TRAJECTORY OPTIMISATION FOR AUTONOMOUS VEHICLES: A MINIUAV APPLICATION

Elisa Capello, Giorgio Guglieri, Fulvia Quagliotti, Gianluca Speciale

Politecnico di Torino – DIASP
Corso Duca degli Abruzzi 24
10129 Torino (Italy)

Keywords: *mini-UAV*, *trajectory optimization, genetic algorithm, mission analysis*

Abstract

This paper deals with the optimisation of the trajectories for UAVs (Unmanned Aerial Vehicles). The aircraft considered for the present study is a miniUAV (Micro Aerial Vehicles, MAVS) developed at Politecnico di Torino: the MicroHawk600. The research activity consists of: analysis of operating scenarios, definition of the mathematical model, analysis of optimisation criteria and simulation of different test cases. The aerial vehicle is expected to reach the desired locations in the operational environment expressed in terms of planned waypoints. Assuming the vehicle as a point mass model, the best solution has been investigated through a genetic algorithm search procedure. The optimization problem has been solved by modifying a micro-genetic algorithm software.

Several waypoint distributions were successfully tested. Results obtained with this optimization procedure are presented.

1. Introduction

For aircraft handling qualities requirements the performance of controlled systems may be defined in terms of regions of interest instead of desired states. Stability, robustness, good command following, noise rejection and low sensitivity to model uncertainties are also significant concerns for the designer and their inclusion in the design process may be required to ensure adequate control performances.

Within this area of investigation, the search of optimal solutions, that should meet the above mentioned prerequisites, becomes a primary interest in the design of controllers. [1]

In this paper, the design of optimal flight trajectories for an autonomous UAV is discussed. The Aerospace Engineering Department (DIASP) of Politecnico di Torino is working on the development of the MicroHawk family of MAVs since some years. The MicroHawk concept [2] is designed within an European Union funded project (MARVEL, Micro Aerial Vehicles for Multi Purpose Remote Monitoring and Sensing Project) at Politecnico di Torino. The MicroHawk configuration is characterized by a conventional layout: it is a fixed wing, tailless integrated wing-body configuration and tractor propeller driven.

Different versions of the MicroHawk configuration exist, covering a range of dimensions and operating performances. The availability of different MicroHawk versions allows covering an extended range of flight speed (estimated between 7 m/s and 20 m/s) and flight endurance (ranging between 15 and 60 min, according to size and energy source storage).

In this paper the attention is focused on the MicroHawk 600. Its design is mainly addressed to the need for payload weight fraction of about 150 gr and larger internal volumes. The MH600 version can achieve autonomous flight, as it is possible to locate on-board a commercial small size autopilot (Micropilot MP2028) without exceeding wing loading limitations for hand launch. The navigation system, based on inertial and GPS data, is programmable according to tasks and waypoints (Fig. 1).

The capability and the UAVs roles require new concepts for their control. A significant aspect of this control problem is optimizing the trajectory from the starting point to its final destination. Online trajectory generation for flight control application is important in engineering applications with unmanned aerial vehicles to provide feasible flight control. [6,7]

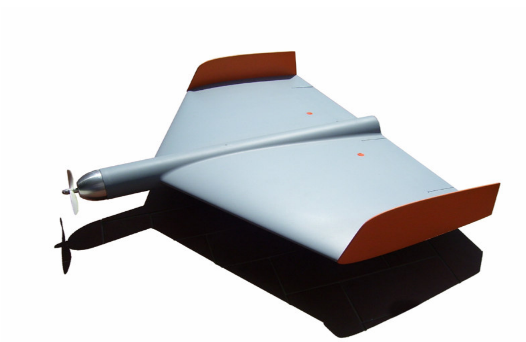

Fig. 1 The MicroHawk platform

The main types of search methods are calculus-based, enumerative and random [3]. Calculus-based methods search the optimal conditions in a neighborhood of the current point. A severe limitation of these methods, in terms of robustness, is the reliance on the existence of derivatives and on the continuity of the function to be maximized. Enumerative methods seek the optimal solution by computing the objective function in every point of the search space. So, this algorithm is substantially inefficient and time consuming when applied to large domains of solutions. Random search is an alternative strategy which can bypass the limitations of the previous methods. The genetic algorithm belong to this last solvers family, as the random choice of the possible solution is combined with criteria for the direction of search derived from natural evolution of species [4,5]. This technique is considered global and robust in terms of search over a space of solutions.

2. The genetic and the micro–genetic algorithm

The genetic solver adopted for the design of the control system is a Fortran version of the driver described by D. Carroll [8,9]. The code initializes a random sample of individuals with different parameters to be optimized using the genetic algorithm (GA) approach. The genetic algorithm operates on the principle of the survival of the fittest. A constant-size population of individuals, represented by a fixed number of parameters which are coded in binary form (chromosomes), encode possible solutions of a given problem. An initial population of individuals (possible solutions) is generated at random. In every generation (evolutionary step), the individuals of the current population are decoded and evaluated. Each possible solution is analyzed by a fitness function which decides whether it will contribute to the next generation of solutions. Once the new population has been selected, chromosomes are ready for crossover and mutation.

The crossover operator combines the features of two parents to create new solutions. Crossover allows an improvement in the species in terms of random evolution of new solutions on each parent and then, complementary fractions from the two parents are linked together to form a new chromosome.

The mutation operator alters a copy of a chromosome reintroducing values that might have been lost or creating totally new features. Each iteration produces a new population of solutions (generation). The genetic algorithm continues to apply the operators and evolve generations of solutions until a near-optimum solution is found or the maximum number of possible generations is produced.

The solution using a micro genetic algorithm is also possible. The micro GA utilization reduces the number of function evaluations and demonstrates faster convergence average to near-optimal region [10,11]. A micro GA starts with a random and very small population. This population evolves and converges in a few generations. Then, a new random population is chosen while keeping the best individual from the previously converged generation and the evolution process restarts. Average population fitness values are not meaningful with a micro GA because of the start-restart nature of the evolution process. After many numerical experiments [8,9] in order to tune the search algorithm adopted, the code is set for maximum population size of five individuals, 48 bits per individuals and three parameters (i.e. 16 binary bits per parameter and 2^{16} possible solutions per parameter). Niching and creep mutation are enabled and one child per pair of parents is considered.

3. The mathematical model

The mathematical model used to describe the vehicle dynamics is a three-dimensional point-mass model written in wind axes frame [7].
The equation are:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{\gamma} \\ \dot{\chi} \\ \dot{V} \end{pmatrix} = \begin{pmatrix} V\cos\gamma\cos\chi \\ V\cos\gamma\sin\chi \\ V\sin\gamma \\ \frac{g}{V}(n\cos\varphi - \cos\gamma) \\ \frac{g}{V}\frac{n\sin\varphi}{\cos\gamma} \\ \frac{T_e - D}{m} - g\sin\gamma \end{pmatrix} \quad (1)$$

In (1) x, y and h are the coordinates of the Centre of Gravity (CG) of the aircraft in NED (North East Down) reference frame. Angles are defined as: φ is the bank angle, χ is the heading angle and γ is the flight-path angle. T_e is the engine thrust, D is the aerodynamic drag, m the aircraft mass, *g* the acceleration of gravity. The ground-speed velocity is assumed to be equal to the airspeed (so wind is neglected). The bank angle φ, the engine thrust T_e and the load factor $n = \frac{L}{mg}$ are the control variables for the aircraft. Hence, the input vector u is:

$$u = [\varphi, T_e, n] \quad (2)$$

The system, complemented with constraints on applicable inputs, form the basis for aircraft trajectory optimization. Constraints are usually written in terms of state variables and controls.
Some constraints are set on aircraft state and control variables like $[\varphi, T_e, n]$. During the navigation, limitations are applied on the flight-path angle in both climbing and descending trajectories and on upper and lower bounds for airspeed V and climb angle γ. A preliminary study of the aircraft response to inputs has been conducted to investigate the platform general behaviour. The platform (MH600) reacts to changes in terms of n and φ with sensible trajectory variations. This means that - in the trajectory design – little variations of input variables must be taken into account for accurate tracking (number of bits/chromosome). Initial and final conditions for the aircraft state variables are also specified as a prerequisites for the design process.

4. Optimal trajectories

The purpose of the analysis is the definition of optimal trajectories; two optimisation methods can be analyzed. First, the optimized trajectory is obtained using the genetic algorithm and it is defined in terms of energy spent by the aircraft during the navigation and in terms of minimum distance from the waypoint. In the second method, the operator can use the Dubin's curves and the objective is to generate paths with continuous curvature which respect the constraints. Dubin's curves are used to generate the shortest path (time optimal) between two waypoints, given initial and final constraints and constraints on motion.

The genetic algorithm method is typically used for a mini-UAV tracking and short range missions, due to the possibility to fit altitude variations considering the energy spent. Instead, the Dubin's curve approach is applied for mapping and on-field missions, due to the possibility to choose the shortest and time-optimal path.

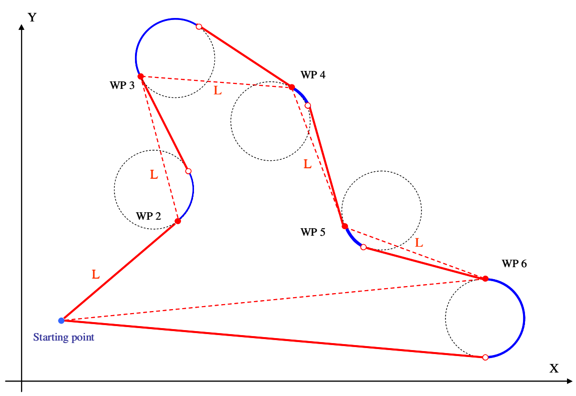

Fig. 2 The behaviour of Dubin's curves

4.1. Trajectories with genetic algorithm

Starting from a list of waypoints through which the vehicle should go across, we will determine which amongst the candidate trajectories will be the optimal in terms of energy. The trajectories that cross through a prefixed list of waypoints are infinite but only some of these can be carried out by the platform and those with discontinuity of derivates are discarded (segmented trajectories).

Different types of curves are able to connect a map of points: for example it is possible to use approximation and/or interpolation curves such as Bézier curves and spline curves. Bézier curves are less prone to cross exactly the waypoints if they are not aligned and, moreover, the genetic algorithm parametrization would be slightly more difficult. For this reason the curve adopted is a polynomial spline and a cubic function two times differentiable in the whole range. The relative second-derivate is equal to zero in the final points. This is useful for the imposition of different auxiliary conditions in the trajectory such as initial attitudes and directions.

The selection of the trajectories is also defined by the design constraints. The first restriction is that the trajectory must pass through all the scheduled waypoints. Since the autopilot programs are not so restrictive in terms of waypoint reaching, it is widely accepted that the waypoint is reached when the distance of the aircraft is less than a given tolerance (e.g. 10 m). In any case, one of the objectives is to minimize the value of this distance. Another constraint is the initial value of the bank angle.

Higher order splines are able to connect all the prefixed waypoints with good precision but would require some additional computing time. In order to avoid this disadvantage, the list of scheduled waypoints has been divided in groups of three and the trajectory has been sequenced: after the optimization of the first group of three waypoints, the process restarts considering, as a first waypoint, the second one in the previous group of three, and so on. The continuity is guaranteed during the following optimizations, as the value of the bank angle is kept across junctions.

The optimal trajectory is a balance between:

- energy cost

- precision on waypoint reaching
- feasibility of the trajectory (platform dependent limitations).

The fitness function (or cost function) is a balance between the energy required to perform the trajectory and the precision of waypoint tracking. The energy cost is given in terms of flight commands. In other words, a trajectory that requires large command changes implies a huge energetic cost. The fitness function is given in the following form:

$$J = w_1 \int_{t_i}^{t_f} \Delta\varphi(t)dt + w_2 \int_{t_i}^{t_f} \Delta n(t)dt + w_3 \int_{t_i}^{t_f} \Delta T_e(t)dt + \\ + w_4 r_1 + w_5 r_2 \qquad (3)$$

where w_1, w_2, w_3, w_4, w_5, are weight factors and r_1, r_2 are the distances from the second and the third waypoint respectively.

4.2. Path planning with Dubin's curves

The possibility to determine the best path is presented in Ref. [12]. Straight lines are formed as tangents to circles and can have different lengths; the arcs are the segments of the circle which connect two tangents together. The radius of the circle is the minimum path curvature which the vehicle can take.

In order to obtain autonomous path, it is necessary to create a simple code to study the possibility to generate a path to connect the waypoints using different parameters, such as the speed, the Probability Density Function (PDF) around the plane and the distance between waypoints. This distance is fixed because for every single step (Δt) the code calculates the position of the next waypoint.

The Figure 3 show how the Dubin's curves work to create the path and plot a path for the return to the starting point. As shown in Fig. 3 the path can have an evolution during the time and it is possible to watch the direction of the aircraft in every single waypoint.

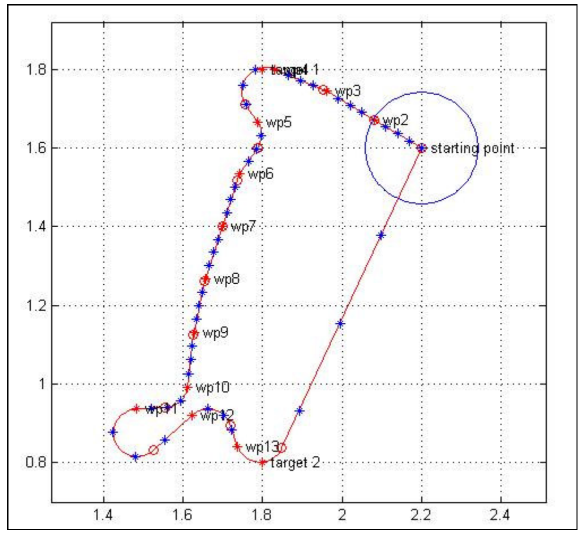

Fig. 3 The trajectory with Dubin's curves

For random scenario considering the mission for surface scanner it doesn't matter the location of the targets because the probability area is only a research area. The target location is unknown for the aircraft, for this reason at every single time Δt the aircraft needs to verify if the target is there or not. At first, it is necessary to create the targets inside the PDF, so the code automatically gives some random coordinates to these points. Then, the algorithm produces an answer for every single points, which can be 1, if the target is found, or 0 if it is not found and decide if it is necessary to stop the creation of the path and go home, or to continue with the search.

5. The trajectory geometry

The outputs of the mathematical model and the geometry of the trajectory are combined in order to evaluate the cost of a single trajectory generated by the genetic algorithm. In this way, after the definition of a population of trajectories, it is possible to calculate the inputs in every single phase of the flight, that can be successfully performed. Integrating the input required along the trajectory, it is possible to obtain the energetic cost. If the inputs required are over the operative capability of the aircraft , the cost function for the trajectory will be evaluated with an energetic cost proportional to its impossibility, forcing the genetic algorithm to move away from this trajectory (deficiency function). The process is described in the following flow chart:

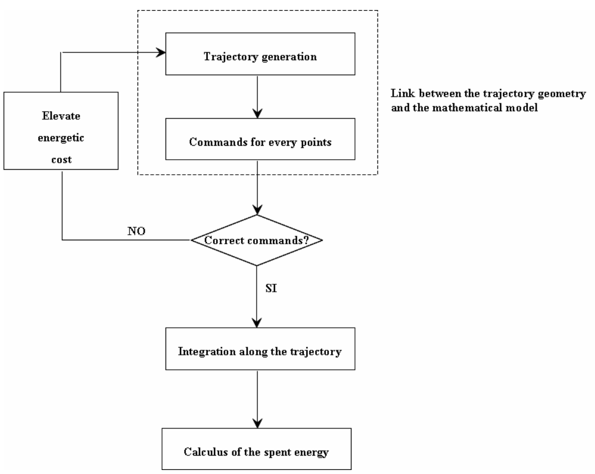

Fig. 4 The trajectory geometry and the mathematical model

When the trajectory has been chosen, all the required geometric parameters are fixed, so that all the spatial derivatives are given.

6. Analysis of the results

Paying the attention on genetic algorithm, some simulations performed for different types of waypoint distribution are shown in Figures 5-16. The number of generations is fixed at 200. For each track two different weight distributions are considered. The first line (solid blue line) refers to energy saving while the second one (dashed black line) enhances the tracking precision.

6.1. Random trajectory

This waypoint distribution is expected to perform a sequence of coordinates with altitude change. In Fig. 5 the 3D view trajectory is represented and in Fig. 6 and Fig. 7 the trajectories in the horizontal and vertical plane are represented.

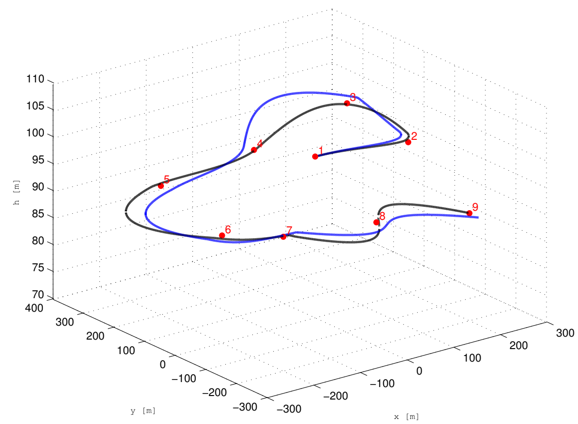

Fig. 5 3D view trajectory

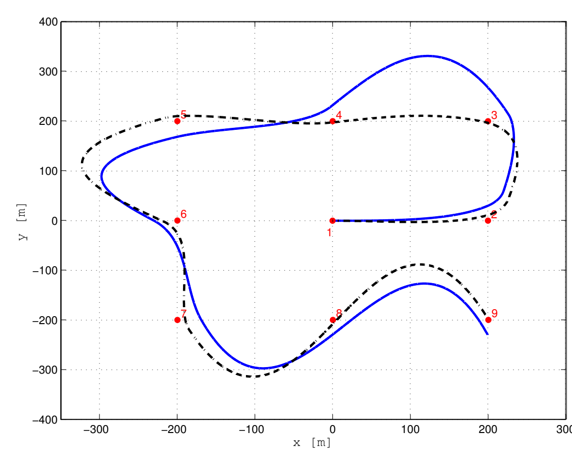

Fig. 6 View in horizontal plane

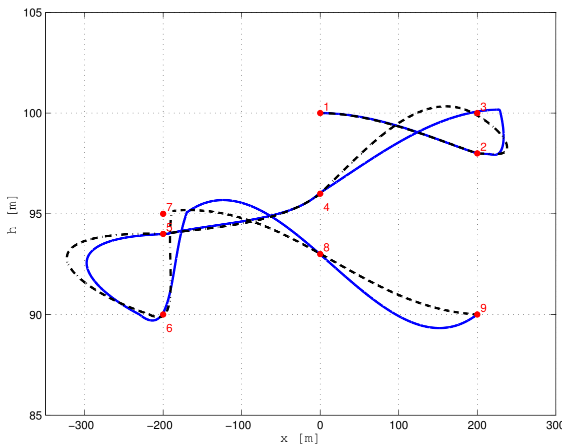

Fig. 7 View in vertical plane

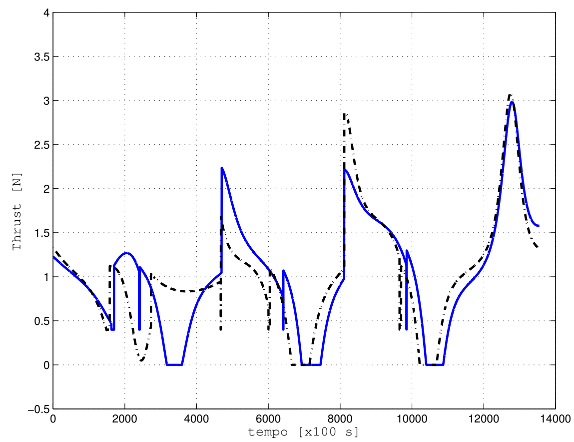

Fig. 8 Thrust trend

By comparing the two different optimized trajectories, the energy saving optimized trajectory is exact enough and adequate compromise between precision and energy saving. The thrust (Fig. 8) has a slightly different time history in the two cases as the impact of this variable in the cost function is less effective as a consequence of the weights selection. Moreover, it is possible to highlight how the φ and n time histories are similar in both cases, as their weights in the cost functions are the same order (Fig. 9 and Fig. 10).

In the following table the E.S.C. and T.P.C. refer respectively to Energy Saving Case and to Tracking Precision Case.

Table 1 Input file parameters value

	E.S.C.	**T.P.C.**
V	10 m/s	10 m/s
χ_{in}	0 deg	0 deg
γ_{in}	0 deg	0 deg
w_1	10^7	10^6
w_2	10^3	10^4
w_3	1	10
w_4	1	10^4
w_5	1	10^4

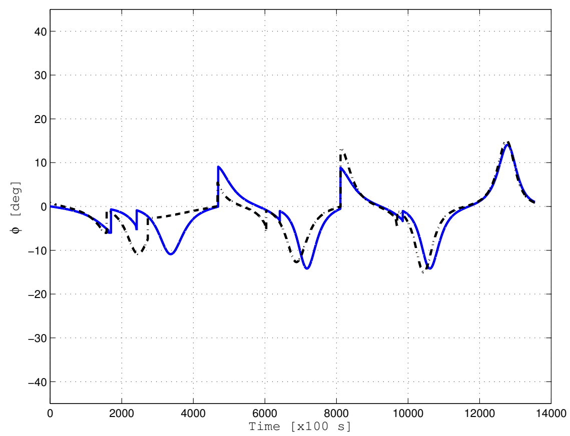

Fig. 9 Roll angle for random trajectory

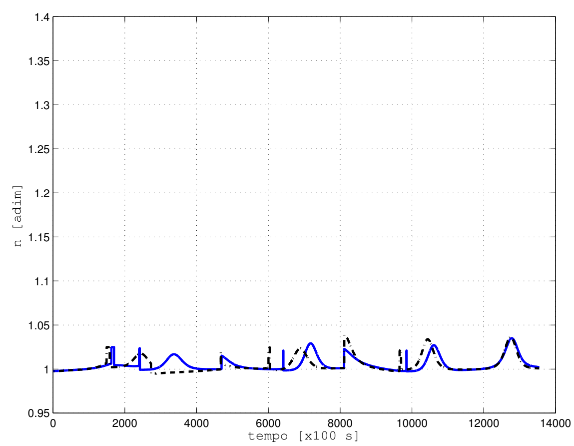

Fig. 10 Load factor trend

6.2. 'Butterfly' trajectory

Butterfly trajectory represents a difficult case of waypoint distribution for UAV trajectories: in fact heading from crosstrack corrections are continuously updated. The task must also be sequentially repeated.

In Fig. 11 the 3D view trajectory is represented and in Fig. 13 and Fig. 14 the trajectories in the horizontal and vertical plane are represented.

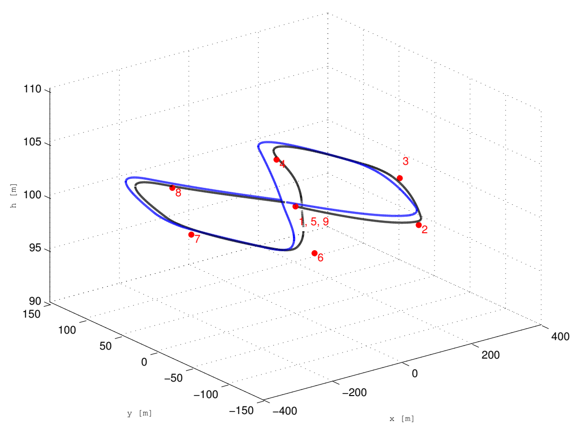

Fig. 11 3D view trajectory

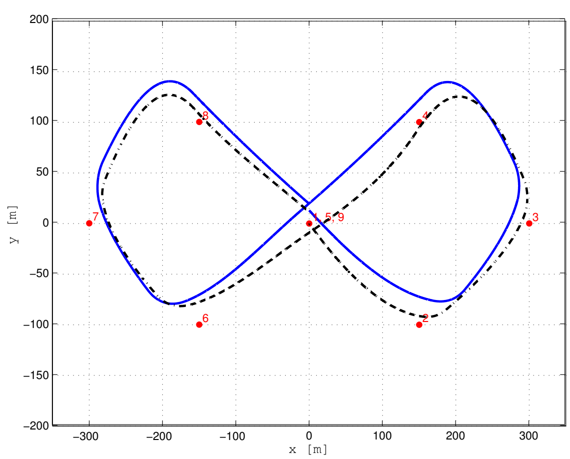

Fig. 12 View in horizontal plane

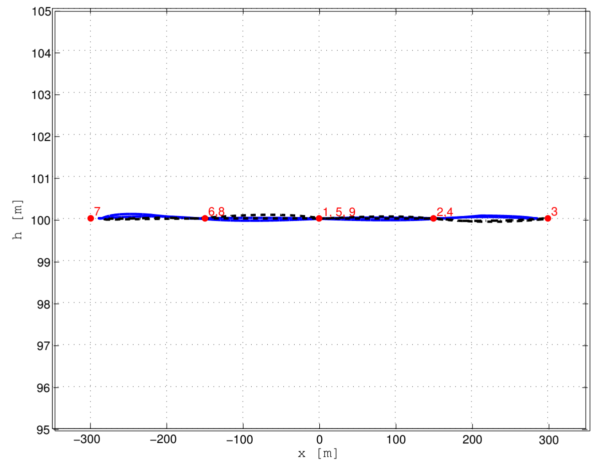

Fig. 13 View in vertical plane

Slight difference between the two separate optimization procedures is highlighted. The total time necessary to complete the whole task is the same.

Table 2 Input file parameters value

	E.S.C.	T.P.C.
V	10 m/s	10 m/s
χ_{in}	-45 deg	-45 deg
γ_{in}	0 deg	0 deg
w_1	10^7	10^6
w_2	10^5	10^4
w_3	1	10
w_4	1	10^4
w_5	1	10^4

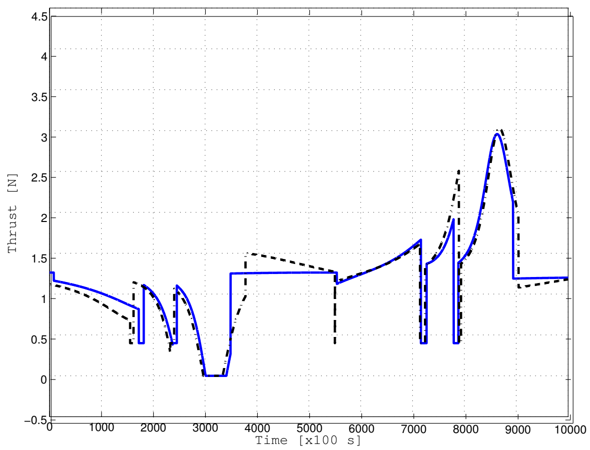

Fig. 14 Thrust trend

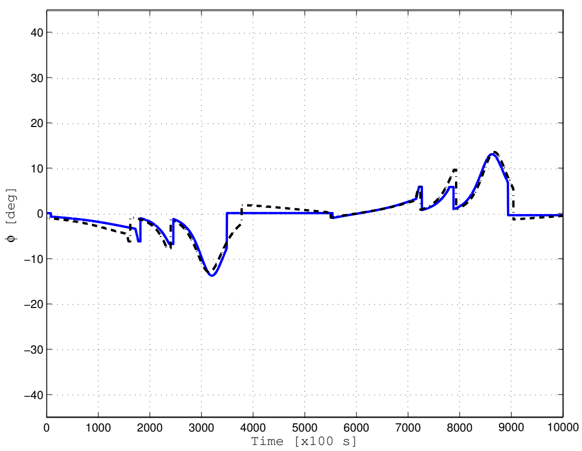

Fig. 15 Roll angle for butterfly trajectory

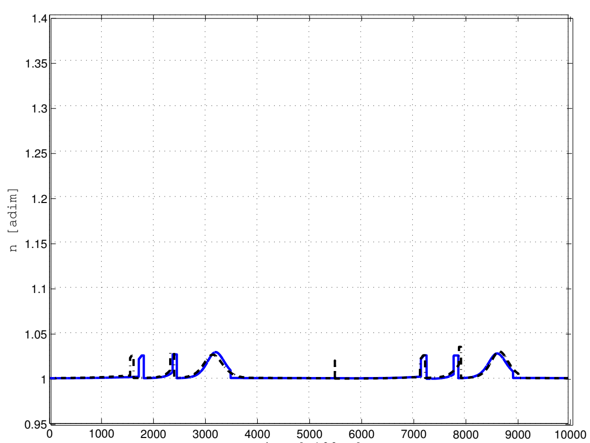

Fig. 16 Load factor trend

6.3. Random trajectory with Dubin's curves

This waypoint distribution is expected to perform a sequence of coordinates without altitude variation. The start waypoint is positioned on runway centre and the other waypoints are random and chosen in the neighborhood of the runway. The resultant trajectory is represented in Fig. 17. The points t_2 and t_3 are the tangent points.

The trajectory represented is the same as in Fig. 6, but the altitude variations are not considered and only the geometric constraints are respected in this case.

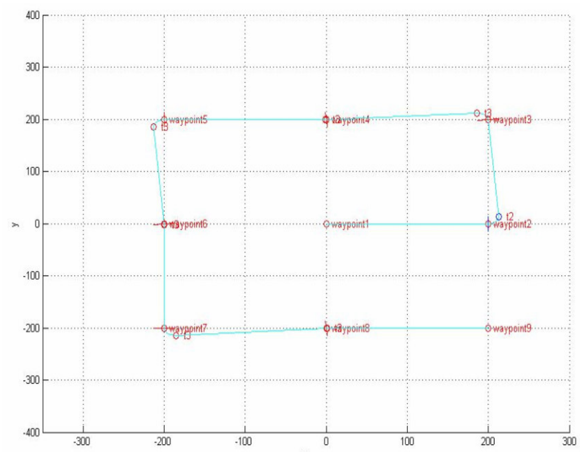

Fig. 17 Random trajectory with Dubin's curves

7. Conclusions

Several waypoint distributions were successfully tested and even random layouts were optimized with limited computational workload.

The genetic algorithm is fast enough to be applied to pre-flight trajectory definition. An extension to real time is potentially possible with an integration between the Dubin's curves and the GA, in order to obtain a time-optimal and a minimum energy spent path. In this way the operator can pick out the waypoints and the search algorithm gives the optimized trajectory. However, with the Dubin approach, actually, the implementation of short range mission with altitude variations is not possible. This is an important limitation, considering the application of this method to trajectory optimization for mini-UAV missions.

If the number of waypoints is too low the optimization shall be weak. Some numerical exercises are required to tune the minimum number of waypoints, in order to obtain a feasible trajectory.

Optimal trajectories are affected by the type of strategy: either energy saving or precision tracking. The user is expected to decide which fits the application, taking into account that, as a conclusion, the best method for a mini-UAV application is the genetic and micro- genetic algorithm.

Acknowledgments

The authors thank Prof. Hugh Stone (University of Sydney) and Ing. Stefano Mazzarello for sharing experience and knowledge on UAV mission design.

References

[1] Fantinutto R., Guglieri G. and Quagliotti F.B. Flight control system design and optimisation with genetic algorithm. *Aerospace Science and Technology*, vol.9, 2005

[2] Quagliotti F.B. and Guglieri G. Micro Aerial Vehicles and their Operational Scenarios. *RTO-MP-SCI-174, Tactical Decision Making and Situation Awareness for Defense against Terrorism,* Turin, 2006

[3] Goldberg D.E. *Genetic algorithm in search, optimisation and machine learning.* Addison Wesley, Reading, MA, 1989.

[4] Gray G.J, Li Y., Murray- Smith D.J and Sharman K.C. Specification of a control system fitness function using constraints for genetic algorithm based design methods. *1st IEE/IEEE Internation conference on Genetic Algorithms in Engineering Systems*, Sheffield, UK, 1995.

[5] Gray G.J, Li Y., Murray- Smith D.J, Romeo E. and Sharman K.C. The application of genetic algorithms to gain-scheduling controller analysis and design. *2nd IEE/IEEE International conference on Genetic Algorithms in Engineering Systems*, Glasgow, UK, 1997.

[6] Gallo G., Guglieri G., Quagliotti F.B. and Speciale G. Optimal Mission Planning For An Autonomous Unmanned Aerial Vehicle. BIOMA 2006, *Bioinspired Optimization Methods and Their Application*, Ljubjana, Slovenia, 2006.

[7] Guglieri G., Quagliotti F.B. and Speciale G. Optimal Trajectory Tracking For An Autonomous UAV. *Journal of Automatic Control in Aerospace* , As-08-001, 2008.

[8] Carroll D.L. Genetic algorithms and optimizing chemical oxygen-iodine lasers. *Developments in Theoretical and Applied Mechanics*, vol. 18, School of Engineering, The University of Alabama, 1996.

[9] Carroll D.L. Chemical modelling with genetic algorithms. *AIAA Journal,* vol. 34, no.2, 1996.

[10] Carroll D.L. Genetic algorithms and Optimizing Chemical Oxygen-Iodine Lasers. *Development in Theoretical and Applied Mechanics*, vol. XVIII, eds H. Wilson.

[11] Carroll D.L. Chemical laser modelling with genetic algorithms. *AIAA Journal*, vol.34, no.2, 1996.

[12] Mazzarello S., High level path planning and mission management for Small Unmanned Aerial Vehicles. *Master's Thesis*, Politecnico di Torino, October 2007, prepared at University of Sydney, Australia (Supervisors: Prof. Hugh Stone, Prof. F.B. Quagliotti).