Preprints of the 20th World Congress
The International Federation of Automatic Control
Toulouse, France, July 9-14, 2017

4D Trajectory Design for Vision Only Sense and Avoid Flight Test ⋆

Mark Melczer [*] , **Peter Bauer** [*,**] , **Jozsef Bokor** [*,**]

[*] *MTA SZTAKI, 1111. Budapest, Hungary (e-mail: melczer@sztaki.hu)*
[**] *Author is with MTA-BME Control Engineering Research Group*

Abstract: In order to test an aircraft's Sense & Avoid system featuring vision based detection and avoidance, a test track is needed to be constructed. The easiest way is to design two parallel paths which removes the need for timing. However, intersecting paths are needed to be constructed as well, which turns the design process into a 4D problem with critical timing. The paper introduces geometry generating methods in calm and in windy conditions, consequently validating them with the help of Software-In-The-Loop (SIL) Monte Carlo simulation and Hardware-In-The-Loop (HIL) simulations.

Keywords: Sense & Avoid, collision trajectory, wind compensation

1. INTRODUCTION AND MOTIVATION

The development of simple and relatively easy methods for manufacturing and building small unmanned aerial vehicles - or UAVs - initiated a quick growth in the popularity of these aircraft among a wide audience. In order to make the integration of these UAVs into a national airspace possible, increased flight-supervision by auxiliary systems is an exigency. The research regarding the effective application of lightweight Sense & Avoid systems on small-sized, fixed wing UAVs started to gain interest in the recent years(Degen (2011), Melnyk et al. (2014)), Forlenza (2012)).

For the real-flight testing of such systems, one aircraft provides a platform for the camera system and the other plays the role of an 'intruder'. At the Institute for Computer Science and Control of the Hungarian Academy of Sciences (MTA SZTAKI) there is a currently ongoing project in the phase of real flight tests (Zsedrovits et al. (2016), Bauer et al. (2016)). It is not a demanding task to test the system while flying the aircraft on parallel paths, however, finding the appropriate geometry for paths, which intersect under a given angle, was a challenge to be faced. Since reaching the exact start position is uncertain with manual take off, it was needed to be ensured that the two aircraft reach the collision point at the same time in order to trigger the avoidance algorithm.

There are various methods in the literature (Alturbeh (2014), Kim et al. (2007), Richards and How (2002)) for generating a collision-avoiding path. They introduce how a trajectory can be designed using splines, Model Predictive Control, different kinds of probabilistic methods and even Mixed Integer Linear Programming which considers constraints in the system. Each of the solely aircraft-related articles, however, focus on the design of such trajectory that avoid dangerous encounters. On the other hand, there are examples in the field of space flight where the ultimate goal is to establish physical contact between spacecraft, satellites and stations via performing a rendezvous maneuver. Luo et al. (2010) introduce a rendezvous and docking mission planner system developed by the authors. It is capable of designing the suitable trajectory and flight segments in the pre-mission phase. It proposes an absolute and a relative trajectory planning method as well. In our case, however we need to design absolute trajectories during the mission inasmuch as after the human piloted take-off the aircraft's initial position coordinates in the North-East inertial plane are not known beforehand.

This paper is going to introduce possible methods for planning collision trajectories where the avoidance algorithms could be tested. The goals are still the same for everyone: avoiding the collision and safely continuing the flight; but first the visual perceptional systems are needed to be tested in multiple collision scenarios. The forthcoming section introduces the concepts behind the design of the trajectories then Section 3. will deal with the wind compensation methods. Finally, Monte Carlo simulation test results will be introduced along with the brief overview of the successful HIL simulations and the paper ends with a conclusion.

2. TRAJECTORY DESIGN

This section deals with the design of the UAVs' non-parallel tracks. The initial situation is that the two aircraft are located in different starting points ($\boldsymbol{p}_o$ and $\boldsymbol{p}_i$ as seen in Fig. 1 respectively) looking approximately in the opposite direction, since such configuration is easily achievable by flying the aircraft manually. The UAVs' initial position coordinates are pinned down in the inertial frame (North-East plane with arbitrary origin) as the pilots switch from manual to autonomous flight on their RC-transmitters. When the autopilot takes over, the UAVs are required to follow an orbit first (turn towards each other), then change to straight collision course. Their paths should intersect each other in one particular point and under a given angle α. This path geometry would eventually lead

⋆ This project was financed and carried out by MTA (Hungarian Academy of Sciences) Institute for Computer Sciences and Control

to the collision of the two aircraft in the intersection. Only horizontal trajectory will be designed, as altitude control is independent from trajectory tracking. There are advanced optimal control methods proposed by Riggi and D'Amico (2016) and Claeys et al. (2012) for space rendezvous flights, in this case, however, simple PID controllers for altitude-hold, course tracking and airspeed-hold loops were adequate along with incorporating guidance laws presented by Beard and McLain (2012). Flying on straight paths and on orbits are handled differently but the fundamental method is that the aircraft flies in a vector field that - should the aircraft deviate from the path - redirects it to the track with an effort related to the aircraft's distance from the path.

2.1 Obtaining geometry parameters

The aim of this section is to elaborate on the initial geometric situation and introduce the method for the trajectory design. Fig. 1 illustrates the instant when both aircraft are switched to autopilot mode and shows how both may turn either left or right from their initial positions. The camera-equipped aircraft is referred to as 'own aircraft', labeled as o, and the other is called 'intruder', labeled as i. The situation had to be initialized by some predefined geometry parameters to have the problem lead to an easily solvable system of equations. The predefined parameters initially known are the following:

- Fixed turning radius of the own UAV: R_o (defined to be above the minimal turning radius)
- Track angle of the own UAV after the turn: χ_{to} (can be set as the runway direction of the host airport)
- Initial positions of both UAVs: $\mathbf{p}_o = [x_o, y_o]^T$, $\mathbf{p}_i = [x_i, y_i]^T$
- Indicated - air relative - airspeed (IAS) and initial course angle of both UAVs: v_a^o, v_a^i, χ_o, χ_i
- Required angle between the two intersecting paths: α

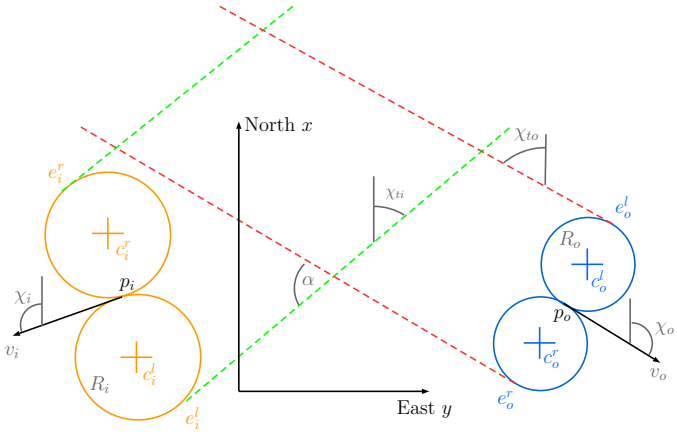

Fig. 1. Geometry of the aircraft's possible paths after turning left or right

First, the whole geometry of the own UAV can be designed using R_o and χ_{to}. The center point ($\mathbf{c}$) and transition (tangent) point ($\mathbf{e}$) coordinates can be obtained using the following equations where superscripts r and l are standing for 'right' and 'left' and 'o' denotes the own aircraft.

$$\begin{aligned}
\mathbf{c}_o^r &= \mathbf{p}_o + R_o \left[\cos\left(\chi_o + \frac{\pi}{2}\right), \sin\left(\chi_o + \frac{\pi}{2}\right)\right]^T \\
\mathbf{c}_o^l &= \mathbf{p}_o + R_o \left[\cos\left(\chi_o - \frac{\pi}{2}\right), \sin\left(\chi_o - \frac{\pi}{2}\right)\right]^T \\
\mathbf{e}_o^r &= \mathbf{c}_o^r - R_o \left[\cos\left(\chi_{to} + \frac{\pi}{2}\right), \sin\left(\chi_{to} + \frac{\pi}{2}\right)\right]^T \\
\mathbf{e}_o^l &= \mathbf{c}_o^l - R_o \left[\cos\left(\chi_{to} - \frac{\pi}{2}\right), \sin\left(\chi_{to} - \frac{\pi}{2}\right)\right]^T
\end{aligned} \tag{1}$$

Afterwards the course angle of the intruder's straight path can be obtained. Assuming that both UAVs are going to fly in the opposite direction in order to meet in the collision point, the difference between their course angles will be $\pm\pi$ for the opposite way and $\pm\alpha$ for the requested path angle.

$$\chi_{ti} = \chi_{to} \pm \pi \pm \alpha \tag{2}$$

The value of α is positive when the intruder's path crosses our path from the right and negative when it crosses from the left. It should be noted that when the track angles are compared both have to be considered in the $[0, 2\pi]$ interval.

The next step is to find the geometry of the intruder's path using the known variables. The collision point ($\mathbf{cp}$) coordinates can be derived from both aircraft's aspect, therefore starting from the tangent points we can write:

$$\begin{aligned}
\mathbf{cp} &= \mathbf{e}_{o_{(r/l)}} + S_o[\cos(\chi_{to}), \sin(\chi_{to})]^T \\
\mathbf{cp} &= \mathbf{e}_{i_{(r/l)}} + S_i[\cos(\chi_{ti}), \sin(\chi_{ti})]^T
\end{aligned} \tag{3}$$

S_o and S_i are parameters representing the length of the straight paths aircraft have to fly along between the tangent and the collision point. Since both UAVs may turn either left or right, $\mathbf{e}_o$ and $\mathbf{e}_i$ must be chosen accordingly. Considering the four possible turning scenarios (Right-Right (RR), Right-Left (RL), Left-Right (LR), Left-Left (LL)) four possible value-sets can occur for S_o and S_i, meaning that we need to choose the optimal route. The radius of the intruder should not exceed a sensible upper limit (e.g $150[m]$) and both aircraft must fly at least four seconds long on their straight legs. Furthermore, S_o, S_i and R_i must be all positive.

$\mathbf{e}_i = [x_{e_i}, y_{e_i}]^T$ from equation (3) can be parameterized with the intruder's turn radius(R_i), considering (1).

$$\mathbf{e}_i = \mathbf{p}_i + R_i \mathbf{A} \tag{4}$$

$\boldsymbol{A}$ is an auxiliary vector that has different values depending on the direction of the intruder's orbit (right or left).

$$\begin{aligned}
\mathbf{A}_r &= \begin{bmatrix} \cos\left(\chi_i + \frac{\pi}{2}\right) - \cos\left(\chi_{ti} + \frac{\pi}{2}\right) \\ \sin\left(\chi_i + \frac{\pi}{2}\right) - \sin\left(\chi_{ti} + \frac{\pi}{2}\right) \end{bmatrix} \\
\mathbf{A}_l &= \begin{bmatrix} \cos\left(\chi_i - \frac{\pi}{2}\right) - \cos\left(\chi_{ti} - \frac{\pi}{2}\right) \\ \sin\left(\chi_i - \frac{\pi}{2}\right) - \sin\left(\chi_{ti} - \frac{\pi}{2}\right) \end{bmatrix}
\end{aligned} \tag{5}$$

After the parametrization of the collision point, the flight times should be guaranteed to be equal. Both UAVs start on an orbit then switch to a straight path. The airspeed values (v_a^k), and the angles covered in orbital flight (ϑ_k) can be calculated from the predefined track angles and the initial course angles (see Beard and McLain (2012) for details). Thus the formula for the target flight time can be written as:

$$t_{tgt} = \frac{R_k \vartheta_k + S_k}{v_a^k} \qquad k = o, i \tag{6}$$

where s_k is the straight line length parameter.

Supposing that the collision points in equation (3) are

identical, after substituting the corresponding formulae containing the turn radius (as seen in equation (4)) the system of equations results as follows:

$$\begin{cases} x_{e_o} + S_o \cos(\chi_{to}) = x_i + R_i \mathbf{A}(1) + S_i \cos(\chi_{ti}) \\ y_{e_o} + S_o \sin(\chi_{to}) = y_i + R_i \mathbf{A}(2) + S_i \sin(\chi_{ti}) \\ \frac{R_o \vartheta_o + S_o}{v_a^o} = \frac{R_i \vartheta_i + S_i}{v_a^i} \end{cases} \tag{7}$$

(7) contains three equations with three unknown variables S_o, S_i, R_i, hence the best way to solve the problem is to parameterize S_o and S_i by R_i. After the parameterization the following equations are obtained where *sin* is abbreviated as s and *cos* as c.

$$\begin{aligned} S_o = & \underbrace{\frac{s\chi_{ti}(x_i - x_{e_o}) - c\chi_{ti}(y_i - y_{e_o})}{s(\chi_{ti} - \chi_{to})}}_{\mathrm{C}} - \\ & - R_i \underbrace{\frac{s\chi_{ti}A(1) - c\chi_{ti}A(2)}{s(\chi_{ti} - \chi_{to})}}_{\mathrm{D}} \\ S_i = & \underbrace{\frac{s\chi_{to}(x_i - x_{e_o}) - c\chi_{to}(y_i - y_{e_o})}{s(\chi_{ti} - \chi_{to})}}_{\mathrm{E}} - \\ & - R_i \underbrace{\frac{s\chi_{to}A(1) - c\chi_{to}A(2)}{s(\chi_{ti} - \chi_{to})}}_{\mathrm{F}} \end{aligned} \tag{8}$$

As we can see C, D, E and F are constants containing variables that are already known. Therefore, S_o and S_i can be substituted with the short forms from equation (8) into equation (7) and the solution for R_i can be obtained. The introduction of a new variable as the proportion of the two airspeed values ($W = \frac{v_a^i}{v_a^o}$) shall make the final equation more straightforward as well.

$$R_i = \frac{R_o W \vartheta_o + CW - E}{\vartheta_i + F - DW} \tag{9}$$

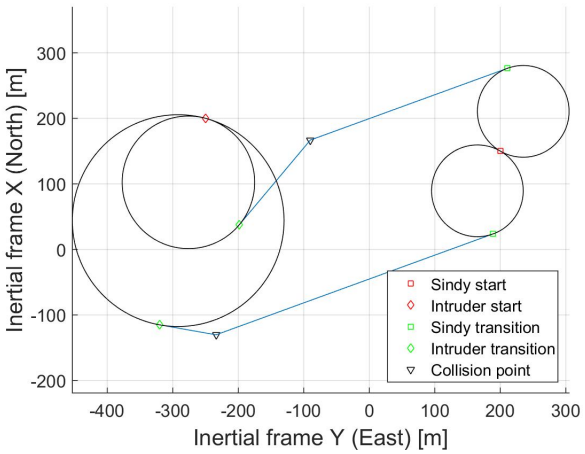

Fig. 2. Two examples of the generated trajectories

2.2 Singularities

Before executing the algorithm, the occurrence of singularities should be checked. Considering the forms of the equations in (3) we can rearrange the terms to the following shape:

$$\begin{bmatrix} x_{eo} - x_{ei} \\ y_{eo} - y_{ei} \end{bmatrix} = \begin{bmatrix} -\cos(\chi_{to}) & \cos(\chi_{ti}) \\ -\sin(\chi_{to}) & \sin(\chi_{ti}) \end{bmatrix} \begin{bmatrix} t_o \\ t_i \end{bmatrix} \tag{10}$$

Using the inverse of the 2×2 matrix above can lead to a singularity if the determinant of the matrix is equal to zero. The determinant of the 2×2 matrix:

$$Det = -c\chi_{to}s\chi_{ti} + s\chi_{to}c\chi_{ti} = 0 \tag{11}$$

Substituting χ_{ti} with equation (2) and reordering equation (11) we get that $Det = -s(\alpha + \pi)$. This means that the determinant is zero when the following is true:

$$\alpha + \pi = k\pi \qquad \alpha = \overline{k}\pi \qquad \overline{k} \in \mathbb{Z} \tag{12}$$

As seen from equation (12), singularity occurs only when the course angles of the two straight lines are either identical or shifted by a multiple of π. The incurred scenario implies that the two lines are parallel, and thus have infinite intersecting points, meaning there is no particular collision point. Figure 2 shows two samples of the algorithm-generated geometries plotted in the inertial frame. The angle of the intersecting straight paths are $-30°$ and $+30°$ respectively.

3. COMPENSATING THE EFFECTS OF WIND DISTURBANCES

During real flight testing wind gusts, turbulence and even thermals can affect the UAV in an undesirable way, reducing the accuracy of the time- and position-sensitive tests. However, if one is able to tune the autopilot well, it may increase the system's robustness against uncertainties and minor environmental impacts, such as wind gusts.
Nevertheless, the presence of steady wind will certainly affect the ground speed of the UAVs and cause considerable change in the flight time, since the trajectories are generated by an algorithm that does not take the effects of wind into account.

3.1 The wind triangle

The core of the wind correction is derived from the wind triangle, as it provides nexus between the vectors of the IAS, ground speed and wind. Actually the ground speed is the vector sum of the wind and IAS vectors.

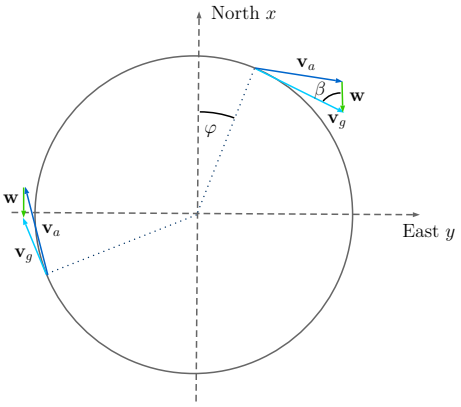

Fig. 3. Wind triangle in different orbital positions

It can be inferred from Fig. 3 that with the angle (β) between the ground speed vector ($\boldsymbol{v}_g$) and the wind vector

($\boldsymbol{w}$), the length of the IAS vector ($\boldsymbol{v}_a$) can be expressed using the law of cosines where all parameters are the function of the actual orbital angle (φ_{orb}). However, both the direction and the magnitude of the wind vector are unknown, therefore, assuming constant steady wind during the flight, one needs to find a way to determine the wind parameters.

3.2 Ground speed formulation

With the measured magnitude and direction of the wind the ground speed can be formulated using the law of cosines considering Fig. 3.

$$v_a^2 = v_g^2 + w^2 - 2v_g w\cos\beta$$

Solving the above equation for the ground speed the only acceptable root is yielded by the form:

$$v_g = w\cos\beta + \sqrt{v_a^2 - w^2\sin^2\beta} \tag{13}$$

3.3 Wind measurement

The literature (e.g. Langelaan et al. (2010)) contains advanced level methods describing in-flight wind field estimation. These methods are widely used on aircraft featuring complex on-board systems and sensors. However, the UAVs at SZTAKI lack these (heavy and expensive) equipment, therefore a simple wind measurement method, based on the continuous sampling of GPS ground speed and the respective orbital angle data while flying on an orbit in a steady wind, can be applied.
If one considers eq. (13) it seems clear that the wind parameters should be determined from a sine-like function with unknown amplitude (A_w) and phase-shift (φ_s).

$$v_g = A_w\sin(\varphi_{orb} - \varphi_s) + v_a$$

Hence, as both aircraft store flight data, the algorithm performs nonlinear curve fitting to the measured samples when the full circle had been flown. By choosing suitable initial values for A_w (1.0) and φ_s (0.0) the algorithm needs only a few ($\sim 4-9$) cycles to compute the unknown parameters. Ultimately, the absolute value of the amplitude will be equal to the wind's magnitude ($w = A_w$) and its direction - based on the sign of the amplitude - will be $\chi_w = \frac{3\pi}{2} - \varphi_s$ (if A_w is negative) and $\chi_w = \frac{\pi}{2} - \varphi_s$ (in case A_w is non-negative). Fig. (4) shows the result of curve fitting to a noisy ground speed sample.

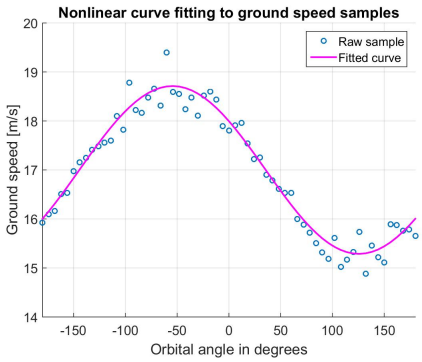

Fig. 4. Fitting a curve to ground speed samples

Originally, the IAS and the orbit radius of the intruder is different from the own aircraft's configuration, thus their orbit times differ as well. Therefore, in order to have the same flight time, while the own aircraft is flying on its wind measuring orbit the intruder has to follow a circle with a radius scaled to its originally computed one. The intruder's modified radius of the orbit can be formulated as:

$$R_i^{mod} = \frac{v_a^i t_{orbit}^i}{2\pi}$$

where $t_{orbit}^i = t_{orbit}^o = \frac{2\pi R_o}{v_a^o}$ is the own orbit time. The center point coordinates of the intruder's modified orbit can be expressed as seen in equation (1), using the modified radius, direction of orbit (λ) and initial position. Theoretically the resultant effect of the wind is going to be zero if the aircraft flies around on a whole orbit. This means that ideally the intruder's flight time on its scaled orbit will be the same as the own aircraft's on its own orbit. Both the SIL and HIL simulations have validated this assumption, as the two aircraft finished their respective orbits simultaneously even in highly turbulent environment.

3.4 The method of flight time compensation

Flying on the orbits means that the wind triangle is going to transform continuously, as β is changing. According to Fig. 3 the UAV's position on the arc can be represented by a φ angle; it is actually the course angle of the radial vector pointing from the center point to the position of the UAV. Similarly, $\varphi_w = \chi_w - \lambda\frac{\pi}{2}$ is the orbital angle of the tangentially placed wind vector. λ denotes the direction of the arc (1 for CW and -1 for CCW) and χ_t is the course (track) angle of the tangent line in the UAV's orbital position. While the ground speed vector is always tangential φ can be formulated as:

$$\varphi = \chi_t - \lambda\frac{\pi}{2}$$

Since $\beta = \varphi - \varphi_w$, as φ changes so will β. χ_w and φ_w are presumed to remain constant during the flight. The formula for the angle between the ground speed and wind vector writes as:

$$\beta = \mid \chi_t - \lambda\frac{\pi}{2} - \varphi_w \mid = \mid \chi_t - \chi_w \mid \tag{14}$$

β is a function of φ and the ground speed can actually be written as the multiple of the arc's radius (R) and the current angular speed along the orbit (ω). In addition, the angular speed is the time derivative of the angular displacement φ. Hence, the formula in equation (13) can be reshaped into a differential equation:

$$R\dot{\varphi} = w\cos\beta(\varphi) + \sqrt{v_a^2 - w^2\sin^2\beta(\varphi)} \tag{15}$$

The flight time compensating method is based on in-flight measurements. First of all, both aircraft need to record their flight time between the initial angular position and the orbit-straight transition point (final angular position) on the wind measuring orbit. The measured arc time is denoted as t_{arc}. The original (target) flight time, on which the trajectory design is based in equation (6), can be expressed as $t_{tgt} = t_{arc}^k + t_{line}^k$ where $t_{arc}^k = \frac{R_k\vartheta_k}{v_g^k}$, $t_{line}^k = \frac{S_k}{v_g^k}$ and S_k is the length of the UAVs' straight paths ($k = o, i$).
As for the intruder, it does not fly through its original

transition point, since it is flying on a scaled orbit, therefore its measured arc time should be scaled back to its arc with the original radius. This can be approximated as $t^i_{arc} = t^i_{arc_{meas}} \frac{R_i}{R^{mod}_i}$. This linear correlation approximation between the flight times originates from the expression $t_{arc} = \frac{R\Delta\vartheta}{\overline{v}_g}$. Given that the airspeed is fixed, the average value of the ground speed $\overline{v}_g$ can be considered constant on the same arc with different radii. Fig. 3 indicates that if the wind vector is unaltered and the IAS is regulated to remain constant then the actual ground speed vector will also stay the same regardless of the arc's radius at a given φ position on the orbit. Using the measured arc time of the own aircraft and the scaled arc time of the intruder, the following system of equations can be constructed. The only unknown variable in (16) will be the IAS value $(v^k_a)'$ that is needed for keeping the target flight time in windy conditions.

$$\begin{cases} t_{tgt} = t^k_{arc} + t^k_{line} \\ t^k_{arc} = t^k_{arc_{meas}} \frac{\overline{v}^k_g(v^k_a)}{(\overline{v}^k_g)'((v^k_a)')} \approx t^k_{arc_{meas}} \frac{v^k_a}{(v^k_a)'} \qquad k = o, i \\ t_{line} = \frac{S_{track}}{v_g\left(v'_a\right)} \end{cases} \tag{16}$$

In the second equation, it is assumed, that when the IAS changes, the registered ground speed curve shifts accordingly. The fundamental idea is that the mean of the ground speed sample (Fig. 4) for the whole circle will be the predefined IAS. Therefore, between a particular pair of orbital angles (φ_1 & φ_2), the difference between the average of the sampled ground speed values ($\overline{v}^k_g$ and $(\overline{v}^k_g)'$) will be the same as the difference of the two airspeed (v^k_a and $(v^k_a)'$). Furthermore, since the difference of the airspeed and average ground speed values are relatively small, their ratios are considered the same (prevailing in the second equation), resulting a maximum of 4% error (but mostly $\sim$ 1%).In the third equation, $v_g\left(v'_a\right)$ should be accounted as seen in equation (13), where v'_a is the unknown, new IAS and the wind is measured. The system of equations leads to a fourth-order function of v'_a.

$$t^2_{tgt}(v^k_a)'^4 - 2J_k t_{tgt}(v^k_a)'^3 + (J^2_k - t^2_{tgt}M - N^2_k)(v^k_a)'^2 + \\ + (2J_k t_{tgt}M + 2N_k L_k)(v^k_a)' + (-J^2_k M - L^2_k) = 0$$

$$J_k = t^k_{orb} v^k_a \qquad K = t_{tgt} w \cos(\alpha) \qquad L_k = J_k w \cos(\alpha) \\ M = w^2 \sin^2(\alpha) \qquad N_k = K - S_k \qquad k = o, i \tag{17}$$

The suitable root of equation (17) can be easily determined using the Newton-Raphson method. The original IAS (v^k_a) should be chosen as a first approximation of the root hence the method will yield the closest root to this value.

4. MONTE CARLO SIMULATION

After deriving the formulae for the trajectory design and wind compensation, a Monte Carlo SIL simulation was executed to test the applicability of the methods. The aircraft and their control systems were modeled as in Beard and McLain (2012). The randomly predefined features in this test are the initial position coordinates, the IAS values, the initial course angles and the wind parameters. All of them are generated inside a specified domain of suitable values (visualized in Fig. 5):

- Each UAV's initial position coordinates are restricted into two different $100m \times 100m$ squares in the North-East plane of the inertial frame
- The initial course angles are generated between 45° and 135° for the 'own' aircraft and between −45° and −135° for the intruder
- The airspeed values are defined between $18 - 23\left[\frac{m}{s}\right]$
- The steady wind magnitude might be between $0 - 3\left[\frac{m}{s}\right]$ and its direction can have any value in the interval of $\left[-180°, 180°\right]$
- The orbit radius $(150 - 250[m])$ and course angle (depends on the runway's orientation) of the straight path are fixed for the own aircraft

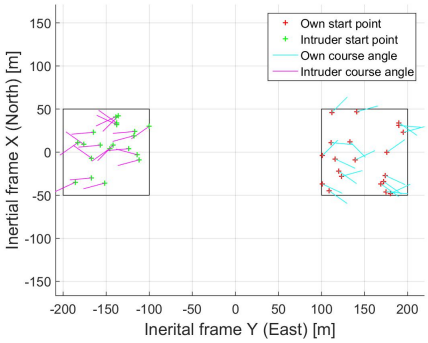

Fig. 5. Initial positions and course angles generated randomly

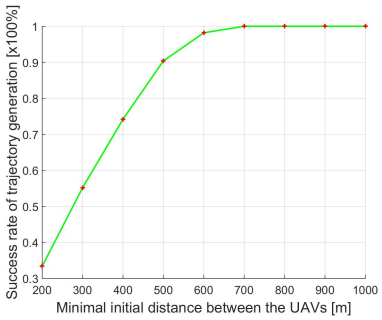

Fig. 6. Average success rate of trajectory generation

After the random variables were set, the algorithm tried to generate a suitable trajectory for path angles −45°, −15°, 15° and 45°; $\alpha = 0°$ had to be omitted because of the occurrence of singularities as derived in the previous section. Nevertheless, the algorithm was checked for 25 possibilities with the four α configurations each, which resulted in a 100 samples. The trajectory generation was considered successful if the requirements stated in section 2, for the selection of an optimal path, were fulfilled. The number of successful generations depended on the initial distance between the two aircraft. The further the aircraft were positioned, the higher the success rate grew, since the radius and straight path parameters could vary with greater freedom.

Fig. 6 illustrate the average success rate of trajectory builds within a set of five simulations (100 random samples each) for different minimum start distances $(200, 300, \ldots [m])$ between the aircraft. However, during outside tests the spatial constraints must be taken into

account, as the boundaries of the airspace shall not be violated. In case of MTA SZTAKI, the greatest minimum distance should be around $300[m]$ during real-life testing.

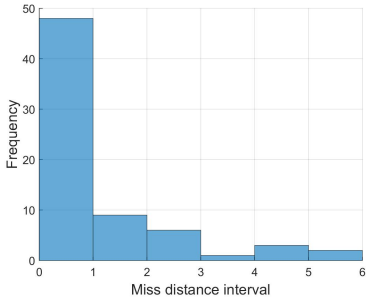

Fig. 7. Absolute frequency histogram of the miss distance samples

One randomly selected simulation yielded 69 out of 100 successful generations. If the miss distance samples are divided into intervals of one meter, the absolute frequency histogram will look like as shown in Fig. 7. In the majority of cases the miss distance samples are under one meter, only centimeters from the target zero distance. Therefore, it might be confidently stated that the wind compensation algorithm performed well in the SIL tests, as the avoidance maneuver is triggered under an intruder proximity of 10 meters. As for the tail of the histogram, the uniform regression is quite straightforward, despite the gap between $3-4$ meters which can be neglected.

5. HIL SIMULATION

Since the final goal of the project is to test the avoidance algorithm in real-flight conditions, the SIL-validated models were implemented to be tested in HIL simulations. The mathematical models of the aircraft were handled in MATLAB on two individual PCs. However, the algorithms for guidance, control, path generation, collision detection and avoidance were running on two separate MPC5200 microcontrollers. They communicate via XBee modules with an effective range of only a few hundred meters. Therefore, we need a ground antenna that transmits messages between the two aircraft.

We have conducted successful HIL simulations with the following schedule: firstly, the camera-equipped aircraft switches to autopilot, and waits for the intruder's signal that it also have switched; until then it follows a straight line without alterations. As soon as the intruder's signal is received (along with GPS position and course angle data) both aircraft flies a full circle while storing the necessary data for the sine fitting algorithm, thus acquiring wind parameters. However, while on this full circle, the own aircraft calculates the collision trajectory based on data received from the intruder. After the wind parameters are determined both aircraft switch to their own collision paths while maintaining an compensatory IAS that is required for meeting at the same time. Videos presenting the successful HIL-simulations can be found at the *Aero GNC* YouTube channel.

6. CONCLUSION

The article proposed a possible trajectory design method for generating a flexible test track for SZTAKI's Sense and Avoid algorithms. At first, the equations for the design of intersecting paths were derived while presuming calm (windless) weather conditions. Then the calculations were modified to guarantee the collision with compensating the effect of steady wind. HIL-simulations had validated these algorithms in both calm and windy environment and the camera-equipped aircraft performed successful avoidance maneuvers when needed.

REFERENCES

Alturbeh, H. (2014). Collision Avoideance Systems For UAS Operating In Civil Airspace.

Bauer, P., Hiba, A., Vanek, B., Zarandy, A., and Bokor, J. (2016). Monocular Image-based Time to Collision and Closest Point of Approach Estimation. In *In proceedings of 24th Mediterranean Conference on Control and Automation (MED'16)*. Athens, Greece.

Beard, R.W. and McLain, T.W. (2012). *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press.

Claeys, M., Arzelier, D., Henrion, D., and Lasserre, J.B. (2012). Measures and LMI for impulsive optimal control with applications to space rendezvous problems. *American Control Conference (ACC)*.

Degen, S. (2011). *Reactive Image-based Collision Avoidance System for Unmanned Aircraft Systems*. Master's thesis, Australian Research Centre for Aerospace Automation.

Forlenza, L. (2012). *Vision based strategies for implementing Sense and Avoid capabilities onboard Unmanned Aerial Systems*. Ph.D. thesis, UNIVERSITÁ DEGLI STUDI DI NAPOLI FEDERICO II.

Kim, K.Y., Park, J.W., and Tahk, M.J. (2007). UAV Collision Avoidance Using Probabilistic Method in 3-D.

Langelaan, J.W., Alley, N., and Neidhoefer, J. (2010). Wind Field Estimation for Small Unmanned Aerial Vehicles. In *AIAA Guidance, Navigation and Control Conference, Toronto, Canada*.

Luo, Y.Z., Zhang, J., Li, H.Y., and Tang, G.J. (2010). Space Rendezvous and Docking Mission Planning System Using Object-Oriented Method. In *AIAA Modeling and Simulation Technologies Conference, Toronto, Ontario, Canada*.

Melnyk, R., Schrage, D., Volovoi, V., and Jimenez, H. (2014). Sense and Avoid Requirements for Unmanned Aircraft Systems Using a Target Level of Safety Approach. *Risk Analysis*, 34(10), 1894–1906.

Richards, A. and How, J. (2002). Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming.

Riggi, L. and D'Amico, S. (2016). Optimal Impulsive Closed-Form Control for Spacecraft Formation Flying and Rendezvous. In *American Control Conference (ACC)*.

Zsedrovits, T., Bauer, P., Pencz, B.J.M., Hiba, A., Gőzse, I., Kisantal, M., Németh, M., Nagy, Z., Vanek, B., Zarándy, A., and Bokor, J. (2016). Onboard Visual Sense and Avoid System for Small Aircraft. *IEEE A&E SYSTEMS MAGAZINE*, 31, 18–27.