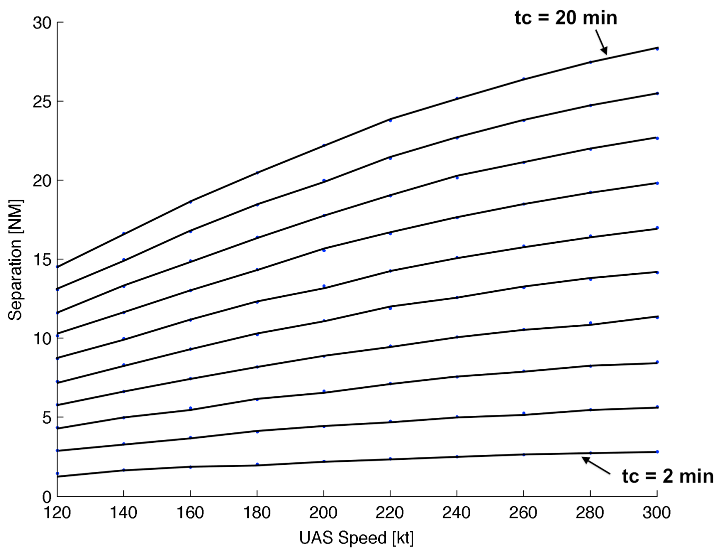


Figure 2.18: d_{min} as a function of speed for 50º heading change. Each line represents a diferent time to conflict

As we can see for a higher heading change the distance is higher, as it was explained before, and it follows the same tendency as the previous case.
In the next plot the speed of the intruder is increased to 672 kt, to show the dependence of the separation with the speed of the intruder.

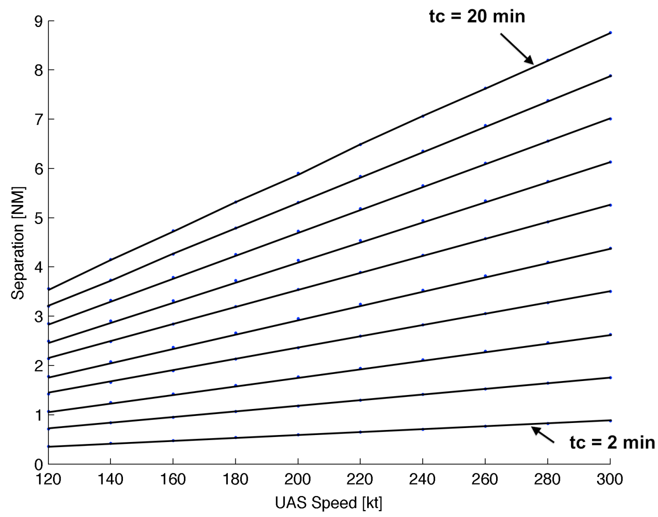


Figure 2.19: d_{min} as a function of the UAS speed and for an intruder speed of 672 kt. Each line is a different time to conflicte.

The plot is similar for the case of 600 kt. So the speed of the intruder is not relevant.

So it can be concluded that if the speed of the UAS is increased the separation distance will also be increased. As the heading change is increased the minimum distance will also be increased.

The other variable that can be analysed is the speed of the intruder, which will normally be an airliner. For this purpose the same kind of plot is done but instead of using the UAS speed, the intruder speed is used. The speed of the UAS will take a constant value of 180 kt and a constant Δh of 10º.

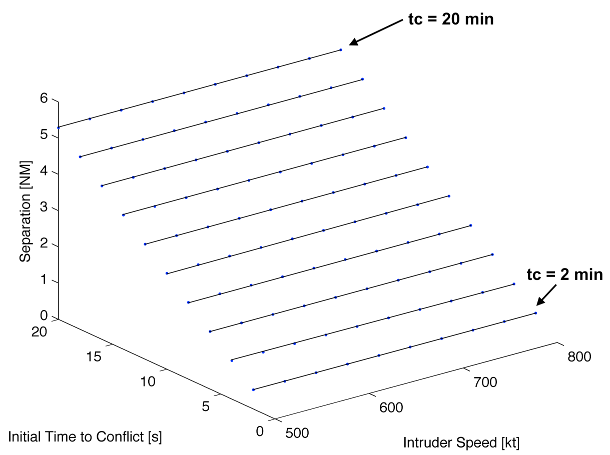


Figure 2.20: d_{min} as a function of the intruder speed for a constant speed of the intruder and Δh. Each line is a different time to conflict

So for a constant speed of the UAS and a variable speed of the intruder the separation remains constant. This is because the aircraft that is performing the separation maneuver is the UAS and not the intruder. So changing the speed of the UAS would change the value of the separation.

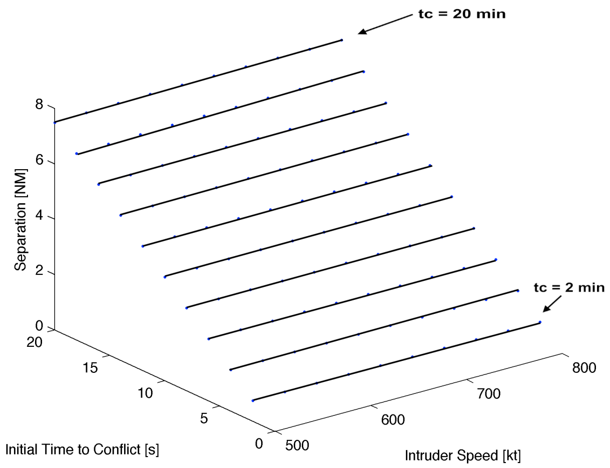


Figure 2.21: d_{min} as a function of the intruder speed, for a UAS speed of 260 kt. Each line represents a differet time to conflicte.

This plot is made for the same heading change, but for a speed of the UAS of 260 kt. This time the distance has increased. So finally we can conclude that the intruder speed has an indirect effect on the separation distance, it also depends on the UAS speed.

In conclusion the variables that most effect on the separation are the time to conflict, speed of the UAS, and the change of heading. When the time to conflict increases the separation also does. For the change of heading it decreases the separation from -90º to 0º and it increases from 0º to 90º. Finally when the speed of the UAS increases the separation also does, because the distance to conflict will also be increased. The speed of the intruder does not affect to the separation.

CHAPTER 3. OBLIQUE MANEUVER

3.1. Description of the maneuver

As it is explained in a previous chapter the oblique maneuver will be analysed. In order to execute the oblique maneuvers the UAS will turn to achieve a parallel track. This is because it is easier for the ATC controller to give the order of keeping the same distance in the entire maneuver. The other option would be giving an angle that will make the distance to vary during the maneuver and achieving the minimum distance at some point. This way if some error is committed it makes sure that there will not be a conflicting point were the minimum distance is not fulfilled.

The UAS will keep this parallel track until the conflict is cleared. The necessary change of heading (Δh) in order to achieve this parallel track is the same angle as β. The situation explained is showed in the following figure:

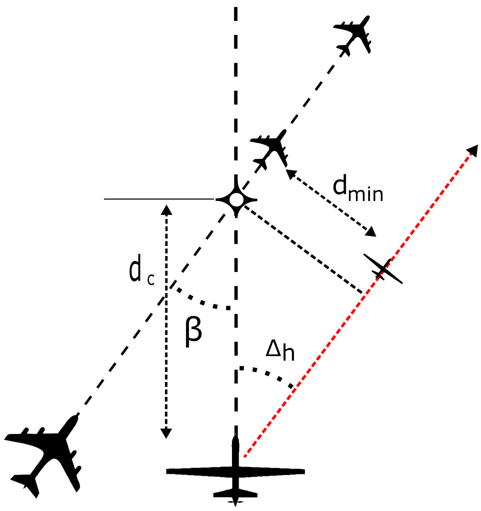


Figure 3.1: Oblique maneuver

In the figure Δh corresponds to β. As it is explained in the paragraph before the d_{min} is kept during the entire maneuver.

This maneuver is done supposing that d_{min} is achieved at the same time the collision conflict is detected. It is possible that the distance that the UAS would achieve at the detection time of the conflict would be smaller than d_{min} for that moment. In this case the maneuver is not possible. But when the distance achieved when the conflict is detected is bigger than d_{min} at that moment there are two possibilities:

- Keeping d_{min}, which means that the UAS will have to cover an extra distance.
- Execute the maneuver at the time when the conflict is detected.

The following chapter will be focussed on the first case, where the UAS covers an extra distance to get d_{min} at that moment. This maneuver is explained in figure 3.2:

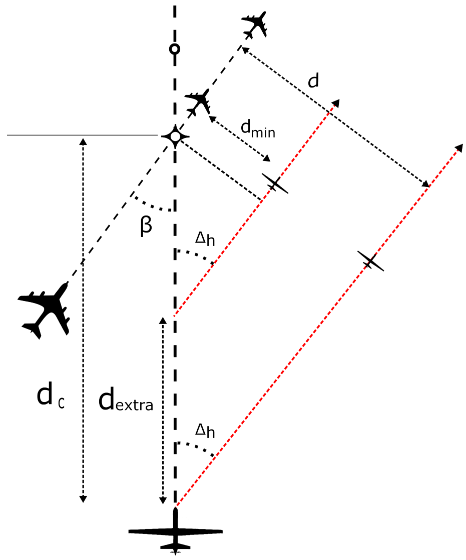


Figure 3.2: Oblique maneuver with an extra distance

Figure 3.2 shows the extra distance that the UAS will have to cover in order to achieve d_{min} at that moment. As we can see the change of heading is the same in both cases, and in both cases we achieve a parallel track, but with different distances, where $d > d_{min}$, so the UAS will have to cover a distance called d_{extra} in order to reach d_{min}.

3.2. Extra time

The UAS will have to cover the extra distance; an extra time will have to be added to the time to conflict. The extra time that the UAS will be flying in order to cover the extra distance can be computed with simple kinematics, as the linear motion. With the diagram in figure 3.2 the following expressions can be used to compute the extra time.

$$\sin(\Delta h) = \frac{d_{min}}{d_{creq}} \quad \textbf{(3.1)}$$

$$d_{creq} = \frac{d_{min}}{\sin(\Delta \mathrm{h})} \quad \textbf{(3.2)}$$

$$d_{extra} = d_c - d_{creq} = v_1 t_{extra} \quad \textbf{(3.3)}$$

$$t_{extra} = \frac{d_c - d_{creq}}{v_1} \quad \textbf{(3.4)}$$

In these equations d_{creq} is the distance to conflict needed in order to reach d_{min}. So coding these equations in a Matlab function with pre-calculated d_{min} the extra time can be computed.

The code used in order to calculate the extra time uses the function that computes d_{min} for every speed and change of headings, but with a modification. It computes the distance assuming that the UAS will always take a parallel track with the intruder aircraft. So the heading of the intruder aircraft will always be the same as the heading change. The heading of the UAS is assumed to be always 0º as a simplification of the solution because the 0º - 20º situation is the same as 20º - 40º. The second situation only changes the orientation of the reference frame but β, that is the important angle for the maneuver, keeps being of 20º. The code is the following:

```
function [t,d_min]=obliqua(dreq)
%We load a matrix computed previously in order to make the
program faster
load dmin_obliqua.mat;
deltaH=linspace(-90,90,15);
v1=linspace(120,300,15);
v2=linspace(500, 800, length(v1));
tc = linspace(120,1200, 15);
for i=1:1:length (tc)
    for j=1:1:length (v1)
        for k=1:1:length(v2)
            for m=1:1:length(deltaH)

                if d_min(i,j,k,m)>dreq % if the distance is
higher then we compute the extra time

                    dcreq=dreq/sin(deltaH(m));

dc(i,j,k,m)=d_min(i,j,k,m)/sin(deltaH(m));

                    if dc(i,j,k,m)>dcreq
                        t(i,j,k,m)=((dc(i,j,k,m)-
dcreq)/v1(j))*3600;
                    else
                        t(i,j,k,m)=-1;
                    end

                elseif d_min(i,j,k,m)==dreq

                    t(i,j,k,m)=0; %If the distance is the
same the resulting time will be zero

                else
                    t(i,j,k,m)=-1;  %If the distance is
smaller the maneuver is impossible, so a -1 will mean that
it's not possible to realise it
                end
            end
```

```
        end
    end
end
```

Figure 3.3: Code used to compute the extra time

It computes the time needed to cover the extra distance, the extra time the UAS will take to complete the maneuver correctly. Three cases are possible:

- $d > d_{min}$. This case the UAS will have to cover the extra distance to execute the correct maneuver. The resulting t is computed with the corresponding ecuations, and the result is given in seconds.
- $d = d_{min}$. This case the minimum distance corresponds to the distance that the UAS would get if the maneuver were done at the moment of detection. As there's no extra time to be added, the corresponding t is 0 seconds.
- $d < d_{min}$. This case the minimum distance cannot be achieved, because the distance we get at the detection of the conflict is smaller than the minimum distance. In order to differentiate this case from the other two cases, the resulting t will take a value of -1.

3.3. Minimum and Maximum angle for the oblique maneuver

So as to completely characterize the oblique maneuver, the maximum β and the minimum β have to be defined, to differentiate when the oblique maneuver is considered and when it is not. There will also be analysed the symmetries of the maneuver.

3.3.1. Minimum and maximum angles

The minimum angle will be determined using a Matlab function. What this function will do is to compute the corresponding speeds, Δh, and the time to conflict. This function saves the corresponding values in a vector for each variable. Then these vectors are plotted, and the evolution of each variable is showed for a given d_{min}, plotting the points where a solution is possible.
As the minimum distance matrix has 4 dimensions, in order to make a plot a variable must be taken as a constant. This variable will be the speed of the intruder that will take a constant value of 696 kt. The resulting plot is showed on figure 3.4:

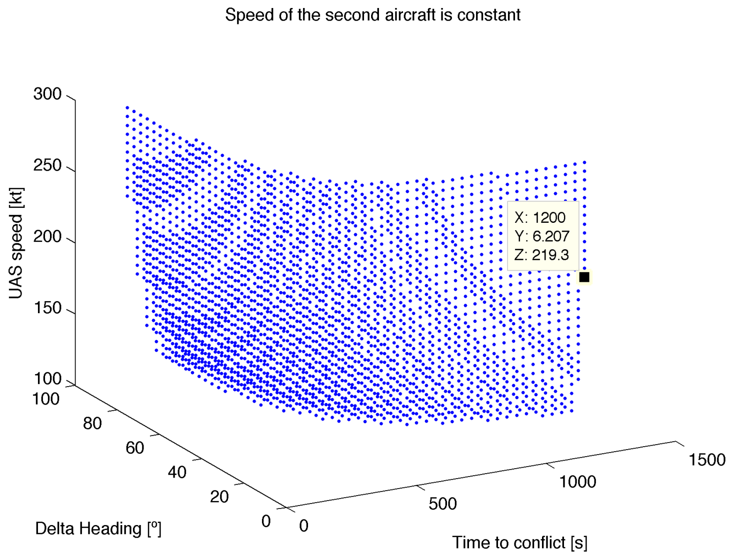


Figure 3.4: Behaviour of the UAS speed, ∆h and time to conflict for a constant intruder speed, for 5 NM separation. The box message shows the values of the three variables for the minimum case.

The part that we need to focus are the points for the smallest ∆h, that will determine the minimum β to guarantee a 5 NM separation. So the minimum β to guarantee the required separation for 5 NM is 6.027º. This change of heading must be executed with a speed of 219 kt and a time to conflict of 20 minutes. This would be the most restrictive case that could be used.
Now the maximum β must be determined using the same procedure. Figure 3.5 will show the same as figure 3.3 but for the maximum angle:

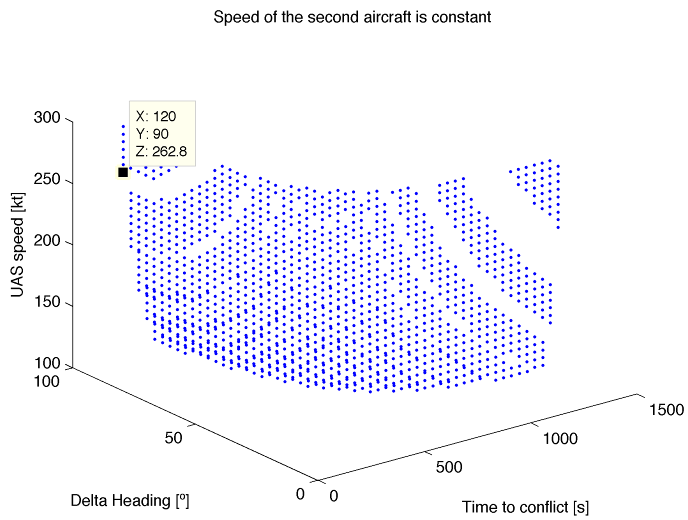


Figure 3.5: Behaviour of the UAS speed, ∆h and time to conflict for a constant intruder speed, for 5 NM separation. The box message shows the values of the three variables for the maximum case

In order to get the maximum β we will focus on the opposite part where we focused before, where the maximum angles for the separation distance of 5 NM are placed. So the maximum β is 90º. This heading change will have to be executed at a speed of 262.8 kt and when the collision conflict is detected 2 minutes before.
So for 5 NM the angles have been determined, but for a better comprehension of the maneuver we need to know the behaviour of the maximum and the minimum β. For this purpose the same procedure is done but for a separation distance of 10 NM. So figure 3.6 shows the minimum β for a distance of 10 NM.

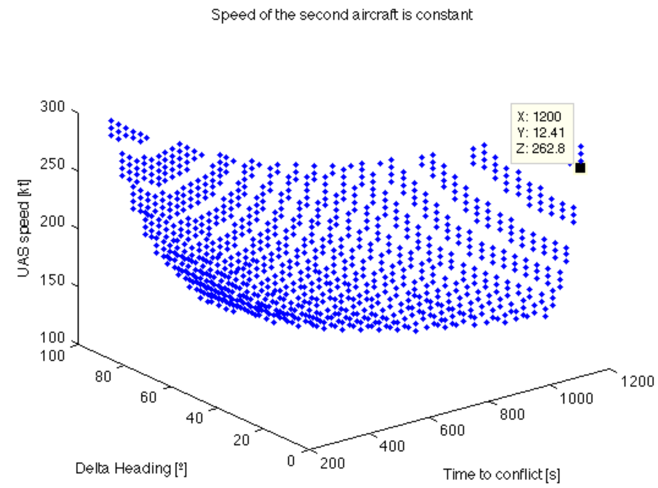


Figure 3.6: Behaviour of the UAS speed, ∆h and time to conflict for a constant intruder speed, for 10 NM separation. The box message shows the values of the three variables for the minimum case

In this case the corresponding angle is 12.41º, so the β is doubled and the speed of the UAS has increased and the time to conflict is the same. The speed has increased but not in a noticeable way. So when the distance is doubled, the ∆h and β also do.

For the maximum β, it continues being of 90º. But the time to conflict has increased a to 283 seconds as the distance has been increased.

So it can be concluded that the minimum β to consider a maneuver as oblique is 6.027º and the maximum is 90º for a d_{min} of 5 NM. When this distance is doubled the minimum β also does, so the angle increases linearly with the separation distance.

3.3.2 Symmetry of the maneuver

For the purpose to analyse the symmetry of the maneuver a wider range of β and ∆h must be analysed. The angles that will be analysed will be:

- From -90º to 90º
- From 0º to -180º
- From 0º to 180º

Figure 3.7 shows these cases, where the red areas are the cases described before:

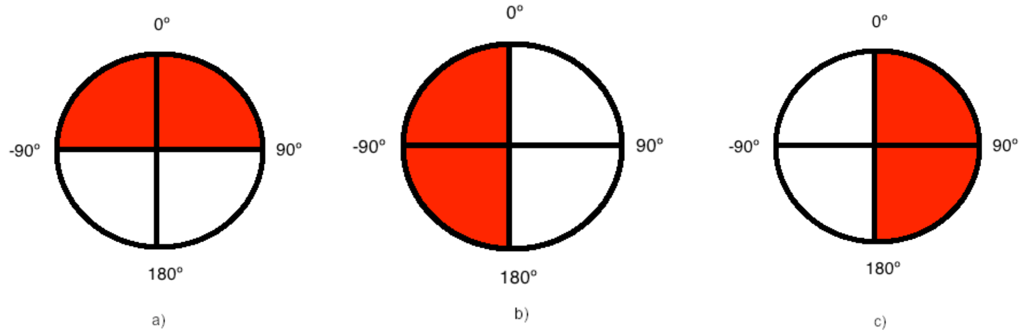


Figure 3.7: Regions which symmetry will be analysed

a) From -90º to 90º
b) From 0º to -180º
c) From 0º to 180º

The first case that will be studied will be the case for a range of β that will range from -90º to 90º, case a) in figure 3.5, in order to view if the maneuvers between a range of β from 0º to -90º are symmetric to the ones executed at 0º to 90º. Using Matlab a similar plot to the one used to characterize the angles will be used, but the heading change will range from -90º to 90º. Figure 3.8 shows this result:

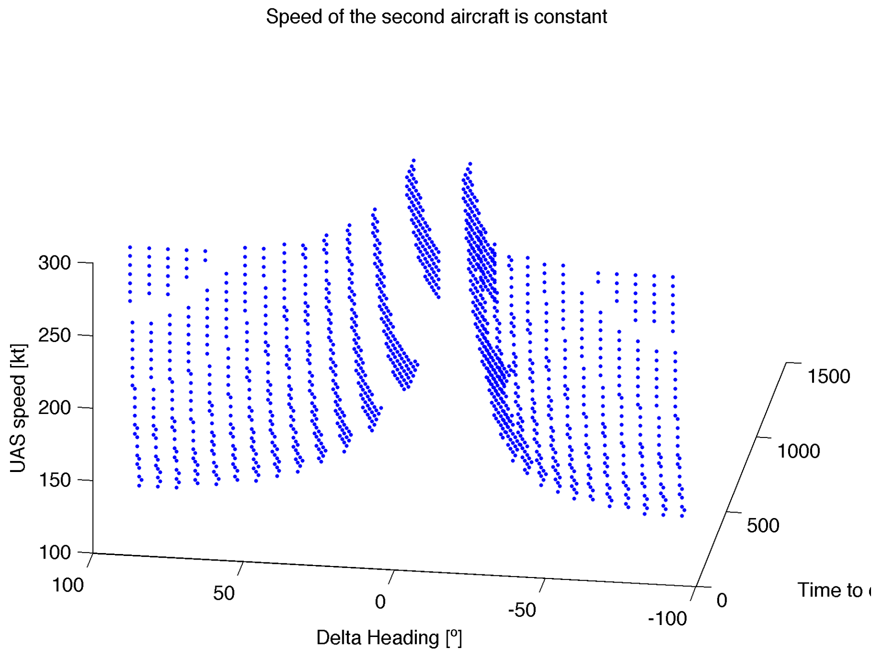


Figure 3.8: Behaviour of the UAS speed, Δh (ranging from -90º to 90º) and time to conflict for a constant intruder speed, for 5 NM separation.

In figure 3.8 we can see that there is a region near the 0º values where the separation distance of 5 NM cannot be guaranteed. Both regions where the solution is possible are symmetric. So for angles from 0º to -90º the oblique maneuver could be executed starting at -6.027º until an angle of -90º and it will have the same behaviour as the case from 0º to 90º.

Now the second case, with the angles from 0º to -180º, will be analysed. The purpose of this case is to analyse if the maneuver is the same for the cases from 0º to -90º and for the cases from -90º to -180º. Figure 3.9 will show this case for a 5 NM separation distance:

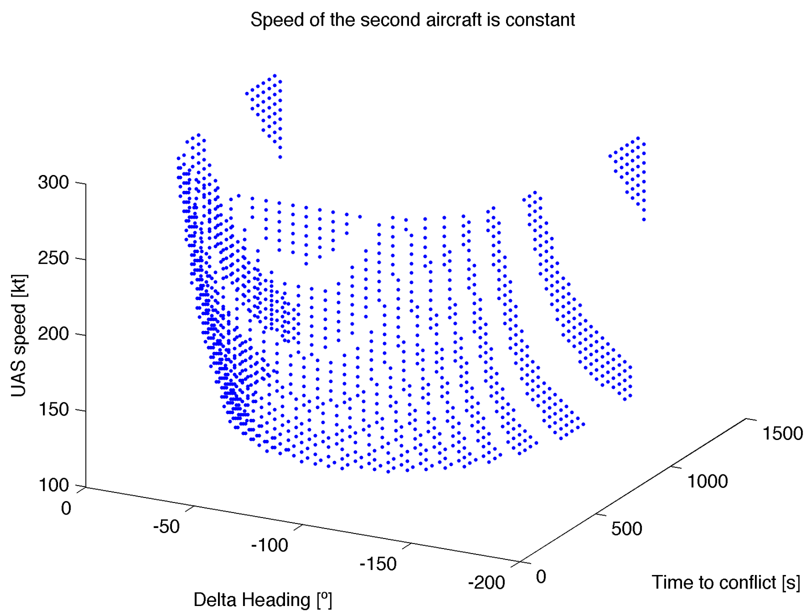


Figure 3.9: Behaviour of the UAS speed, Δh (ranging from -180º to 0º) and time to conflict for a constant intruder speed, for 5 NM separation.

Figure 3.9 shows that there is symmetry from the angles ranging from 0º to -90º and the angles ranging from -90º to -180º. The limiting angles are at the sides of the plot where one side is -6.027º and the other side is -173.8º, which corresponds to the minimum angle of the oblique maneuver minus 180º. So it can be concluded that for the case from 0º to -180º the maneuver is also symmetric.

For the third case, with the angles ranging from 0º to 180º, the same procedure is done. So in this case the angles from 90º to 180º will be analysed if they accomplish the same behaviour as the angles from 0º to 90º. Figure 3.10 will show the results:

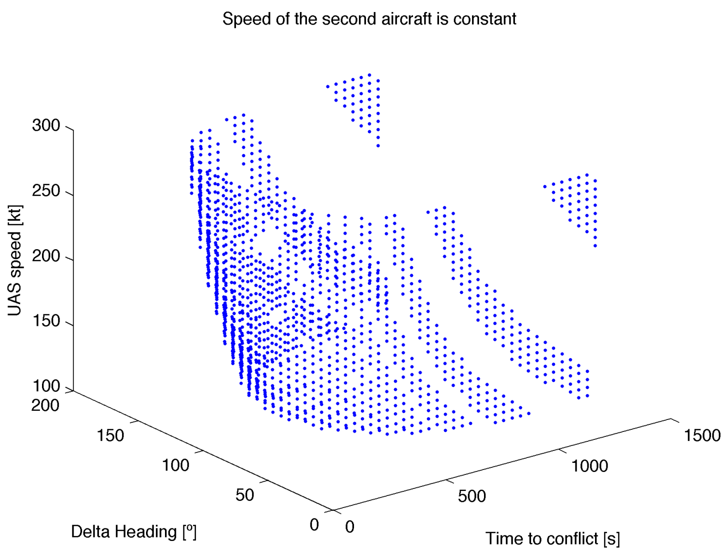


Figure 3.10: Behaviour of the UAS speed, Δh (ranging from 0º to 180º) and time to conflict for a constant intruder speed, for 5 NM separation.

As we can see in figure 3.10 the result is equal as the result in figure 3.9. So it can be concluded that for this case the behaviour is also symmetric and with limiting angles of 6.027º and 173.8º.

Summing up there are two regions, near 0º and near 180º, where the oblique maneuver cannot be executed. The regions where the oblique maneuver can be executed are shown in figure 3.11, where the green regions is where the maneuver can be executed and the red regions are where the maneuver cannot be executed. The following figure is done for the 5 NM case, but for the 10 NM case the angles 6.027º and -6.027º are doubled.

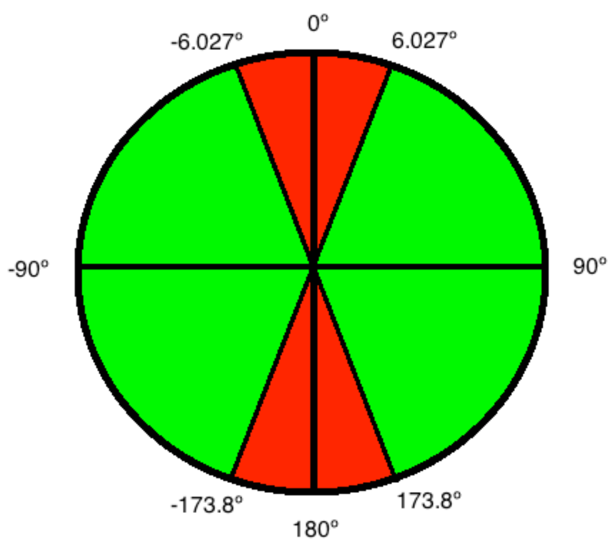


Figure 3.11: Regions where the oblique maneuver is possible

CHAPTER 4. INTERPOLATION

When the UAS is in a mission it would fly at a constant speed, with a given heading. If a collision conflict is detected, its speed and heading have to be the correct ones in order to execute the manoeuvre correctly. So the ATC controller would have to tell the UAS which speed and which heading change is needed for the purpose to clear the collision conflict in a correct way. After this orders the UAS would have to adapt its speed and heading to the ones the ATC controller told to it. So this speed, and change of heading have to be computed for different cases of the required separation. For the different computed cases the ATC controller would have to give the best order to the UAS in terms of separation distance.

4.1. Calculus of the required variables

In order to solve the previous situation it is needed to compute the speed of the UAS, the speed of the intruder, the change of heading of the UAS and the separation time needed for the separation.

To solve this situation a Matlab code is used. This code will search for every position in a matrix with the solutions of the distance, previously computed with the code used in chapter 2, when both aircraft have the same heading, $\beta = 0º$. As the matrix is computed for different values of speed, times to conflict and heading changes, d_{req} is almost impossible to be found inside the matrix, so the function looks for the exact reference value, but if it is not found it looks for a number between a margin of 0.5 NM. Then it saves the values of time to conflict, speed of the UAS, speed of the intruder and Δh in a vector for every variable. These vectors are the outputs of the function.

The following code is used for this purpose:

```
function [Tc, DELTAH, V1, V2]=interpola(dreq)

load d.mat;

v1=linspace(120,300,25);
v2=linspace(500, 800, length(v1));
v1 = v1 * 1852/3600;
v2 = v2 * 1852/3600;

tsep = linspace(120,1200, 18);
deltaH = linspace(-90, 90, 25);
  I=1;

for i=1:1:length(tsep)
    for j=1:1: length(deltaH)
        for k=1:1:length(v1)
            for m=1:1:length(v2)
```

```
                    if d(i,j,k,m)==dreq
                        Tc(I)=tsep(i);
                        DELTAH(I)=deltaH(j);
                        V1(I)=v1(k)*3600/1852;
                        V2(I)=v2(m)*3600/1852;

                        I=I+1;

                    elseif        d(i,j,k,m)<=dreq+0.5         &&
d(i,j,k,m)>=dreq-0.5
                        Tc(I)=tsep(i);
                        DELTAH(I)=deltaH(j);
                        V1(I)=v1(k)*3600/1852;
                        V2(I)=v2(m)*3600/1852;

                        I=I+1;
                    end
                end
            end
        end
end
```

Figure 4.1: Code used in order to find the values of the time, Δh, and speeds for a given d_{req}

Where `Tc` corresponds to the time to conflict needed and it is given in seconds, `DELTAH` is the change of heading and it is expressed in degrees, and `V1` and `V2` are the speed of the UAS and the speed of the intruder respectively and they are given in knots.

If there is no possible solution for d_{req}, the function would not work, and in the Command Window on Matlab would give an error message that the output variables, `Tc`, `DELTAH`, `V1` and `V2`, are not defined in the program, they have no value.

The following tables show some examples of this interpolation (the values are depicted in the table are only five of them from the whole vector):

Distance: 5 NM	Computed distance [NM]	Tc [s]	DELTAH [º]	V1 [kt]	V2 [kt]
	4,53	183,53	-90	285	800
	4.64	183,53	-75	292,5	750
	4.58	247.05	-90	270	575
	4.73	310.58	45	195	550
	5.17	755.29	-22.5	127.5	737.5

Table 4.1: Interpolation for 5 NM

As it is explained before the possibility of d_{req} corresponding to the computed distance is almost impossible, so there will usually be values between d_{req} + 0.5 and d_{req} – 0.5.

4.1. Influence of the variables on the minimum distance (d_{min})

In chapter 2 the evolution of distance with all the variables was plotted, and it was possible to see the influence of them. In this chapter the influence of all the variables in the calculus of the distance is going to be analysed.

The procedure is the following:
- First we are going to execute the interpolating function, with a required distance.
- We will keep one of the variables as a constant, these constant variables will correspond to different cases that will be explained later.
- Apart from saving the values of time, speed and heading change the positions of the value in the matrix of distances will be also saved.
- The saved indexs will be represented in a 3D plot (that is why it is needed that a variable must be constant, the matrix of distances has four dimensions).

This way the position of the variables in the matrix of distance is plotted, assuming one of the variables as a constant. This is a simple way to analyse the influence of the variables in the distance calculation.

So as it is commented before, a variable must be taken as a constant, this means that different cases can be analysed:
1. The intruder does not cooperate. That means that the intruder will keep its speed constant during all the collision conflict, so $v_2$will remain as a constant.
2. The UAS cannot change its speed (due to some kind of issue of the mission it would be completing). This means that the speed of the UAS will remain as a constant, that will make v_1 constant.
3. The UAS can only make a determined change of heading, due to it is performing a oblique maneuver. This means that the change of heading has to remain constant, so Δh will be constant.
4. The detection is only possible at a given time. This would make the time to conflict constant, so t_c will be constant.

4.2.1 CASE 1: The intruder does not cooperate

The first case that is going to be studied is the case when the intruder aircraft does not cooperate. This case the intruder aircraft cannot change its speed, so it will remain constant during the entire maneuver.

Using the previous code with some variations in order to maintain the intruder speed as a constant value, the influence of the other three variables can be plotted. The intruder will maintain a constant speed value of of 637.5 kt, and the resulting plot, for a distance d_{min} of 5 NM, is shown in the next figure:

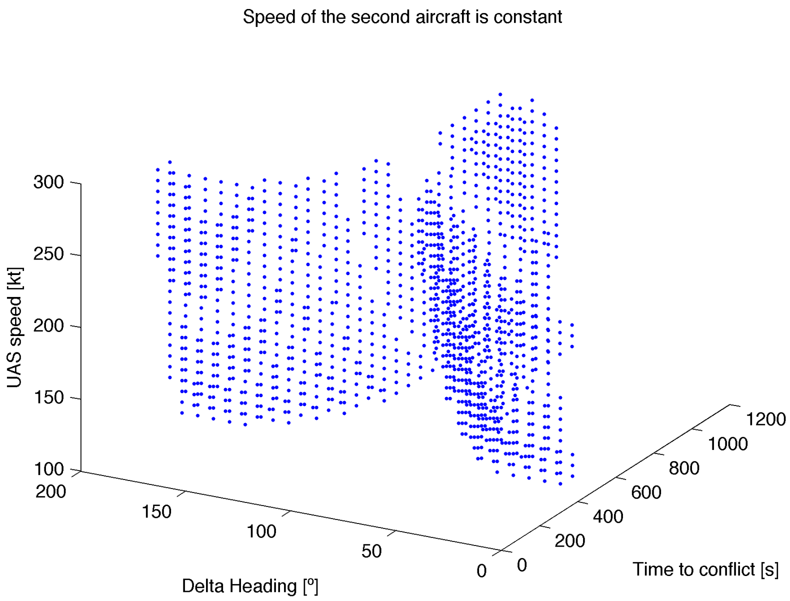


Figure 4.2: Influence of time, Δh and speed of the UAS

The next figure shows the same case for 15 NM d_{min} in order to differentiate from the 5 NM case:

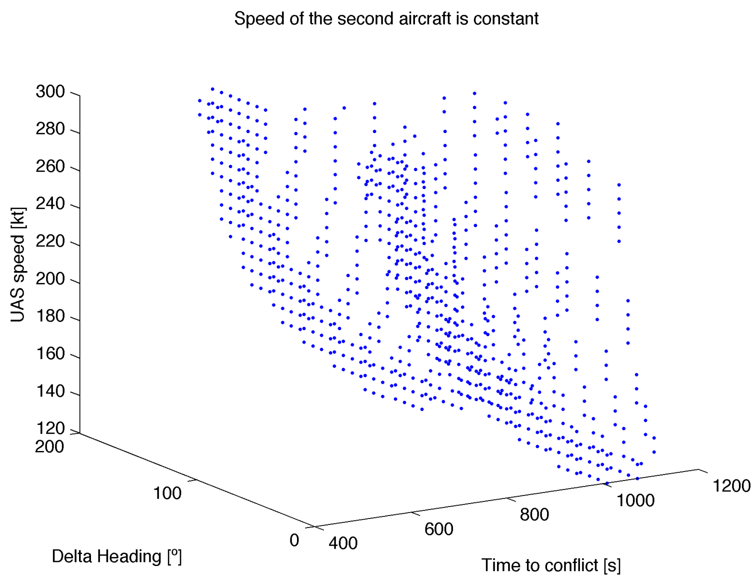


Figure 4.3: Influence of time, Δh and speed of the UAS

Comparing both figures we can determine which is the most restrictive variable. For the case of the intruder speed as a constant, the most restrictive variable is the speed of the UAS that it would have to be noticeably increased, in order to reach the desired separation, for a same Δh, and a same time to conflict.

If we forget about the speed and we focus to the other two variables it will be possible to determine their influence. So for the same cases explained before the same plots are obtained, first for 5 NM:

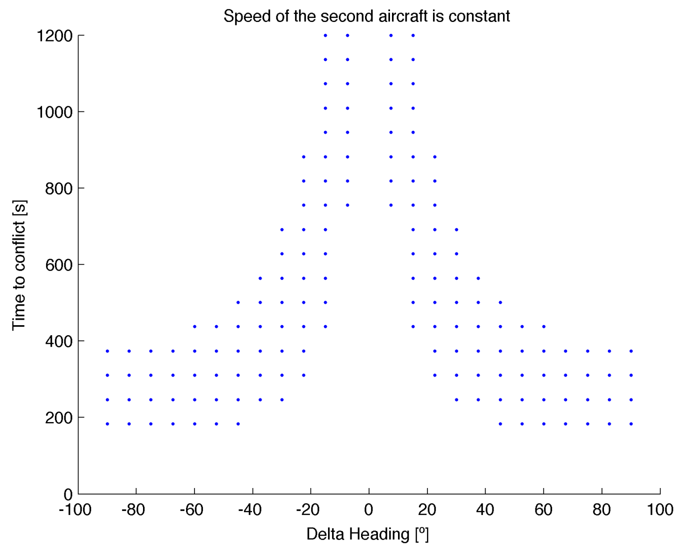


Figure 4.4: Influence of time, Δh and speed of the UAS from the top view

The same is done for 15 NM d_{min}:

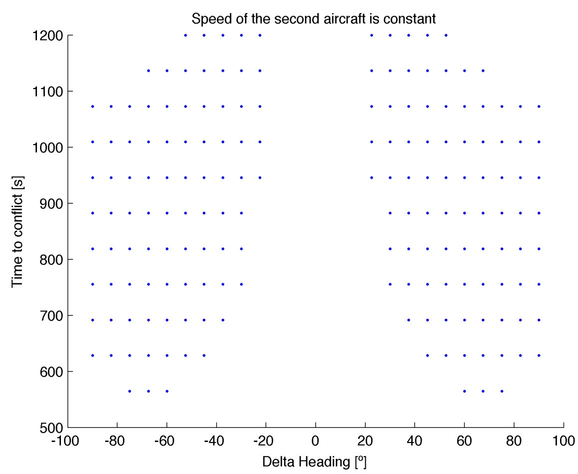


Figure 4.5: Influence of time and Δh

Comparing the two figures we can see that d_{req} can be accomplished for every time to conflict starting at 628,2 s in the case of 15 NM, and for every time to conflict starting at 183,5 s for 5 NM, but as d_{req} is increased; the change of heading must be increased too.

In conclusion the most restrictive parameters are the change of heading done by the UAS that will be given by the geometry. So the variables that most affect the separation are the time to conflict and the speed of the UAS.

4.2.2. CASE 2: The UAS cannot change its speed

The second case that will be studied is when the UAS will keep its speed; it will not change its speed, due to a requirement of the mission.

The same procedure as in the previous case is done. But for this case instead of keeping the intruder speed as a constant, the UAS speed will be constant, taking a value of 235,7 kt. The first plot is done for a distance of 5 NM:

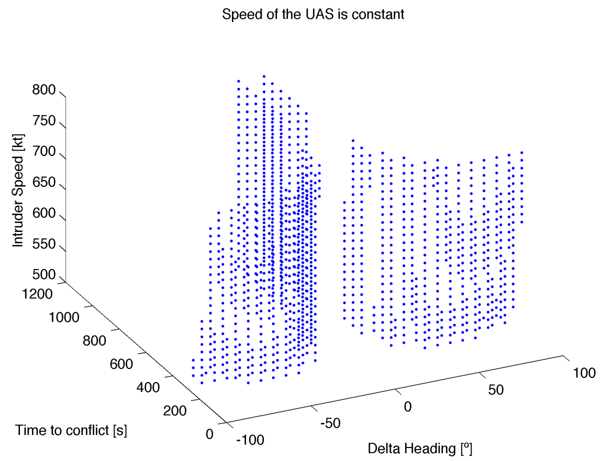


Figure 4.6: Influence of time, Δh and speed of the intruder

For 15 NM d_{min} the plot is:

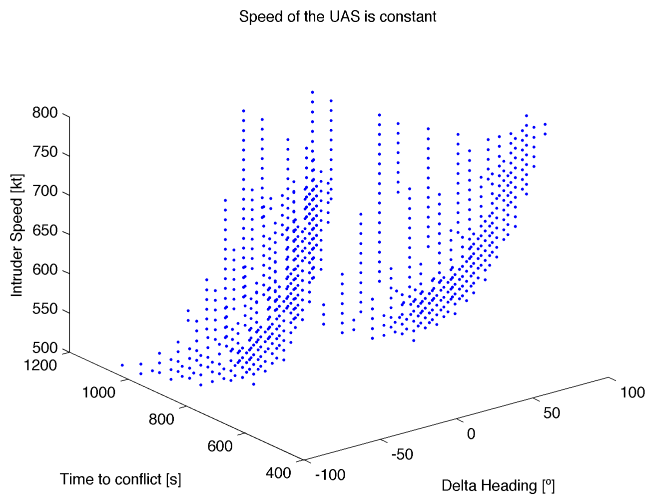


Figure 4.7: Influence of time, Δh and speed of the intruder

Comparing both figures we can see that if the distance is increased the corresponding speed must also be increased, but not as much as the previous case, where the speed of the UAS had to be increased a bigger amount than in this case, for a given time to conflict. There are also less possibilities of the time to conflict. The first result is placed at the corresponding position for 891,4 s for 15 NM, and for 5 NM corresponds to 274.3 s.

Comparing the same plots but in the view from the top we could analyse the influence of Δh in the manoeuvre:

For the case of 5 NM d_{min}:

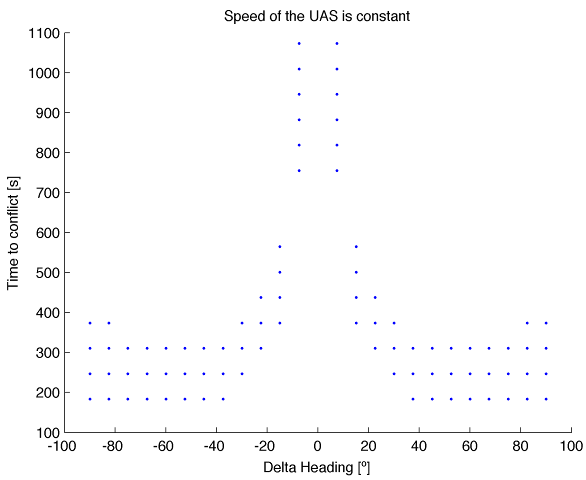


Figure 4.8: Influence of time and Δh

For the case of 15 NM d_{min}:

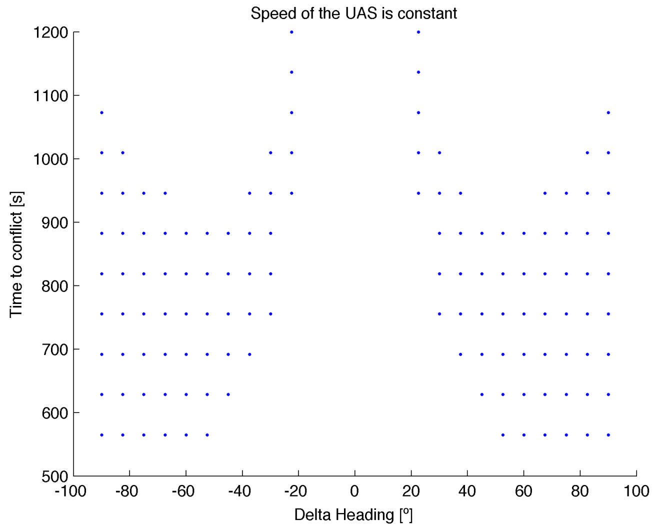


Figure 4.9: Influence of time and Δh

In this case the change of heading that has to be performed increases when d_{req} is increased, as in the plots where in the 15 NM the change of heading increases considerably compared to the 5 NM for the same cases of time to conflict.

In conclusion, when the speed of the UAS is maintained constant the variables that most affect the separation distance are the change of heading and the time to conflict.

4.2.3. CASE 3: The UAS can only make a determined change of heading

The third case that will be studied is when the UAS can only make a given change of heading. This could happen when the UAS must not separate from the path of its mission. So in this case the change of heading will remain constant.

For this case Δh will be constant, taking a value of 30º. The first plot is done for a distance of 5 NM:

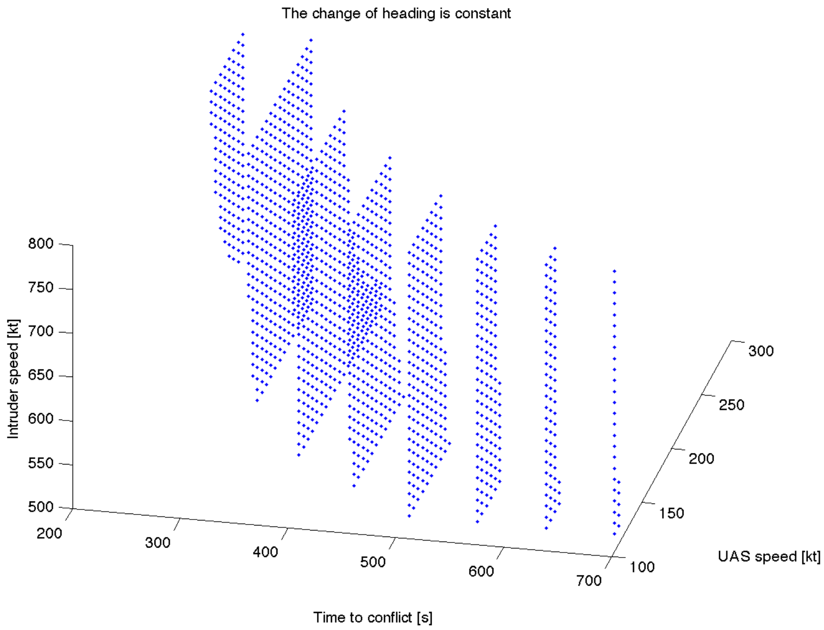


Figure 4.10: Influence of time, intruder speed and UAS speed

For the case of 15 NM:

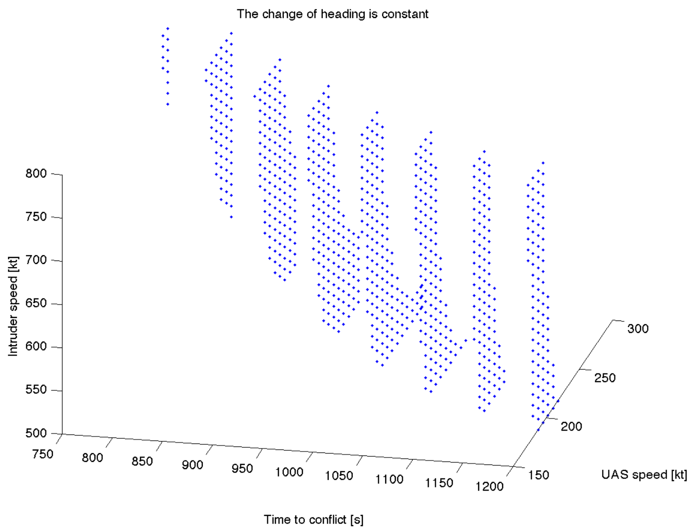


Figure 4.11: Influence of time, intruder speed and UAS speed

For 5 NM the top view is:

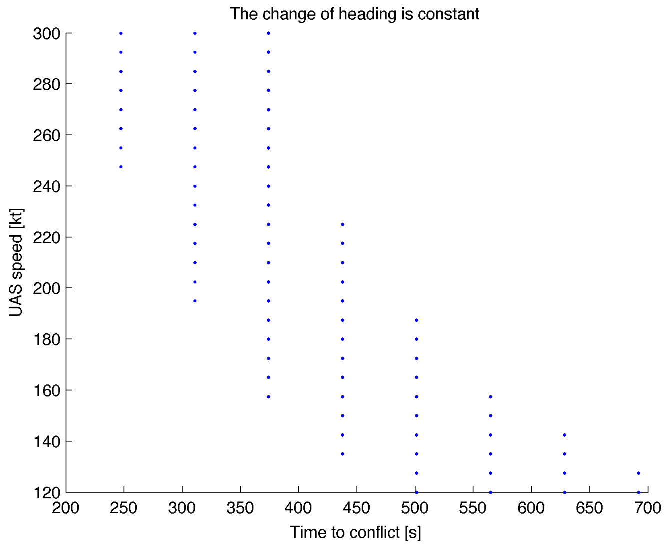


Figure 4.12: Influence of time and the UAS speed

For 15 NM the top view is:

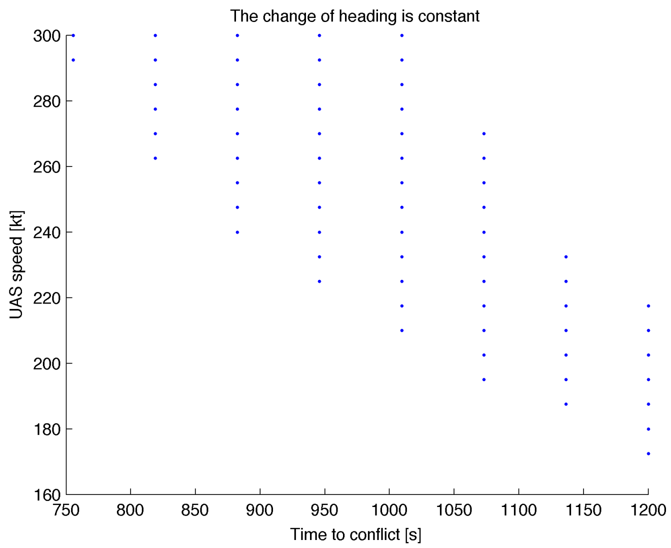


Figure 4.13: Influence of time and the UAS speed

Comparing the plots we can establish which is the variable that most affects the separation. As in the previous cases, not all the times to conflict generate the desired separation. They also start at 274.3 s for the case of 5 NM and at the 891.4 s for the case of 15 NM. But the last value that accomplishes the requirement of 5 NM is for a time of 691.7 seconds; from this time to the last value the separation will be higher. It also is affected by the speed, for a smaller time to conflict a higher speed is needed. The intruder speed has a smaller effect, due to that for every speed of the intruder there's a possible solution.

In conclusion, when the change of heading is maintained as a constant the variables that most affect the separation are the speed of the UAS and the time to conflict, so this are the variables that must be taken into account in this case.

4.2.4. CASE 4: The detection is only possible at a given time

The last case studied will be when the detection can be done only for a given time. This will make the detection time as a constant.

The constant time of detection will take a constant value of 882.3 seconds. As in the previous cases the first plot is for the 5 NM situation:

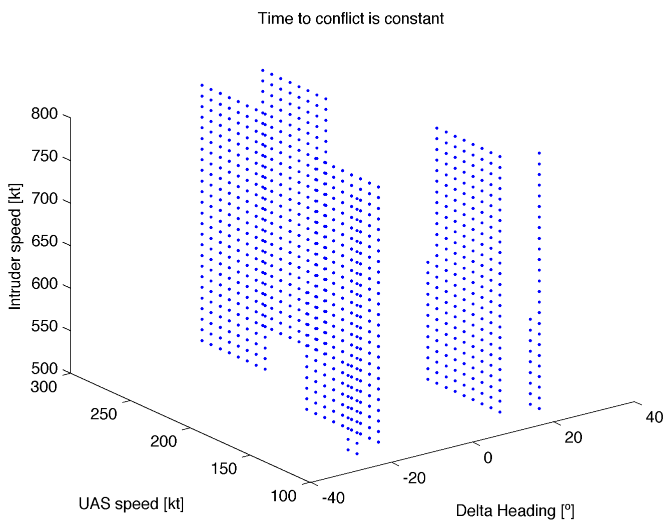


Figure 4.14: Influence of Δh, intruder speed and UAS speed

For 15 NM:

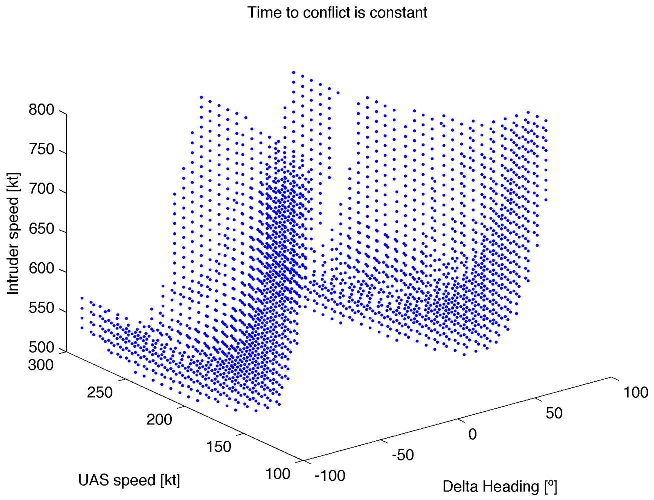


Figure 4.15: Influence of Δh, intruder speed and UAS speed

The top view for 5 NM is:

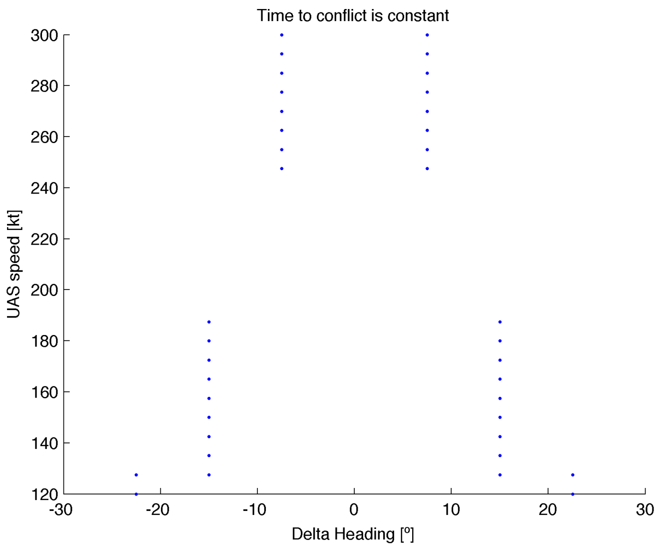


Figure 4.16: Influence of the heading change and the UAS speed

The top view for 15 NM is:

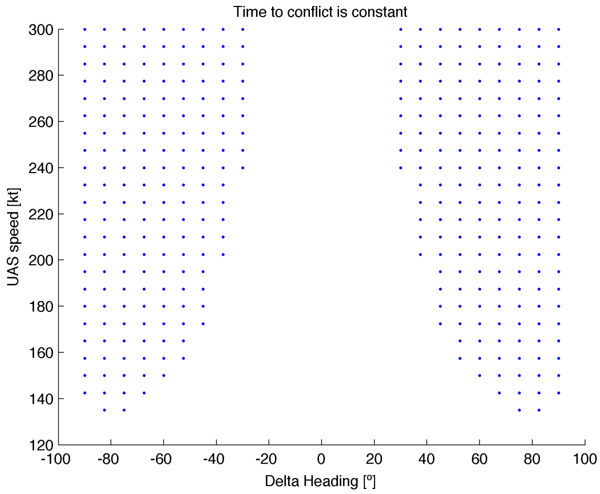


Figure 4.17: Influence of the heading change and the UAS speed

Comparing both plots they seem to have a different shape, but taking a closer look both plots have he same tendency, but for the case of 5 NM the possibilities of this distance are enclosed in a small range of the intruder speed. For the case of 15 NM the range of aircraft speed is higher. So for the case of the time to conflict taken as a constant the intruder speed has a bigger importance than in the other cases. There's also important the change of heading that for higher speeds would have to be decreased in order to get the correct separation.

4.2.5. Conclusions

All the variables used in the separation maneuver have been analysed, and their influence in it. The most important variable is the speed of the UAS that with small changes will increase notably the separation. The other variable that highly affects the separation is the change of heading that for small changes will highly increase the value of the separation. There is also very important the time to conflict, which also increases the distance of separation, but it's not a variable that the UAS or the ATC controller would be able to control. Finally the speed of the intruder is less important because a change of its speed does not affect in a notorious way at the separation. So it can be concluded that the most important variables that must be taken into account when the separation maneuver must be executed in order to clear a collision conflict are the speed of the UAS and the change of heading.

CHAPTER 5. CONCLUSIONS

During the development of this project the issue of the separation distances between UAS has been analysed using Matlab for the purpose to simulate several situations of the collision conflict. The purpose of these simulations is to determine which is the behaviour of the different variables that affect the separation distance, and determine which ones affect to maneuver.

In the second chapter of the project the separation distance for different values of speed, change of heading and time to conflict are computed. This way it is possible to see the separation distance that the UAS can achieve doing a separation maneuver. With these calculations the behaviour of the variables in the calculus of d_{min} is showed.

In the third chapter the oblique maneuver has ben analysed and the angles that define this maneuver have been obtained. Over the maximum angle and below the minimum angle will not be considered as an oblique maneuver. There has also been determined the symmetry of the maneuver. The time the UAS will have to wait until it can start the maneuver for a given value of the separation distance has been computed. This way the ATC controller will be able to give the order to the UAS to wait the necessary time and then execute the maneuver.

Finally using an interpolation function the behaviour of the variables has been studied. The variable that most affect the separation is the speed of the UAS. Other variables like the heading change and the time to conflict are also very important, but the time to conflict cannot be selected, it is given for the situation, and the change of heading can change the value of the separation distance with a small change, as the speed of the UAS does.

BIBLIOGRAPHY

[1] Dalamagkidis K, Valavanis K. P. and Piegl L. A, *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification and recommendations*, International series on intelligent systems, control, and automation: science and engineering, S. G. Tzafestas, Ed. Springer-Verlag, 2009.

[2] Pérez M., Pastor E. and Prats X., *Evaluation of separation strategies for Unmanned Aerial Systems,* Technical University of Catalonia.

[3] Brooker P., *Air traffic control separation minima: Part 1-the current stasis*, Journal of navigation, vol. 64, pp. 449- 465, 2011.

[4] Pérez M., Pastor E., Royo P., Prats X. and Barrado C., *A taxonomy of UAS separation maneuvers and their automated execution,* ICARUS Research group.