



## AUTONOMOUS UNMANNED AERIAL VEHICLE FLIGHT ACCURACY EVALUATION FOR THREE DIFFERENT PATH-TRACKING ALGORITHMS

Ramūnas KIKUTIS<sup>1\*</sup>, Jonas STANKŪNAS<sup>2</sup>, Darius RUDINSKAS<sup>3</sup>

<sup>1</sup>*Dept of Aviation Technologies, Vilnius Gediminas Technical University, Lithuania*

<sup>2,3</sup>*Dept of Aeronautical Engineering, Vilnius Gediminas Technical University, Lithuania*

Received 1 February 2017; revised 21 July 2017, 12 September 2017; accepted 1 November 2017

**Abstract.** This paper shows mathematical results of three methods, which can be used for Unmanned Aerial Vehicle (UAV) to make transition from one flight leg to another. In paper, we present general equations, which can be used for generating waypoint-switching methods when for experiment purpose mathematical UAV model is used. UAV is modelled as moving dot, which eliminates all of the aerodynamics factors and we can concentrate only on the navigation problems. Lots of attention is dedicated to show possible flight path error values with representation of modelled flight path trajectories and deviations from the flight mission path. All of the modelled flight missions are done in two-dimensional space and all the results are evaluated by looking at Probability Density Function (PDF) values, as we are mostly interested in the probability of the error.

**Keywords:** navigation, Dubins paths, waypoint-switching method, flight path error, unmanned aerial vehicle, dynamic model.

### Introduction

Present research shows lots of interest in Unmanned Aerial Vehicle (UAV) navigation, guidance, path tracking, mission and path planning algorithms (Maillot *et al.* 2015). Research in these fields include UAV path planning under uncertainties, coordinated flight control, detection of ground obstacles, mapping etc. On the other hand, at this moment small unmanned aerial vehicles have lots of restrictions for their flights released by national or international legal organizations. Due to this, new innovations in previously mentioned research fields make big importance for the future UAV integration into national airspace (Zhong, Yan 2014). Altogether, with the integration of UAVs into national airspace it would allow us to use small UAVs for various practical applications like delivery of small cargo or electrical line monitoring, forest fire fighting (Casbeer *et al.* 2006).

However, integration of the UAVs into national airspace requires more detailed evaluation not just of the new and more sophisticated UAV autonomous navigation or path tracking algorithms, but also their accuracy (Dadkhah, Mettler 2012). Increase in accuracy could be later used for search and avoid algorithm, which is useful for ground obstacle avoidance. Otherwise, with current

small UAV technology any failure to detect the obstacle or other airplane leads to an accident (Zhong, Yan 2014). The same is true for the guidance part of the search and avoid algorithm (Wang *et al.* 2016). For this reason, we need to pay more attention to flight path accuracy. In papers by Stojcsics (2014), Ariff and Go (2011), Capello *et al.* (2017), ideas of calculating flight path or cross track error is proposed, but not much attention is paid to look at the flight path error tendencies. In some of the papers, we can find that different track change algorithms could be used. Mostly all of the time new methods to use Dubins paths are being proposed (Owen *et al.* 2015). However, various modifications of these Dubins paths could be used too. In paper authors propose to use flight over navigation points, around waypoint or to use internally-tangent flight trajectory near the waypoint (Capello *et al.* 2017). On the other hand, still not much attention is paid to flight accuracy.

In order to reduce restrictions for UAV flights and to increase the knowledge of small UAV flight accuracy, the goal of this research is to evaluate general tendency of the three different UAVs autonomous waypoint-switching algorithm and their accuracy for UAV navigation during the flight mission.

\*Corresponding author. E-mail: [ramunas.kikutis@vgtu.lt](mailto:ramunas.kikutis@vgtu.lt)

### 1. Autonomous flight mission execution model of the UAV

In this chapter, we present the UAV autonomous flight model used for further experiments. Block diagram of this model is showed in Figure 1. It has three structural blocks as described in research by Benghezal *et al.* (2015). Path follow block is used to describe UAV flight dynamics and it contains all of the differential equations, which simulate UAV as a single mass object. From path follow block, we get all navigation data like UAV position, speed, heading, and flight path error. In this research, we choose single mass UAV model because it eliminates part of flight path error, which is caused due to aerodynamics. This way we can evaluate accuracy of the waypoint-switching algorithms more accurately.

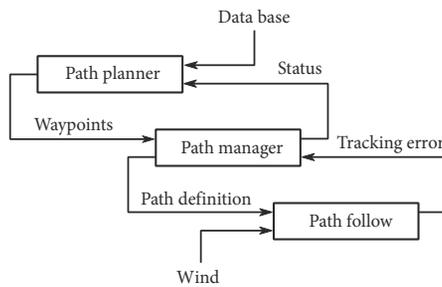


Figure 1. Block diagram of UAV autopilot, which is used to carry out autonomous flight mission

The second model block is path manager block. Previously calculated flight path error is provided for the path manager block. Flight path error value is used to calculate new UAV heading, which is used to correct and compensate flight path error (Nelson *et al.* 2006; Wang *et al.* 2016). Furthermore, this block keeps the information of which waypoint-switching algorithm is used and needs an input about the planned flight mission with all of the coordinates or waypoints. This information should arrive from the path planner block, which completes full circle of automation.

If we would look at this model from the perspective that the UAV is not able to make decisions according the environmental details like buildings, trees or other obstacles, we can eliminate the path planner block (Chamseddine *et al.* 2012). In our case, we are not using path planner block and all of the waypoints will be unchangeable during the mission.

### 2. Principles of UAV autonomous guidance

In this chapter, we present the UAV autonomous guidance method, which is used inside of this model for further research.

For this research, the optimised Frenet–Serret coordinate frame is used. In our situation, we put this coordinate frame inside of local North–East–Down (NED) frame. General idea of our coordinate frames and UAV guidance method are presented in Figure 2. If the UAV is flying in

between two waypoints  $w_i$  and  $w_{i+1}$  and is off track to the side by distance  $e_N$  upwards or downwards from the defined trajectory, the vectorial field attracts the UAV back to the path. More away from the path UAV is stronger vector field is (Wang *et al.* 2016).

This optimized and simplified UAV guidance vector field can be calculated by Equation (1):

$$X_d(e_N) = -X^\infty \cdot \frac{2}{\pi} \cdot \frac{1}{\tan(k_p \cdot e_N)}, \tag{1}$$

where:  $X_d$  – desired flight course;  $e_N$  – flight path deviation;  $X$  – flight course at infinite distance from the flight path;  $k_p$  – Proportional Integral Derivative (PID) controller coefficient.

From Equation (1) we can see that the course  $X_d$  is the function of flight path deviation  $e_N$  and in calm conditions  $X_d = \eta$ , where  $\eta$  – the flight course after wind drift compensation, which represents airplane heading to the next waypoint or to some intermediate point on the flight leg like  $P_2$ , which is shown in Figure 3.

However, this straight-line tracking can be represented even without local NED frame. As this coordinate system is always attached to initial waypoint  $w$  for each flight segment, we do not need to make coordinate transformations to NED.

In that case, the modulus of the flight path error  $e_N$  can be calculated by Equation (2).

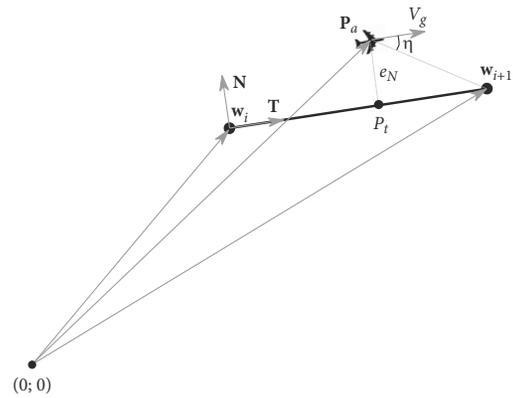


Figure 2. Simplified idea of autonomous UAV guidance

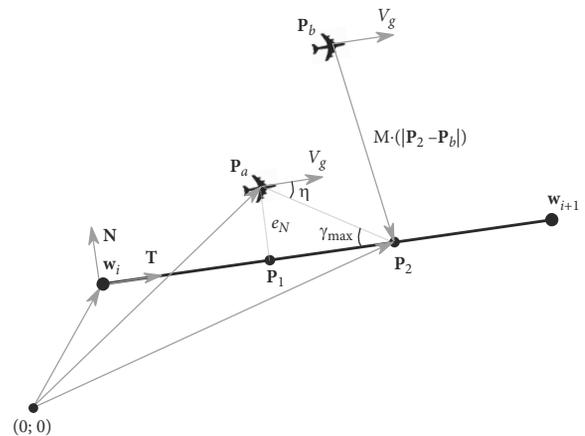


Figure 3. Full idea of autonomous UAV guidance

In Equation (2) we do not evaluate UAV vertical position and do all calculations only in 2D space:

$$|e_N| = \left( \frac{\begin{bmatrix} -y_T & x_T & 0 \\ -y_T & x_T & 0 \end{bmatrix}^T}{\begin{bmatrix} -y_T & x_T & 0 \\ -y_T & x_T & 0 \end{bmatrix}} \right) \cdot (\mathbf{P}_a - \mathbf{w}_i), \quad (2)$$

where:  $e_N$  – flight path error (deviation);  $x_T$ ,  $y_T$  – the lengths of the flight leg  $\mathbf{w}_i\mathbf{w}_{i+1}$  in  $x$  and  $y$  directions;  $\mathbf{P}_a$  – vector representing current UAV location;  $\mathbf{w}_i$  – waypoint from which UAV has departed.

Next, by using vector form we can find the remaining distance from the current position to the next waypoint:

$$\mathbf{P}_a\mathbf{w}_{i+1} = \mathbf{w}_i\mathbf{w}_{i+1} - \mathbf{P}_a, \quad (3)$$

where:  $\mathbf{P}_a\mathbf{w}_{i+1}$  – line connecting UAV location  $\mathbf{P}_a$  and the second waypoint  $\mathbf{w}_{i+1}$  in the UAV route;  $\mathbf{w}_i\mathbf{w}_{i+1}$  – line connecting first waypoint  $\mathbf{w}_i$  and the second waypoint  $\mathbf{w}_{i+1}$  in the UAV route;

$$\mathbf{P}_a\mathbf{w}_{i+1} = \left| \mathbf{P}_i\mathbf{w}_{i+1} \right| \cdot \frac{(\mathbf{w}_{i+1} - \mathbf{w}_i)}{|\mathbf{w}_{i+1} - \mathbf{w}_i|} - |e_N| \cdot \frac{\begin{bmatrix} -y_T & x_T & 0 \\ -y_T & x_T & 0 \end{bmatrix}^T}{\begin{bmatrix} -y_T & x_T & 0 \\ -y_T & x_T & 0 \end{bmatrix}}, \quad (4)$$

where:  $\mathbf{P}_i\mathbf{w}_{i+1}$  – line connecting UAV location projection on the flight path route point  $\mathbf{P}_i$  and the second waypoint  $\mathbf{w}_{i+1}$  in the UAV route.

If we know vector from the present UAV coordinate to the next waypoint, we can make the appropriate change of the heading. On the other hand, by using this method track error can become too big and it would lead to a big overshoot. For this reason, additional coefficient  $M^*$  can be used to correct that problem as shown in Figure 3. For this situation, correction should be calculated as in Equation (5):

$$\mathbf{D}_{\min} = \min(\mathbf{D}, |\mathbf{L}| \cdot M^*), \quad (5)$$

where:  $\mathbf{D}$  – vector from the current position on the track to the point on the track where UAV needs to return represented as  $\mathbf{P}_i\mathbf{w}_{i+1}$  in Figure 2 or as  $\mathbf{P}_1\mathbf{P}_2$  in Figure 3;  $\mathbf{L}$  – vector from the position of the UAV to the point of return to the track as  $\mathbf{P}_a\mathbf{w}_{i+1}$  in Figure 2 or as  $\mathbf{P}_b\mathbf{P}_2$  in Figure 3.

In our case, we will use optimized vector field with the reference coordinate NED frame. This method gives smaller overshoots and is more reliable (Beard, Humpherys 2011).

### 3. Three methods of switching UAVs mission waypoints

In this chapter, we introduce three different methods for UAV to make waypoint transitions from one flight leg to another flight leg (Beard, Humpherys 2011). All three waypoint-switching methods later will be compared and flight accuracy will be evaluated by using UAV flight simulation (Capello et al. 2009).

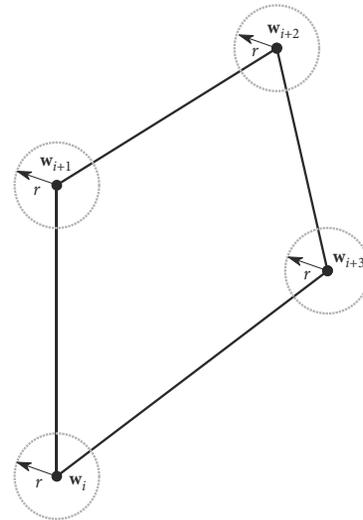


Figure 4. Visualization of the classical UAV autonomous navigation algorithm when the airplane makes waypoint transition by entering predefined radius sphere

The first of the methods is the classical, which is used till today in most of the autopilot systems. The idea is presented in Figure 4. In this case, UAV makes transition from one waypoint to another by entering the predefined sphere, which has the radius  $r$ .

This algorithm can also be represented by simple Equation (6). It states what if the difference between the coordinates of UAV and the coordinates of the waypoint is equal or less than the predefined sphere radius, UAV makes the transition.

$$|\mathbf{p}(t) - \mathbf{w}_{i+1}| \leq r, \quad (6)$$

where:  $\mathbf{p}(t)$  – the coordinates of UAV;  $\mathbf{w}_{i+1}$  – coordinates of the next waypoint;  $r$  – radius of the sphere.

This algorithm has an advantage of simplicity, but it has much more disadvantages:

- the trajectory is unpredictable inside the sphere;
- if the radius of the sphere is too big, accuracy of the flight can greatly reduce;
- if the radius of the sphere is too small, UAV can never reach the waypoint and transition will not be made;
- it does not present real flight path of the UAV inside of the sphere.

In case to eliminate some of the disadvantages, another algorithm could be used. Second algorithm does not depend on the radius of the sphere. It depends on the position of the plane  $H$ . This algorithm is represented in Figure 5. In this case, UAV makes the transition from one waypoint to another by crossing the predefined plane  $H$ . Because plane  $H$  is infinite, UAV will never miss the waypoint. The accuracy of the flight near the waypoint depends only on the accuracy of the UAV, but not on the algorithm itself.

For this algorithm, we need to mathematically define plane  $H$ . In order to do this, few unit vectors  $\mathbf{q}$  are being used. Vectors  $\mathbf{q}_i$  and  $\mathbf{q}_{i+1}$  in Equations (7) and (8) show direction of the present and future flightpath. Both vectors

are calculated by using coordinates of three waypoints: past, present and future.

$$\mathbf{q}_i = \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|}, \quad (7)$$

where:  $\mathbf{w}_i$  – coordinates of the previous waypoint;  $\mathbf{w}_{i+1}$  – coordinates of the current waypoint;  $\mathbf{q}_i$  – unit vector showing the direction of the current flight leg;

$$\mathbf{q}_{i+1} = \frac{\mathbf{w}_{i+2} - \mathbf{w}_{i+1}}{\|\mathbf{w}_{i+2} - \mathbf{w}_{i+1}\|}, \quad (8)$$

where:  $\mathbf{w}_{i+2}$  – coordinates of the next waypoint;  $\mathbf{q}_{i+1}$  – unit vector showing the direction of the next flight leg.

Next, the unit vector  $\mathbf{n}$  is calculated, which can be used to find the orientation of the plane  $H$ . Vector  $\mathbf{n}$  is calculated by Equation (9) using values of unit vectors  $\mathbf{q}_i$  and  $\mathbf{q}_{i+1}$ . The plane  $H$  always divides angle  $a$ , which is in between two flight mission segments  $\mathbf{w}_i\mathbf{w}_{i+1}$  and  $\mathbf{w}_{i+1}\mathbf{w}_{i+2}$ .

$$\mathbf{n}_i = \frac{\mathbf{q}_i + \mathbf{q}_{i+1}}{\|\mathbf{q}_i + \mathbf{q}_{i+1}\|}, \quad (9)$$

where:  $\mathbf{n}_i$  – unit vector showing orientation of the designed plane  $H$ .

The Equation (10) tells what if the UAV position is on one of the plane  $H$  coordinates or beyond it, UAV must make a turn:

$$\mathbf{p}(t) \in H(\mathbf{w}_{i+1}, \mathbf{n}_i). \quad (10)$$

While this method in general looks much better, we still do not know is it more accurate than the classical one or not. On the other hand, it still can be upgraded and made more sophisticated.

The third of the waypoint transition methods is made by using Dubins paths (Yeol, Hwang 2016; Owen *et al.* 2015). It is composed by using straight-line and circular arc segments. We can see Dubins paths transition method in Figure 6.

For Dubins paths waypoint-switching algorithm we need to make some additional calculations. First of all, angle  $a$  must be found. It depends on the geometry of the flight path and shows the difference between two straight-line segments of the flight mission. It can be calculated by using Equation (11):

$$\angle a = \frac{1}{\cos(-\mathbf{q}_i^T \cdot \mathbf{q}_{i+1})}, \quad (11)$$

where:  $a$  – the angle showing difference between the current and the next flight legs.

When we have the value of angle  $a$ , it can be used for calculation of distance vector  $\mathbf{k}$ . This vector is added or subtracted from the next waypoint  $\mathbf{w}_{i+1}$  and gives the coordinates where the planes  $H_1$  and  $H_2$  should be located. Finally, by using Equation (14) we can find centre position of the imaginable Dubins circle. It is used for generating circular flight segment. By changing the radius of this circle we can increase or decrease the smoothness of the transition from one flight leg to another flight leg.

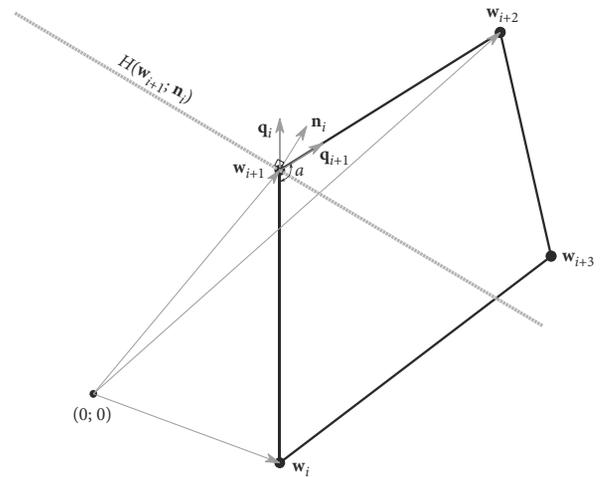


Figure 5. Visualization of UAV autonomous navigation algorithm when the airplane makes waypoint transition by crossing the plane ( $H$  – plane for waypoint transition;  $\mathbf{w}_i$  – coordinates of the waypoint;  $\mathbf{q}_i$  – unit vector, which show the direction of the flight leg;  $\mathbf{n}_i$  – unit vector for orientation of plane  $H$ )

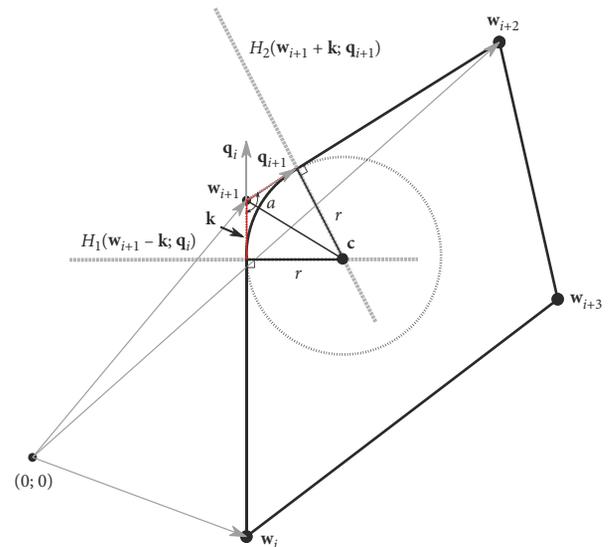


Figure 6. Visualization of UAV autonomous navigation algorithm when the airplane makes waypoint transition by using Dubins trajectory ( $H_1$  and  $H_2$  – planes where the transition from straight flight segment into circular (or opposite) flight segment should happen;  $\mathbf{c}$  – coordinate of Dubins circle centre;  $r$  – radius of the Dubins circle)

$$-\mathbf{k} = -\left(\frac{r}{\tan \frac{\angle a}{2}}\right) \cdot \mathbf{q}_i, \quad (12)$$

where:  $\mathbf{k}$  – vectorial distance showing at which coordinate from the next flight mission waypoint UAV must start a turn into the next flight leg.

$$\mathbf{k} = -\left(\frac{r}{\tan \frac{\angle a}{2}}\right) \cdot \mathbf{q}_{i+1}. \quad (13)$$

By using further Equation (14) coordinates of the turning trajectory circle can be calculated. It will help to visualise the geometry of the predicted flight trajectory during the turn.

$$\mathbf{c} = \mathbf{w}_{i+1} - \left( \frac{r}{\sin \frac{\angle a}{2}} \right) \cdot \frac{\mathbf{q}_i - \mathbf{q}_{i+1}}{\|\mathbf{q}_i - \mathbf{q}_{i+1}\|}, \quad (14)$$

where:  $\mathbf{c}$  – coordinates of the circle representing the UAV flight path during the turn.

It is useful to mention that Dubins paths waypoint-switching method has an advantage over the classical or “plane” method by providing the most predictable UAV flight path. We can still expect to see some UAV drift to the side from the flight path, but we can predict that the biggest flight path error in this case would appear due to an algorithm itself. We can see that for this method there is also one disadvantage – UAV will never fly directly above the waypoint. The circular arc – fillet (Kothari *et al.* 2014) does not allow that.

#### 4. Flight path accuracy results

In this chapter, we will look at the experiment results when comparing all three waypoint-switching methods.

Different approaches can be used to evaluate flight path accuracy (Brezoescu *et al.* 2011; Avellar *et al.* 2015; Li *et al.* 2015). However, our idea was to choose random constant speed of 10 m/s for UAV flight (which is as normal flight speed for small UAV) and compare the flight path error values for all free waypoint-switching methods while the same triangular route was used. All of the flight legs are 500 m long. From previous research we know, that the increase or decrease in speed has the same influence on the flight path error. It means if the speed increases, UAV flight path error will also increase or vice versa. Flight path error dependency on flight speed is presented in Figure 7.

From Figure 7 we find that for any waypoint-switching method, flight path error will always increase by quadratic tendency. In this experiment, we want to check how flight path error changes due to changing waypoint-switching method and the radius of the sphere or circle. We choose to make changes to the sphere radius size because it is the main variable of the waypoint-switching algorithm.

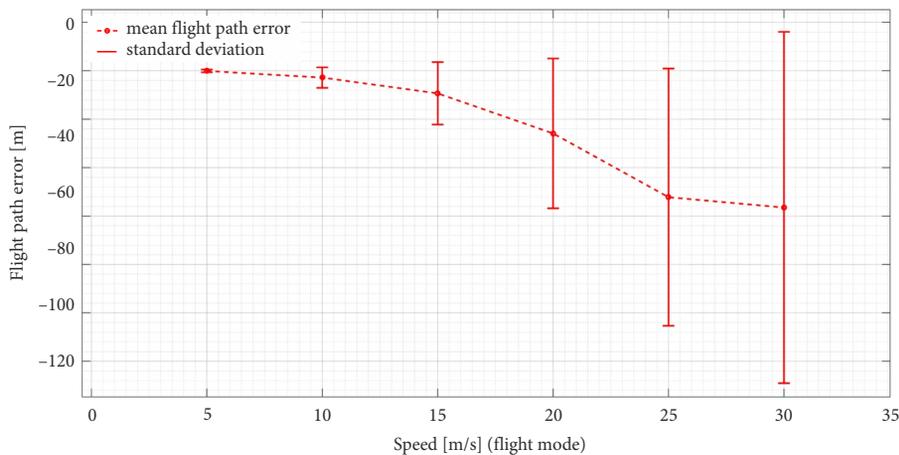


Figure 7. Average flight path error values and standard deviation ranges for different airspeeds

Changes to this radius can lead to smaller or bigger overshoots during the waypoint transition.

Experiment results are presented in Figure 8. Results of switching waypoints by using “plane” method are in all cases the same. It is because there is no radius involved in this algorithm. We only compare “plane” switching method result with all other algorithm combinations when using other waypoint-switching sphere or radius sizes. For smaller radius values, we get more accurate results, when fillet technique is used. For 10 m/s speed, it becomes optimal for the radius of 20 m. However, classical method with an increase of the sphere radius becomes more accurate and if we would compare results for 30 m radius, most accurate is the classical algorithm. In this case, filleted Dubins paths algorithm is not an optimal choice. Besides that, switching the waypoints by using “plane” method, proves that it gives biggest overshoots and it would give the same results as the situation of the classical algorithm with 0 m radius sphere.

All of the flight path error results can be presented in a time line graphs. For general visualization purpose we include two extreme cases for 5 m radius and 30 m radius experiments. In Figure 9, we can compare flight path error values for different waypoint transition algorithms more precisely. It is clear that average flight path error with 5 m radius is smaller for Dubins paths waypoint-switching algorithm, but for another extreme case, classical waypoint-switching method would give smaller average flight path error.

From the data in Figure 9, it is not possible to see probability of the flight path error if one or another waypoint transition method is used. For this reason and to prove our results we choose to check one more time with Probability Density Function (PDF). General idea of this method can be expressed in Equation (15):

$$\rho(x) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{x^2}{2}}, \quad (15)$$

where:  $x$  is flight path error value;  $\rho(x)$  represents probability density for this error to appear.

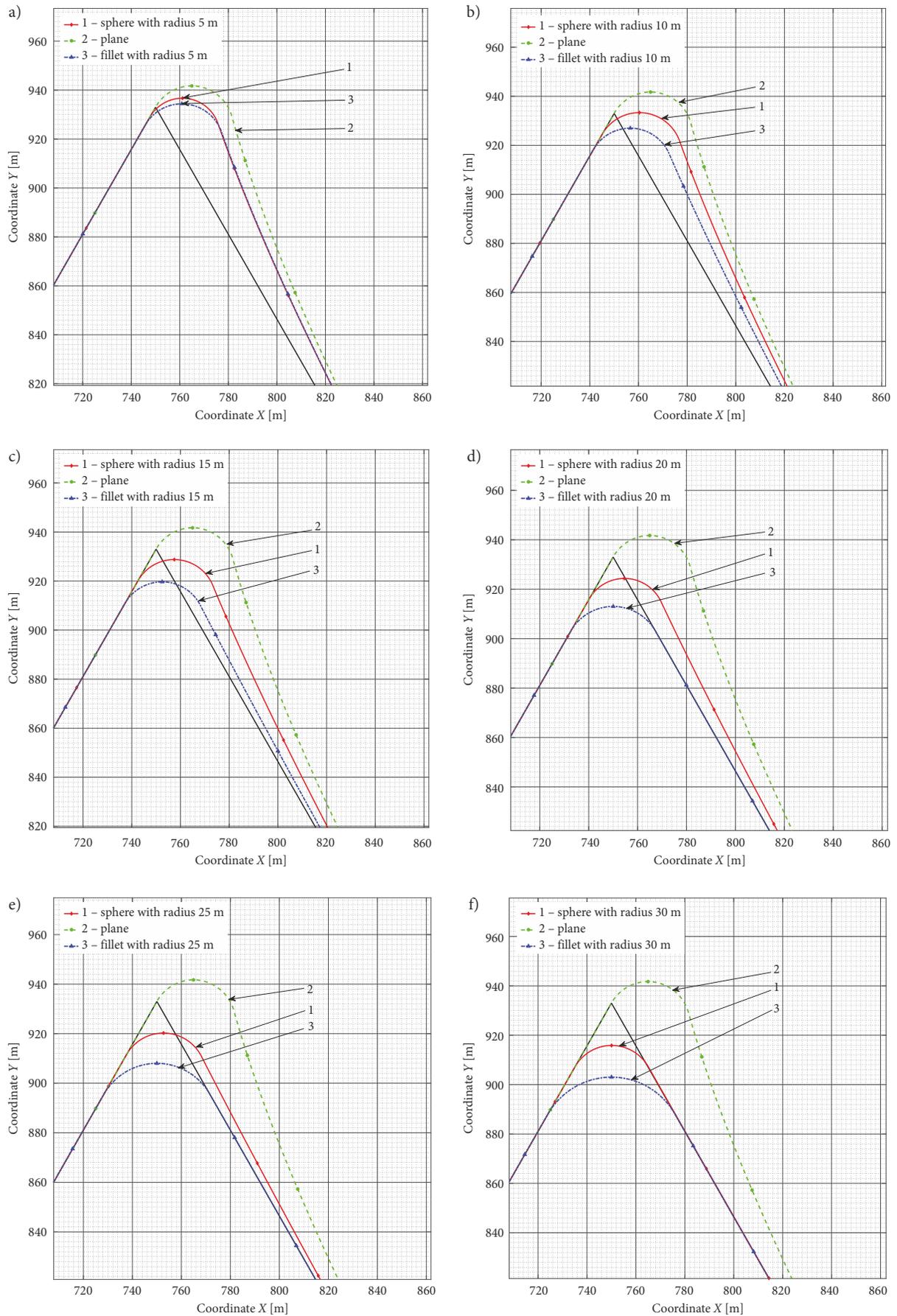


Figure 8. Flight path error values for the three waypoint-switching methods when radius  $r$  of the sphere or circle is used: a -  $r = 5$  m; b -  $r = 10$  m; c -  $r = 15$  m; d -  $r = 20$  m; e -  $r = 25$  m; f -  $r = 30$  m

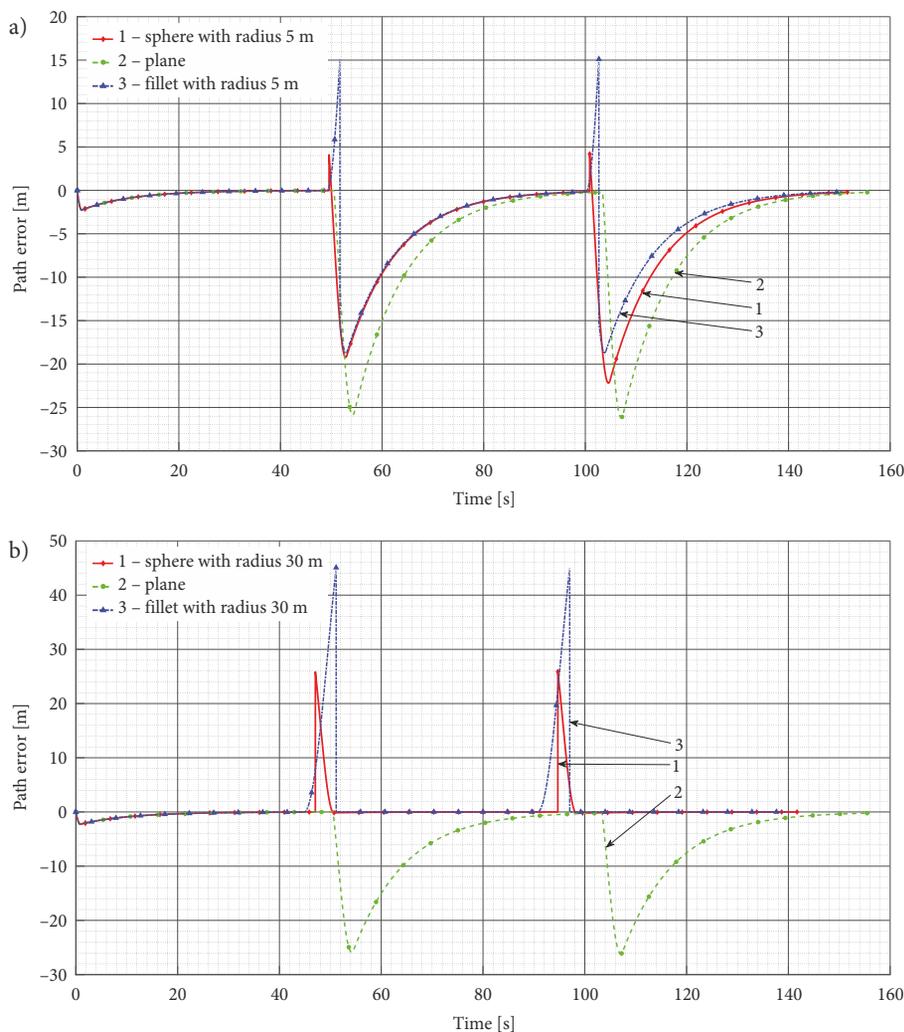


Figure 9. Flight path error values with respect to time with radius  $r$ , when different waypoint-switching method is used: a -  $r = 5$  m; b -  $r = 30$  m

Results of the extreme cases are presented in Figure 10.

Results in Figure 10a prove that up to the radius of 20 m Dubins paths waypoint-switching method is much more accurate, but for bigger sphere radius values classical method has a tendency of smaller average flight path errors. In Figure 10a we can see that PDF is slightly bigger if we use Dubins method. This difference is bigger for the flight path error values, which are close to 0 m.

However, from Figure 10b we can see, that PDF of classical method grows very steep when UAV flight trajectory is not overshooting flight path, which is near the waypoints. We can observe this situation also in Figure 8f. It proves that for bigger sphere or Dubins circle radius values, the classical waypoint-switching algorithm is much more accurate.

To do the last check, we made more experiments using flight missions of different shapes. The same radius values and the same 10 m/s flight speed were used. In these cases UAV needed to make more, but less sharp turns in the range from 45 (octagonal shape mission) to 90° (quadrangle shape mission). Flight mission length was always the

same – 1500 m. All the results are shown in Figure 11 for classical algorithm and Figure 12 for Dubins waypoint-switching algorithm.

Figure 11 shows that when the classical waypoint-switching algorithm is used, (but different mission shape should be flown), optimal sphere radius would also be different. For equilateral triangle, we have 28 m but for octagon shape mission it is only 7 m. For other flight missions optimal results are in between 7 and 28 m as it is shown in Figure 11.

For Dubins flight paths waypoint-switching algorithm tendency is slightly different. If we would choose smaller fillet radius, flight would be more accurate for the mission where smaller course change angles are needed – like octagon shape mission. For bigger fillet radiuses it is also, the same – more accurate flight would be if we would fly mission where smaller course change angles are needed. However, optimal fillet radius would be almost the same for all types of flight missions. As we can see from Figure 12 this range is from 16.5 up to 18 m. This proves that Dubins paths would be better choice if we care about flight path accuracy.

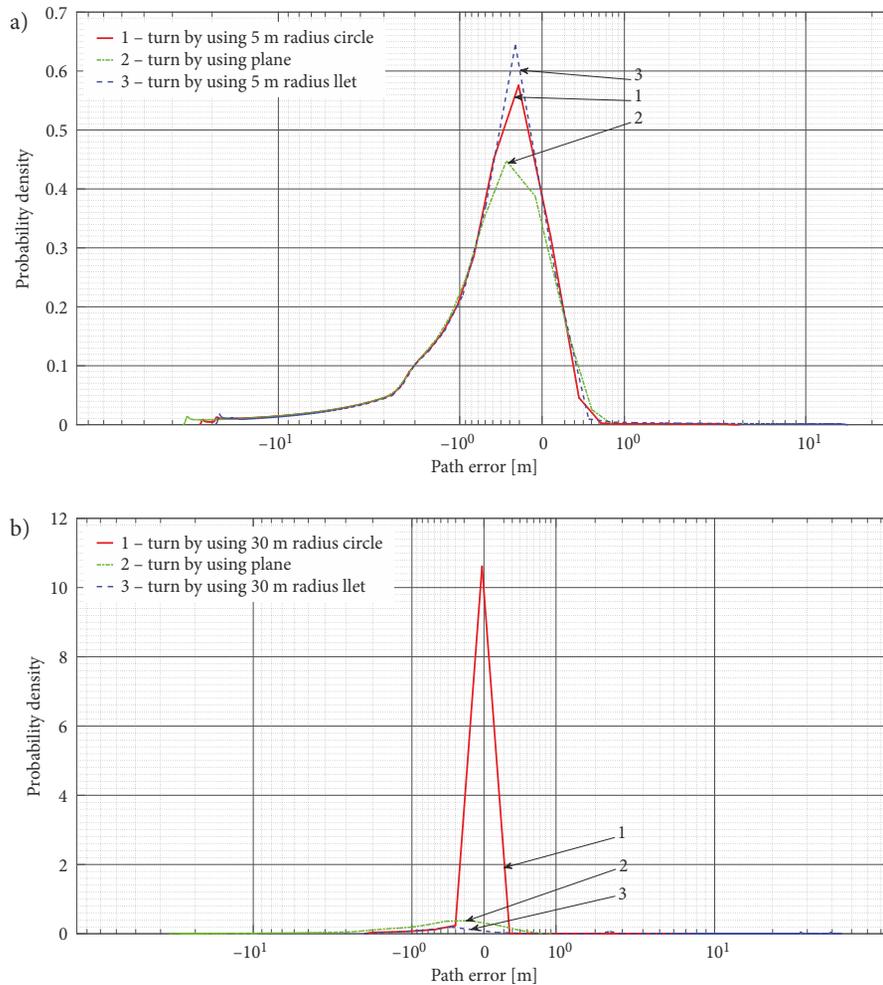


Figure 10. PDF for different waypoint-switching methods, when radius  $r$  is used: a –  $r = 5$  m; b –  $r = 30$  m

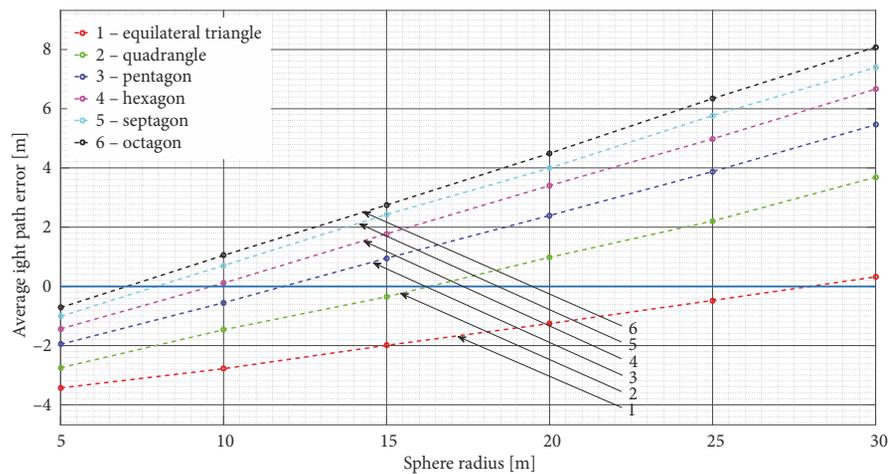


Figure 11. Dependency of average flight path error on sphere radius for different flight trajectories when classical waypoint-switching algorithm is used

On the other hand, if we would compare results from Figure 11 with the results in Figure 12, they would prove that there is no optimal flight mission setup, which would ensure smallest flight path errors. For every flight mission we can find optimal sphere or fillet radius. The difference between some of these optimal values can be large enough, as for example: for octagon mission with classi-

cal algorithm the sphere radius is about 7 m, but for the same mission when Dubins flight path algorithm is used it would be about 18 m. However, if we would use Dubins flight path algorithm instead of the classical, we could at least approximately choose the optimal fillet radius (which fits all flight missions) and use it for further research.

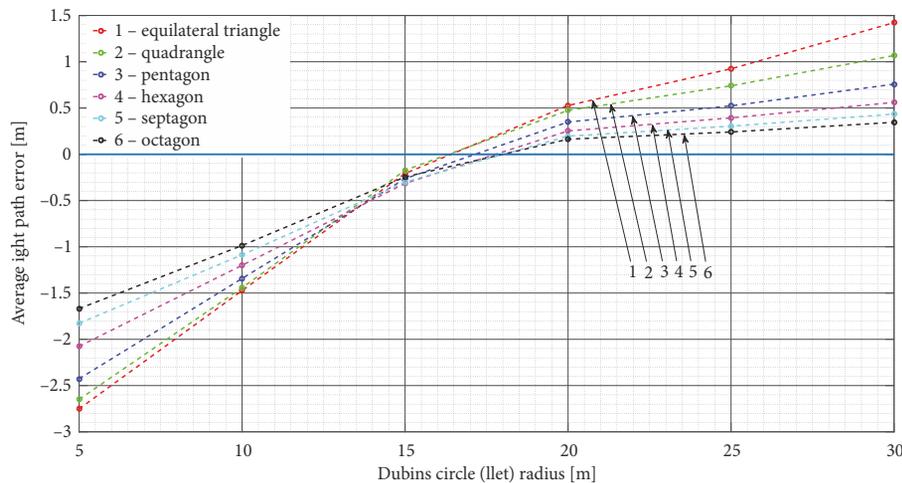


Figure 12. Dependency of average flight path error on Dubins circle (fillet) radius for different flight trajectories, when Dubins paths waypoint-switching algorithm is used

## Conclusions

From all of the experimental results we can make conclusions that the mean flight path error could be smallest either for the classical waypoint-switching method or for the filleted waypoint-switching method. This depends generally on UAV configuration (mainly flight speed) and the sphere or Dubins paths fillet radius. When UAV flies the mission where waypoints make equilateral triangle with side length of 500 m, the maximum and minimum errors are changing from 22 up to 26 m for classical algorithm and from 20 up to 45 m for the fillet. On the other hand, average values in most cases are better for Dubins paths method with fillets. Addition to that, if we increase sphere or fillet radius, probability density (to have mean flight path error 0) initially is better for Dubins algorithm, but for bigger radiuses tendencies change to the classical algorithm. It is due to the fact that, the Dubins paths try not to overshoot the trajectory. For classical algorithm and large sphere radiuses probability density becomes more than 10. It is due to the fact, that in this case classical algorithm is much more flight path error efficient than others and tends to look like Dubins trajectory.

In addition, we could say that UAV would make smallest drift near waypoints if the combination of waypoint-switching methods would be used. UAV needs automatically choose appropriate waypoint-switching method according to the radius it needs to fly during the mission. The second option to increase flight path accuracy could be if we would provide an option for UAV to choose waypoint-switching algorithm automatically for every waypoint (especially if these waypoints have differently defined sphere or circle radiuses). In that case, UAV could use database values presented in Figures 11 and 12.

From Figures 11 and 12 we can also see that Dubins paths algorithm is more predictable. UAV flight path error could be reduced just by choosing approximately optimal fillet radius, which is almost the same for all types of flight missions. For the flight speed of 10 m/s, this fillet radius is in the range from 16.5 up to 18 m.

For further research current database of flight path accuracy values could be extended by evaluating wind factor or used for generating more advanced automated flight UAV algorithms (more advanced ground obstacle avoidance, time and distance optimal paths).

## References

- Ariff, O. K.; Go, T. H. 2011. Waypoint navigation of small-scale UAV incorporating dynamic soaring, *The Journal of Navigation* 64(1): 29–44. <https://doi.org/10.1017/s0373463310000378>
- Avellar, G. S. C.; Pereira, G. A. S.; Pimenta, L. C. A.; Iscolod, P. 2015. Multi-UAV routing for area coverage and remote sensing with minimum time, *Sensors* 15(11): 27783–27803. <https://doi.org/10.3390/s151127783>
- Beard, R. W.; Humpherys, J. 2011. Following straight line and orbital paths with input constraints, *Proceedings of the 2011 American Control Conference*, 29 June–1 July 2011, San Francisco, CA, US, 1587–1592. <https://doi.org/10.1109/acc.2011.5990840>
- Benghezal, A.; Louali, R.; Bazoula, A.; Chettibi, T. 2015. Trajectory generation for a fixed-wing UAV by the potential field method, in *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, 25–27 May 2015, Tlemcen, Algeria, 1–6. <https://doi.org/10.1109/CEIT.2015.7233049>
- Brezoescu, A.; Castillo, P.; Lozano, R. 2011. Straight-line path following in windy conditions, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38(1/C22): 283–288. <https://doi.org/10.5194/isprsarchives-xxxviii-1-c22-283-2011>
- Capello, E.; Guglieri, G.; Quagliotti, F. B. 2009. UAVs and simulation: an experience on MAVs, *Aircraft Engineering and Aerospace Technology* 81(1): 38–50. <https://doi.org/10.1108/00022660910926890>
- Capello, E.; Guglieri, G.; Ristorto, G. 2017. Guidance and control algorithms for mini UAV autopilots, *Aircraft Engineering and Aerospace Technology* 89(1): 133–144. <https://doi.org/10.1108/aeat-10-2014-0161>
- Casbeer, D. W.; Kingston, D. B.; Beard, R. W.; McLain, T. W. 2006. Cooperative forest fire surveillance using a team of small unmanned air vehicles, *International Journal of Systems Science* 37(6): 351–360. <https://doi.org/10.1080/00207720500438480>

- Chamseddine, A.; Zhang, Y.; Rabbath, C. A.; Theilliol, D. 2012. Trajectory planning and replanning strategies applied to a quadrotor unmanned aerial vehicle, *Journal of Guidance, Control, and Dynamics* 35(5): 1667–1671. <https://doi.org/10.2514/1.56606>
- Dadkhah, N.; Mettler, B. 2012. Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance, *Journal of Intelligent & Robotic Systems* 65(1–4): 233–246. <https://doi.org/10.1007/s10846-011-9642-9>
- Kothari, M.; Postlethwaite, I.; Gu, D.-W. 2014. UAV path following in windy urban environments, *Journal of Intelligent & Robotic Systems* 74(3–4): 1013–1028. <https://doi.org/10.1007/s10846-013-9873-z>
- Li, W.; Chen, W.; Wang, C.; Liu, M.; Ge, Y.; Song, Q. 2015. A 3D path planning approach for quadrotor UAV navigation, in *2015 IEEE International Conference on Information and Automation*, 8–10 August 2015, Lijiang, China, 2481–2486. <https://doi.org/10.1109/icinfa.2015.7279703>
- Maillot, T.; Boscaïn, U.; Gauthier, J.-P.; Serres, U. 2015. Lyapunov and minimum-time path planning for drones, *Journal of Dynamical and Control Systems* 21(1): 47–80. <https://doi.org/10.1007/s10883-014-9222-y>
- Nelson, D. R.; Barber, D. B.; McLain, T. W.; Beard, R. W. 2006. Vector field path following for small unmanned air vehicles, in *2006 American Control Conference*, 14–16 June 2016, Minneapolis, MN, US, 5788–5794. <https://doi.org/10.1109/acc.2006.1657648>
- Owen, M.; Beard, R. W.; McLain, T. W. 2015. Implementing Dubins airplane paths on fixed-wing UAVs, in K. Valavanis, G. Vachtsevanos (Eds.). *Handbook of Unmanned Aerial Vehicles*, 1677–1701. [https://doi.org/10.1007/978-90-481-9707-1\\_120](https://doi.org/10.1007/978-90-481-9707-1_120)
- Stojcsics, D. 2014. Autonomous waypoint-based guidance methods for small size unmanned aerial vehicles, *Acta Polytechnica Hungarica* 11(10): 215–233.
- Wang, T.; Le Maitre, O. P.; Hoteit, I.; Knio, O. M. 2016. Path planning in uncertain flow fields using ensemble method, *Ocean Dynamics* 66(10): 1231–1251. <https://doi.org/10.1007/s10236-016-0979-2>
- Yeol, J. W.; Hwang, Y.-W. 2016. Parametrization of nonlinear trajectory for time optimal 2D path planning for unmanned aerial vehicles, in *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, 28–30 April 2016, Hong Kong, China, 335–339. <https://doi.org/10.1109/iccar.2016.7486751>
- Zhong, W.; Yan, L. 2014. A target visiting path planning algorithm for the fixed-wing UAV in obstacle environment, in *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 8–10 August 2014, Yantai, China, 2774–2778. <https://doi.org/10.1109/CGNCC.2014.7007603>