Abstract—This paper presents a guidance algorithm for an unmanned aerial vehicle (UAV). It combines a nonlinear lateral guidance control law, originally designed for UAVs tracking circles for mid-air rendezvous, with a new simple adaptive path planning algorithm. Preflight path planning only consists of storing a few way points guiding the aircraft to its targets. The paper presents an efficient way to model no-fly zones, to generate a path in real-time to avoid known or "pop-up" obstacles, and to reconfigure the flight path in the event of reduced aircraft performance. Simulation results show the good performance of this reconfigurable guidance system which, moreover, is computationally efficient.

I. INTRODUCTION

Over the last two decades, many path planning algorithms have been investigated, especially for ground robots, for a single unmanned aerial vehicle (UAV), and more recently for a formation of UAVs. Among the methods used in path-planning, we can mention the Probabilistic Road Maps (PRM) method [1] which explores all the possible paths within the space surrounding the vehicle, and finally selects the lowest cost route. However, the computational load makes the PRM method impractical for real-time path planning in small UAVs. An extension to the PRM method has recently been presented in [2]. It is called modified Rapidly-exploring Random Trees (RRT), which is capable of efficiently searching for feasible paths in the space while taking into account constraints from the vehicle performance. However, efforts are still going on to implement an on-the-fly path replanning system as pop-up obstacles are discovered, or when the performance of the vehicle degrades. There are other methods based on potential field functions. However, the primitive forms of potential field functions present some difficulties when choosing an appropriate potential function, and the algorithm may be stuck at some local minimum [3]. Since then, a whole family of potential field methods with superior performance has been developed. They are known as Navigation Functions [4], [5]. Other path-planning techniques are based on optimization methods, such as Mixed Integer Linear Programming or Model Predictive Control techniques [6], which still involve heavy computations.

In this paper, we present a guidance algorithm for a UAV. It newly combines the lateral guidance control law from [7], originally designed for UAVs tracking circles for mid-air rendezvous, with a new, simple adaptive path planning algorithm, which takes advantage of the curve path-following property of the above-mentioned lateral guidance law. This path-planning method generates online a flight path based on predefined waypoints, takes into account the aircraft performance, avoids known or appearing obstacles, is simple to implement, and requires low computational power.

II. LATERAL GUIDANCE SYSTEM

A. Lateral guidance control law for trajectory tracking

Consider Fig. 1, where an aircraft has to be guided to track the desired path. From the current location of the aircraft O, we can draw an arc of a circle that intersects the desired path at a “reference point” P, where R is the radius of the circle-arc OP, \( L_1 \) is the segment that joins the center of the aircraft O to the reference point P, and \( \eta \) is the angle between the aircraft’s velocity vector and the line \( L_1 \). The lateral acceleration required to bring the aircraft to the reference point following the arc of a circle is

\[
\alpha_l = \frac{V^2}{R},
\]

with \( V \) being the longitudinal ground speed of the aircraft (taken in the inertial frame).

Let us express \( R \) in terms of the distance \( L_1 \) and the angle \( \eta \). The triangle (OCP) is isosceles in C, therefore we have \( L_1 = 2R \sin \theta \). We also have \( L_1 = 2R \cos \gamma \). Moreover \( \gamma = \frac{\pi}{2} - \eta \), and consequently \( L_1 = 2R \sin \eta \). The lateral acceleration can now be written as

\[
\alpha_l = \frac{2V^2}{L_1} \sin \eta.
\]

In turn the lateral acceleration \( \alpha_l \) is converted to a bank angle command, \( \phi_{com} \approx \frac{\alpha_l}{g} \).

Fig. 1. Guidance law geometry
1) Remarkable performance for curve path following:

This control law is remarkable in the sense that it is particularly suited to fly circles. Indeed, if the aircraft is following a desired circular path, then the acceleration command \(a_t\) generated by the guidance system is exactly the same as the associated centripetal acceleration. In other words, the guidance logic chooses a reference point on the desired path at a distance \(L_1\) ahead of the aircraft, and generates the acceleration command that would lead the vehicle to hit the point, thus leading to zero-steady state error for a circle path. As shown in chapter 3 of [7], the performance of such a lateral guidance law for circle-following in the presence of wind is superior to that obtained with PD or PID controllers. It can be noted that with the nonlinear guidance law \(a_t = \frac{2V^2}{L_1}\sin\eta\), the vehicle ground speed \(V\) is used to generate the acceleration command, which intrinsically takes into account the inertial velocity changes due to the wind effects, and adapts to the situation accordingly.

2) Properties associated with the angle \(\eta\): We can see from (2) that the direction of the acceleration depends on the sign of \(\eta\). For example, if the reference point is on the right side of the direction of the aircraft velocity vector, then the aircraft will be commanded to accelerate to the right, and finally the aircraft will tend to align its velocity vector with the direction of \(L_1\).

Since the angle \(\eta\) used in the control law contains the information about the upcoming path, this geometric factor has the effect of a feedforward control.

C. Selection of \(L_1\)

The design parameter in the lateral guidance logic is the distance \(L_1\) between the vehicle and the reference point. Fig. 2 shows that for a small magnitude of \(\eta\), the guidance formula can be approximated in terms of the cross-track error \(y\) as follows:

\[
a_t = \frac{2V^2}{L_1}\sin\eta \approx \frac{2V}{L_1}\left(y + \frac{V}{L_1}y\right)
\]

Equation (3) shows that the guidance law is equivalent to a PD controller, in which the ratio between the vehicle speed \(V\) and the distance \(L_1\) is an important factor that behaves as the gain of the controller. A small value of \(L_1\) leads to a high control gain and vice versa. The control gain is limited by the inner loop bank control bandwidth (2~3 rad/s). With a nominal flight velocity of around 25 m/s, the distance \(L_1\) has been chosen to be \(L_1 = 150\) m.

D. Drawback of the method

A drawback of the control law in (2) is that it does not contain the element of an integral control. Thus, the lateral guidance law is not robust to bias in the lateral acceleration measurement. Consequently, a non-biased bank angle estimate is required.

E. Path-planning objective

The main objective of this paper is to generate on-line an appropriate reference path from which the reference point \(P\) can be selected and used by the lateral guidance control law.

III. REGULAR WAYPOINT TRACKING

The regular waypoint tracking algorithm guides the aircraft through the predefined waypoints. An imaginary segment joins two consecutive points, and we define Segment \(k\) to be the segment that joins the waypoints \(WP_k\) and \(WP_{k+1}\). The reference point, \(P\), from the guidance law presented above, lies on the segment and is \(L_1\)-distant from the center of the aircraft, see Fig.3.

A. Computation of the reference point \(P\)

The angle of the segment \(k\) with respect to North is defined as \(\psi_{seg(k)}(\in [-\pi; \pi])\):

\[
\psi_{seg(k)} = \tan^{-1}\left(\frac{WP_{k+1}E - WP_kE}{WP_{k+1}N - WP_kN}\right)
\]  

(4)

The coordinates of the current location of the center of the aircraft are in the north-east plane \((X_N, X_E)\). The angles \(\alpha\) and \(\lambda\), and the distance \(d_1\) from Fig. 3 can be computed with

\[
\alpha = \tan^{-1}\left(\frac{X_E - WP_kE}{X_N - WP_kN}\right) \in [-\pi; \pi]
\]

\[
\lambda = |\psi_{seg(k)}| - |\alpha|
\]

\[d_1 = \sqrt{(X_E - WP_kE)^2 + (X_N - WP_kN)^2}
\]

(5)

The distance between \(WP_k\) and the reference point \(P\) is given by

\[
[WP(k), P] = [WP(k), H] + [H, P] = d \cos \lambda + \sqrt{L_1^2 - d_1^2 \sin^2 \lambda}
\]

(6)
Finally, the coordinates of the reference point \( P \) can be computed with

\[
\begin{align*}
P_N &= WP(k)_N + [WP(k), P] \cos \psi_{seg}(k) \\
P_E &= WP(k)_E + [WP(k), P] \sin \psi_{seg}(k)
\end{align*}
\]  

(7)

B. Logic for segment switching

As the aircraft flies, the reference point \( P \) also moves along the desired trajectory which consists of consecutive segments. Therefore, the guidance algorithm has to select properly the current segment on to which the reference \( P \) is to be selected.

If the distance \( L_1 \) is longer than \( d_2 \) in Fig. 3, this means that the extremity of the current segment (Segment \(_k\)) has been reached, and a new reference point has to be selected on the next segment, Segment \(_{k+1} = [WP_{k+1}; WP_{k+1}]\). On the other hand, if \( d_2 > l_1 \), it means that the aircraft is still far from the end of the current segment. It that case, the path-planning system has to check the lateral distance between the aircraft and the segment followed, in order to make sure that it selects a point on the current segment that is \( L_1 \) distant from the aircraft.

Two cases can be distinguished:

- If \( |\lambda| > \pi/2 \), then the aircraft is somewhere behind the first point of the current segment. In that case, the distance \( L_1 \) is selected as \( L_1 = \max(L_1, d_2) \).
- If \( |\lambda| < \pi/2 \), the guidance algorithm proceeds in checking the lateral distance \( [C, H] = d_1 \sin \lambda \). If \( [C, H] > l_1 \), then the distance \( L_1 \) is assigned a new value with \( L_1 = [C, H] \ast 1 \).

In this way, we always ensure that the arm \( L_1 \) is long enough to intersect the desired trajectory, so that a reference point \( P \) always exits.

IV. Failure Detection and Evaluation of Reduced Performance

The failure detection and isolation system (FDI) described in [8] and [9] determines the performance of the aircraft after the occurrence of a failure and passes two parameters to the guidance system presented here. These parameters are the maximum bank angle \( \phi_{max} \) that the guidance algorithm is allowed to command and \( \tau_{roll} \), which is the time needed for the aircraft to bank to \( \phi_{max} \).

V. No-Fly Zone (NFZ) and Obstacle Detection

Before the flight, the location of any known NFZ are stored in the memory of the autopilot. If the UAV is equipped with scanning sensors that can detect pop-up obstacles, our path-planning system will recompute on the fly a new trajectory. It determines whether an NFZ or an obstacle interferes with the planned path by using an imaginary “detection line” of length \( R_{LA} \) in front of the aircraft, as shown in Fig. 4. The distance \( R_{LA} \) defines the so-called “look-ahead distance”. If any part of this detection line penetrates an NFZ or obstacle, avoidance action is immediately taken as described in the next section.

A. Definition of a no-fly zone

Since the vertical limits of the NFZ are not considered, the NFZ is essentially a two-dimensional surface. The aircraft is not allowed to pass over or under the NFZ. The shape of the no-fly zone is modeled as a circle to benefit from the lateral guidance control law especially suited to fly circles. Although this paper only discusses the avoidance of one circular NFZ, the algorithm can be extended to multiple no-fly zones. Also, a complex no-fly zone shape can be represented by multiple circles.

B. Choice of an appropriate look-ahead distance \( R_{LA} \)

Choosing a good value for \( R_{LA} \) is important to allow the aircraft enough time to maneuver away from the NFZ without penetrating it.

The look-ahead distance \( R_{LA} \) is chosen such that the aircraft will fly an arc that stays just outside the NFZ and start the evasion maneuver as late as possible. The value of \( R_{LA} \) depends on the radius of the NFZ \( R_{NFZ} \), the ground speed of the aircraft \( V \), and the maximum bank angle of the aircraft \( \phi_{max} \). Given these parameters, and assuming a coordinated turn, the minimum turn radius that the aircraft can fly is given by

\[
R_{min} = \frac{V^2}{g \tan(\phi_{max})}
\]  

(8)

In the case of an NFZ with infinite radius, the aircraft would have to make a 90° turn, in which case \( R_{LA,min} = R_{min} \). For any NFZ with a finite radius, the aircraft has to turn less than 90° to avoid it.

Fig. 4 shows the situation where the aircraft is at the point where it begins its turn and is to be guided so that its path is tangent to the edge of the NFZ. A triangle can be set up with vertices at the center of the NFZ, at the center of the aircraft, and at a point \( R_{min} \) off the right wing-tip. The minimum look-ahead distance \( R_{LA,min} \) is then given by

\[
R_{LA,min} = \sqrt{2R_{min} + R_{NFZ}\sqrt{R_{NFZ} - R_{NFZ}}}
\]  

(9)

Fig. 4. Approaching a no-fly zone or an obstacle. Definition of the look-ahead distance \( R_{LA} \).

To obtain the final value for \( R_{LA} \), compensation must be made for the delay needed to initiate the turn, and to bank to \( \phi_{max} \). The assumption is made that while the aircraft is initiating the turn, it continues to fly level, and then as soon as it reaches \( \phi_{max} \) it takes a minimum radius
The characteristic time \( \tau_{\text{roll}} \) to roll to \( \phi_{\text{max}} \) can be multiplied by the aircraft’s speed to obtain the distance the aircraft will travel during this delay, which is added to \( R_{\text{LA},\text{min}} \). The resulting look-ahead distance is

\[
R_{\text{LA}} = R_{\text{LA},\text{min}} + V \tau_{\text{roll}}. \tag{10}
\]

C. Detection of the no-fly zone

As mentioned before, the algorithm monitors a line ahead of the aircraft. First, the distance \( D_{\text{NFZ}} \) from the aircraft to the center of the NFZ is calculated.

\[
D_{\text{NFZ}} \leq R_{\text{NFZ}} + R_{\text{LA}} \tag{11}
\]

If the condition set in (11) is satisfied, where \( R_{\text{NFZ}} \) is the radius of the NFZ, then the aircraft is considered to be within range of the NFZ. In this case, a further check is made to see if a part of the detection line is touching the NFZ.

For the second check, there are two possible cases, depending on the position of the aircraft. A pair of triangles is created as shown in Fig. 5 and Fig. 6. The edges \( h \) and \( R_{\text{LA}} \) and the angle \( \alpha \) are known. The lengths of the edges \( y \) and \( a \) can easily be calculated, using

\[
y = h \sin(\alpha) \\
a = h \cos(\alpha). \tag{12}
\]

Case 1 applies if \( a \leq R_{\text{LA}} \). The limiting case occurs when edge \( a \) is tangent to the NFZ. Therefore, \( y \) will have a length equal to \( R_{\text{NFZ}} \). Thus, the NFZ touches the detection line if \( y \leq R_{\text{NFZ}} \).

Case 2 applies if \( a > R_{\text{LA}} \). The limiting case occurs when the end of the detection line is on the edge of the NFZ. This can be checked by comparing the length of edge \( x \) with the radius of the NFZ, so that the NFZ touches the detection line if \( x \leq R_{\text{NFZ}} \), where

\[
x = \sqrt{y^2 + (a - R_{\text{LA}})^2}. 
\]

The check for Case 1 or Case 2 is only done if \( \alpha \) is less than or equal to 90°. If \( \alpha \) is greater than 90°, then the center of the NFZ lies behind the aircraft and no avoidance action is taken.

VI. NO-FLY ZONE AVOIDANCE ALGORITHM

The no-fly zone avoidance algorithm guides the aircraft around any NFZ that the aircraft encounters. The avoidance method is designed to be simple to implement while allowing the aircraft to reach waypoints close to the edge of the no-fly zone.

A. Avoidance path template

One key feature of this avoidance method is the online generation of a circular arc around the NFZ as a reference path, drawn as a dashed line in Fig. 7. Such a path minimizes the distance the aircraft flies to avoid the NFZ. Moreover, we saw at the beginning of this paper that the lateral guidance control law is particularly efficient in tracking such a path. Choosing the reference path to be circular allows the template path to be easily defined in relationship to the NFZ dimensions. It is indeed defined by the center of the NFZ and a path radius, \( R_1 \), which is simply the NFZ radius plus a safety margin. The aircraft follows this path until it is able to continue towards the next waypoint in a straight line and without passing through the NFZ.

B. Generating the template path

Once the evasion maneuver is complete, the extremity of the monitoring line lies outside the NFZ. If the next waypoint is obstructed by the NFZ, the guidance system guides the aircraft around the NFZ until there is a clear line of sight to the next valid waypoint.

Then, the guidance system has to guide the aircraft to a point \( P \) that is on the circle \( R_1 \) and at a distance \( L_1 \) ahead of the aircraft. Using the law of cosines, the angle \( \beta \) (see Fig. 8) between the segment that joins the center
of the aircraft to the center of the NFZ and the segment from the center of the aircraft to the new reference point $P$ is expressed as

$$\beta = \arccos \left( \frac{D_{NFZ}^2 + L_1^2 - R_1^2}{2D_{NFZ} \cdot L_1} \right)$$ (13)

In order to make sure that the generated circle to avoid the NFZ is a feasible path for the aircraft, the radius $R_1$ is selected as $R_1 = \max(R_{\text{min}}, R_{NFZ} + \text{safetyvalue})$.

The coordinates of the reference point $P$ can then be computed as follows:

- if the NFZ is to be avoided on the right side, we compute $\Psi_{\text{avoid}} = \Psi_{NFZ} + \beta$,
- if the NFZ is to be avoided on the left side, we compute $\Psi_{\text{avoid}} = \Psi_{NFZ} - \beta$,

and finally

$$P_N = X_N + L_1 \cos(\Psi_{\text{avoid}})$$
$$P_E = X_E + L_1 \sin(\Psi_{\text{avoid}}).$$ (14)

### C. Avoidance guidance schedule

1) **Choice of avoidance side, $T_1$**: Whether the guidance algorithm chooses to go left or right around the NFZ is determined by the side of the NFZ center to which the aircraft is already flying. If the aircraft’s velocity vector is pointing to the right of the NFZ center, then the aircraft will fly around the right side of the NFZ. If the velocity vector is pointing to the left side, then the aircraft flies around the left side. A circular NFZ makes this decision easy.

2) **Transition to reference path, $T_2$**: Once the aircraft begins its turn, it continues to turn until its velocity vector is tangent to, or points outside of the NFZ. At this point the guidance switches to follow the circular reference path.

3) **Transition to normal waypoint tracking, $T_3$**: Once the aircraft is following the reference path, it continues to do so until it has a straight line-of-sight to the next waypoint that is unobstructed by the NFZ. At $T_3$, the guidance switches out of avoidance mode and guides the aircraft to the next waypoint. Once the next waypoint is reached, guidance continues as normal.

### D. Properties of the guidance schedule

The guidance schedule presented here has several desirable properties. It attempts to minimize the number of waypoints that are unreachable by initiating the avoidance maneuver as late as possible.

It does not require complex logic to decide how to avoid the no-fly zone. Finally it minimizes the time and distance to return to the original flight path. It does this by flying directly to the next waypoint after the NFZ as soon as is safely possible.

### VII. Adaptation to Reduced Aircraft Performance

The key to simplicity is to interpret any reduction in aircraft performance about the roll axis as a limitation on the bank angle. This paper addresses failures, for which the reduction of the aircraft performance may be translated into a reduction in the bank angle.

A limited maximum bank angle has a corresponding minimum turn radius. This minimum turn radius is taken into account in two places, namely the choice of $R_{\text{LA}}$ and the choice of $R_1$. By limiting both of these values, the guidance system operates without knowing anything more about the aircraft system failure.

Under normal operation, we simply make sure that $R_1$ is not smaller than the minimum turn radius of the aircraft. If needed, $R_1$ is increased to accommodate the increased minimum turn radius associated with a smaller bank angle, possibly due to some failure.

The look-ahead $R_{\text{LA}}$ increases as a result of a smaller bank angle. The larger value of $R_{\text{LA}}$ makes the aircraft begin its avoidance maneuver earlier, allowing it to avoid the no-fly zone, hence taking into account possibly reduced aircraft performance.

### VIII. Simulation

#### A. Simulation setup and scenario

Simulations were done on a non-linear 6-DOF computer model of a radio controlled aerobatic aircraft. The model has a 4-axis simple autopilot which allowed us to directly give the autopilot a bank angle command and command it to keep airspeed, altitude, and side-slip constant.

Three similar scenarios were simulated with the results presented below. In all scenarios, the aircraft is following a desired path that passes through a no-fly zone. The simulation was done with maximum banks angles of $30^\circ$ and $15^\circ$ in the case of a normally functioning airplane and a failed airplane, respectively.
### Simulation Results

1) **No failures**: This first scenario, shown in Fig. 9, highlights the basic response of the aircraft to an NFZ blocking its path. The aircraft begins south of the NFZ and flies north along the desired path. The desired path then continues through the NFZ, but the aircraft deviates around it to the right before returning to the desired path north and east of the NFZ.

![Fig. 9. No Failures](image)

2) **Aileron failure with no guidance reconfiguration**: The left aileron is now stuck up at mid-motion range. In this scenario the guidance system is not reconfigured. We can see in Fig. 10 that the aircraft initiates its evasion procedure at the same time as in the no-failure case. However, due to the degraded performance, the aircraft is not able to turn right as fast as before. Therefore the aircraft penetrates the no-fly zone or collides with the obstacle.

![Fig. 10. With aileron failure, no guidance reconfiguration](image)

3) **Aileron failure with guidance reconfiguration**: In the scenario pictured in Fig. 11, the left aileron is still stuck up at mid-motion range. Now the guidance system is reconfigured based on the knowledge of the fault and the new parameters for $\phi_{\text{max}}$ and $\tau_{\text{roll}}$. As a result, the look-ahead distance $R_{La}$ is longer, therefore the aircraft starts its evasion maneuver earlier than in previous cases. Despite the degraded performance, the aircraft is capable of avoiding the obstacle and circumnavigating it on the right side with a larger turn radius. Therefore, the aircraft does not penetrate the no-fly zone and flies around it at a larger distance than in the previous two cases.

![Fig. 11. With aileron failure, with guidance reconfiguration](image)

### IX. Conclusion

This paper presents a new guidance algorithm that remarkably combines simplicity and the ability to avoid a no-fly zone and adaptivity to aircraft performance reduction described by a decrease in the maximum bank angle allowed. The algorithm successfully guides the aircraft with degraded performance around the no-fly zone or the obstacle and then resumes flying along the desired path. Finally, the method is computationally efficient.

### References


