# OPTIMAL TRAJECTORY TRACKING FOR AN AUTONOMOUS UAV

G. Guglieri [*] F.B. Quagliotti [†] G. Speciale [‡]
Politecnico di Torino
Dipartimento di Ingegneria Aeronautica e Spaziale
Torino - Italy

**Abstract**

The aim of the present project is the design of optimal flight trajectories for an automomous aerial vehicle which is expected to reach the desired locations in the operational environment expressed in terms of planned waypoints. The navigation must be performed with the vehicle's best effort, i.e. with the lowest cost. Hence, we want to minimize the input energy, a function of the inputs for the mathematical model which describes the dynamics of the vehicle. The trajectory must satisfy all the constraints and pass through all the planned waypoints. Assuming the vehicle as a point mass model, the best solution has been investigated through a genetic algorithm search procedure. The optimisation problem has been solved by modifying a micro-genetic algorithm software which was initially developed by D.L. Carroll. Between all the possible trajectories we select the more "realistic" connections among the waypoints. First of all, we have left out the trajectories with discontinuity in the derivatives as these are not feasible by the real aircraft. The polynomial spline is a suitable candidate to solve our problem. The algorithm splits the trajectory in sub-trajectories which join a sequence of three waypoints. Starting from the first three waypoints, the following sub-trajectories are superimposed keeping the first waypoint coincident with the last of the previous sub-trajectory. The sequence of polynomials is initialized assuming that jumps in the direction of flight are avoided pointing the heading angle in the presumed direction of flight. The optimal trajectory is a trade-off amongst three factors: the required energy cost, the minimum distance from the required waypoint and the feasibility of the trajectory. Results obtained with this optimization procedure are presented.

## Keywords

Genetic Algorithms, Optimal Trajectory, UAV, $\mu$-GA

## 1 Introduction

In the present paper, a mission design application for the MicroHawk micro aerial vehicle is discussed. The MicroHawk [1] [2] [3] concept was designed within a European Union funded project (Micro Aerial Vehicles for Multi Purpose Remote Monitoring and Sensing Project), by a research group at Politecnico di Torino. It is a micro/mini aerial vehicle for multi purpose monitoring and sensing. It is a fixed wing, tailless integrated wing-body configuration controlled by two trailing edge movable elevons, powered by a DC motor and tractor propeller Fig.1. The configuration has been fitted out with a circular fusulage that locates all subsystems on a modular slide. The standard configuration is equipped with a micro camera to support a basic reconnaissance mission.

Different versions have been developed and tested, characterized by different size and weight. The MH600 is characterized by a 600 mm wingspan and a bare platform weight of 400 g. Its design has been mainly adjusted for higher payload weight fraction and larger internal volumes. The MH600 version can achieve autonomous flight and it locates onboard a commercial small size autopilot without exceeding wing loading limitations for hand launch.

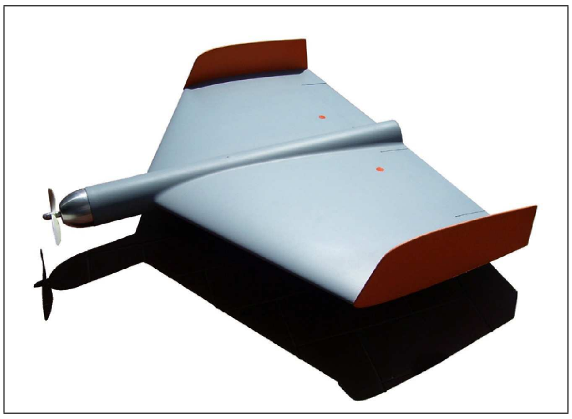
Figure 1: The MicroHawk configuration.

The capability and the roles of UAVs are evolving, and 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. Moreover, online trajectory generation for flight control application is important

[*]Associate Professor, Member AIAA/AHS/AIDAA
[†]Associate Professor, Member AIAA/AIDAA
[‡]Ph. D. Student, Member AIDAA

in engineering applications with unmanned aerial vehicles to provide feasible flight control in complex flight situations.

In general, the solution of the optimal control problem with high dimensional space is hard to compute. This problem is complicated by the fact that the space of possible control action is large. Two well-known methods that have been applied to this problem are Probabilistic Road Maps [4] (PRMs) and Rapidly-exploring Random Trees [5] (RRTs). These methods reduce the dimensionality of the problem by sampling the possible actions, but the resulting trajectories are generally not optimal. Another approach to the optimal trajectory problem consists of applying the Model Predictive Control (MPC). MPC refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of a system [6]. The main idea of MPC is to choose the control action by repeatedly solving, on-line, an optimal control problem. This aims to minimizing a performance criterion over a future horizon, possibly subjected to constraints on the manipulated inputs and outputs, where the future behavior is computed according to a model of the system. An important advantage of MPC is its ability to handle input and state constraints for large scale multivariable plants [7] [8]. Murray [9] has been investigating techniques for generating state and input trajectories which satisfy the equations of motion and trade off tracking performance for inertial stability, using differential flatness. Stochastic search is an alternative strategy that can bypass some limitations of the mentioned methods. The genetic algorithms belong to this last family of solvers, as the random choice of the possible solution is combined with criteria for the direction of search which derive from natural evolution of species. This technique is considered global and robust in terms of search over the space of solutions. The genetic algorithm [10] operates on the principle of the survival of the fittest. A constant-size population of individuals, each of them is 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. The allowable range of variation for each parameter is given. In every evolutionary step, known as a generation, the individuals of the current population (or family) 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 evolution of new solutions at random 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. One or more locations are selected on the chromosome and replaced with new randomly generated values. The three operators are implemented iteratively. 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. Note that, differently from classical search methods, the transition rules from one solution to a new solution in the search space are not given in a deterministic form but using probabilistic operators. Besides, differently from the natural case, the size of the new population is kept constant and each new generation is expected to increase the average fitness. As a final remark, the genetic algorithms have demonstrated a high capability in the search of optimal solutions but they are not well suited for real time application.

## 2 Mathematical Model

### 2.1 Point-Mass Model

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

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{\gamma} \\ \dot{\chi} \\ \dot{V} \end{pmatrix} = \begin{pmatrix} V\cos\gamma\cos\chi \\ V\cos\gamma\sin\chi \\ V\sin\gamma \\ \dfrac{g}{V}\left(n\cos\phi - \cos\gamma\right) \\ \dfrac{g}{V}\dfrac{n\sin\phi}{\cos\gamma} \\ \dfrac{T_e - D}{m} - g\sin\gamma \end{pmatrix} \tag{1}$$

In (1) $x, y, h$ are the coordinates of the Centre of Gravity (CG) of the aircraft and they are usually referred to as North range, cross range and altitude, respectively. Angles are defined as: $\phi$ is the bank angle, $\chi$ is the heading angle and $\gamma$ 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 $V$ is assumed to be equal to the airspeed (wind is neglected). The bank angle $\phi$, the engine thrust $T_e$ and the load factor $n = \dfrac{L}{mg}$ are the control variables for the aircraft. Hence, the input vector $u$ is:

$$u = [\phi, T_e, n] \tag{2}$$

System (1), 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 $n, T_e, \phi$. 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 $\gamma$. A preliminary study of the aircraft response to inputs has been conducted to investigate the platform general behavior. The platform (MH600) reacts to changes in terms of $n$ and $\phi$ 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 prerequisite for the design process.

## 2.2 Micro-Genetic Algorithm $\mu$-GA

The genetic solver adopted for the trajectory optimization is a Fortran version of the driver described by D.L. Carroll [11]. The code initializes a random sample of individuals with different parameters to be optimized using the genetic algorithm (GA) approach. The selection scheme used is a tournament selection with a shuffling technique for choosing random pairs for mating. The routine includes binary coding for the individuals, jump mutation, creep mutation and the option for single-point or uniform crossover. Niching, elitism and an option for the number of children per pair of parents are available. The solution using a micro GA is also possible. This last switch significantly reduced the number of function evaluations and demonstrated faster convergence average to near-optimal region [11] [12]. Note that average population fitness values are not meaningful with a micro-GA because of the start-restart nature of the micro-GA evolution process. Many numerical experiments were performed in Carroll [11] [12] in order to tune the search algorithm adopted and, as a result, the suggest set-up is partially extended for the present application. The code is set for maximum population size of five individuals, 48 bits per individual 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. Table 1 provides summary of the value of the parameters set in the input file.
Irestrt is set to 0 for a new GA run, otherwise equal to 1 for a restart continuation of a GA run. Microga is imposed equal to 1 for micro-GA operation (this will automatically reset some of the other input flags). Moreover Carroll suggests to set Npopsiz equal to 5 when Micro-GA is activated. Nparam is the number of parameters (groups of bits) of each individual. Pmutate is the jump mutation probability, Maxgen is the maximum number of generations evaluated by the GA. Idum is the initial random number seed for the GA run. It must be equal to a negative integer, e.g. Idum=-1000. Pcross is the crossover probability. For single-point crossover, a value of 0.6 or 0.7 is recommended. For uniform crossover, a value of 0.5 is suggested. The GA is presently set up for tournament selection (Itourny equal to 1). Ielite can be defined by two values: 0 for no elitism (best individual not necessarily replicated from one generation to the next) or 1 for elitism to be invoked (best individual replicated into next generation). Icreep must be set to 0 for no creep mutations or 1 for creep mutations: creep mutations are recommended. Pcreep is the creep mutation probability. Iunifrm is equal to 0 for single-point crossover or equal to 1 for uniform crossover: uniform crossover is recommended. In the same way, Iniche equal to 1 means that niching is activated. Nchild can be set equal to 1 for one child per pair of parents or equal to 2 for two children per pair of parents. Usually one child per pair of parent is used.

| | | | |
|---|---|---|---|
| **irestrt** | 0 | **icreep** | 1 |
| **microga** | 1 | **pcreep** | 0.04 |
| **npopsiz** | 5 | **iunifrm** | 1 |
| **nparam** | 2 | **iniche** | 1 |
| **pmutate** | 0.05 | **nchild** | 1 |
| **maxgen** | 200 | **idum** | -1000 |
| **pcross** | 0.05 | **nowrite** | 1 |
| **itourny** | 1 | **ielite** | 1 |

Table 1: Input File Parameters Used in the Simulation.

# 3 Optimal Trajectories

The purpose of the analysis is the definition of optimal trajectories in terms of energy spent by the platform during the navigation and in terms of minimum distance from the waypoint. Starting from a list of waypoints through which we want our vehicle to 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. That's why we prefer polynomial splines. The spline chosen is 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 platform is less than a given tolerance (e.g. 30 m). In any case, one of our objectives is to minimize 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, we proceed restarting the process considering now, 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.

What selects 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:

$$\mathrm{J} = \mathrm{w}_1 \int_{\mathrm{t_i}}^{\mathrm{t_f}} \Delta\phi(\mathrm{t})\,\mathrm{dt} \,+\, \mathrm{w}_2 \int_{\mathrm{t_i}}^{\mathrm{t_f}} \Delta\mathrm{n}(\mathrm{t})\,\mathrm{dt} + \mathrm{w}_3 \int_{\mathrm{t_i}}^{\mathrm{t_f}} \Delta\mathrm{T_e}(\mathrm{t})\,\mathrm{dt} \,+\, \mathrm{w}_4\,\mathrm{r}_1 \,+\, \mathrm{w}_5\,\mathrm{r}_2 \tag{3}$$

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 The mathematical model and the trajectory geometry

In order to evaluate the cost of a single trajectory generated by the genetic algorithm, the outputs of the mathematical model (1) and the geometry of the trajectory are combined. In this way, after the definition of a population of trajectories, it is possible to calculate which will be the inputs in every single phase of the flight so that it 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 that trajectory will be evaluated with an energetic cost proportional to its impossibility, forcing the genetic algorithm to move away from this trajectory (deficiency function).
When the trajectory has been chosen, all the required geometric parameters are fixed, so that all the spatial derivatives are given. The mathematical model is updated in an explicit form:

$$\frac{dx}{dt} = V(t)\cos\gamma(t)\cos\chi(t) \tag{4}$$
$$\frac{dy}{dt} = V(t)\cos\gamma(t)\sin\chi(t) \tag{5}$$
$$\frac{dh}{dt} = V(t)\sin\chi(t) \tag{6}$$
$$\frac{d\gamma}{dt} = \frac{g}{V(t)}\left(n(t)\cos\phi(t) - \cos\gamma(t)\right) \tag{7}$$
$$\frac{d\chi}{dt} = \frac{g}{V(t)}\left(\frac{n(t)\sin\phi(t)}{\cos\gamma(t)}\right) \tag{8}$$
$$\frac{dV}{dt} = \frac{T_e(t) - D(t)}{m} - g\sin\gamma(t) \tag{9}$$

The in-plane and the out-of-plane splines are respectively:

$$y(t) = a_1x(t)^3 + b_1x(t)^2 + c_1x(t) + d_1 \tag{10}$$
$$h(t) = a_2x(t)^3 + b_2x(t)^2 + c_2x(t) + d_2 \tag{11}$$

Combining the equation (4), (5), (6) with the spline equations it is possible to obtain the following equation for the heading angle and the flight-path angle:

$$\chi(t) = \arctan\left(3a_1x(t)^2 + 2b_1x(t) + c_1\right) \tag{12}$$
$$\gamma(t) = \arctan\left(3a_2x(t)^2 + 2b_2x(t) + c_2\right) \tag{13}$$

The horizontal speed $V_G$ is assumed to be constant along the trajectory:

$$V_G = V(t)\cos\gamma(t) \tag{14}$$

The analytical equations for the model inputs become:

$$\phi(t) = \arctan\left[\frac{\dfrac{\frac{d^2y}{dx^2}}{1+\left(\frac{dy}{dx}\right)^2} V_G\cos\chi(t)\cos\gamma(t)}{\dfrac{\frac{d^2h}{dx^2}}{1+\left(\frac{dh}{dx}\right)^2} V_G\cos\chi(t) + \dfrac{g}{V(t)}\cos\gamma(t)}\right] \tag{15}$$

In the same way is possible to write:

$$n(t) = \frac{\frac{d^2y}{dx^2}}{1+\left(\frac{dy}{dx}\right)^2} V_G\cos\chi(t)\frac{V(t)}{g}\left(\frac{\cos\gamma(t)}{\sin\phi(t)}\right) \tag{16}$$

Considering the equations of lift and drag forces, we obtain the expression of the throttle setting as:

$$\mathrm{T_e(t)} = \mathrm{m}\left(\mathrm{V(t)}\tan\gamma(\mathrm{t}) + \mathrm{g}\sin\gamma(\mathrm{t})\right) + \tfrac{1}{2}\rho\mathrm{V}^2(\mathrm{t})\mathrm{S}\left[\mathrm{C_{D0}} + \mathrm{K_{CD}}\left(\frac{\mathrm{n(t)mg}}{\frac{1}{2}\rho\mathrm{V}^2(\mathrm{t})\mathrm{S}}\right)^2\right] \tag{17}$$

In some cases, during the random search process, it may not be possible to calculate the polynomial spline due to unpredictable combination of parameters. To avoid this numerical fault, a coordinate change is recommended (in-plane re-alignment). This option is included in the whole optimization process. If a change of frame is not sufficient to complete the evaluation of the spline, a right turn (error procedure) is imposed to the vehicle.

## 5 Simulation and results

In this section some simulations conducted for different types of waypoint distribution are shown. The number of generations is fixed at 200. For each track two different weight distributions are considered. The first enhances the energy saving (solid blue line) while the second one the precision of the tracking (dashed black line). In tables 2,3 and 4 E.S.C. and T.P.C. refer respectively to Energy Saving Case and Tracking Precision Case.

### 5.1 Random Trajectory

This waypoint distribution is expected to perform a sequence of coordinates with altitude change. In Fig. 2, Fig. 3 and Fig. 4, the trajectory is represented in a 3D view and in the horizontal/vertical plane. In Fig. 5, Fig. 6, Fig. 7 and Fig. 8 the time-histories of glade angle and the control variables are reproduced. In this case, comparing the two different optimized trajectories, it is possible to see that the energy saving optimized trajectory is exact enough with an adequate compromise between precision and energy saving. The thrust has a slightly different time history in the two different cases as the impact of this variable in the cost function is less effective as a consequence of the selection of weights. Moreover, it is possible to highlight how the $\phi$ and $n$ time histories are similar in both cases, as their weights in the cost functions are of the same order.

| | **E. S. C.** | **T. P. C.** |
|---|---|---|
| **V** | 10 m/s | 10 m/s |
| $\chi_{in}$ | 0 deg | 0 deg |
| $\gamma_{in}$ | 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 | $10^4$ |

Table 2: Input File Parameters Value Used in Random Trajectory Simulation.

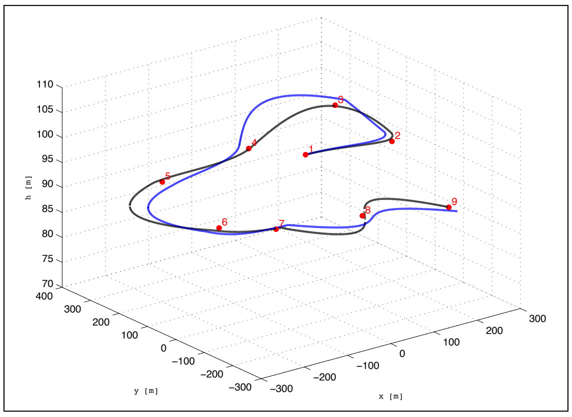

Figure 2: Random Trajectory: 3D View.

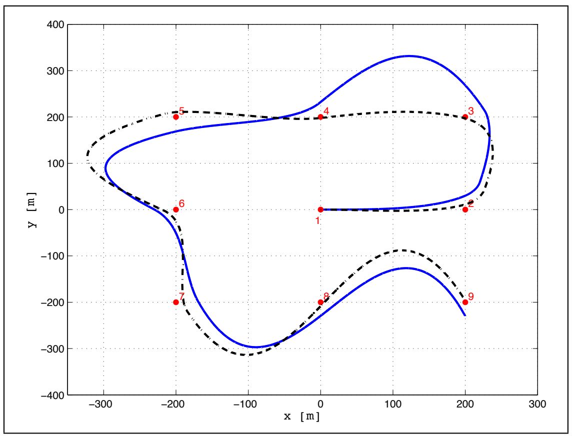

Figure 3: Random Trajectory: View in Horizontal Plane.

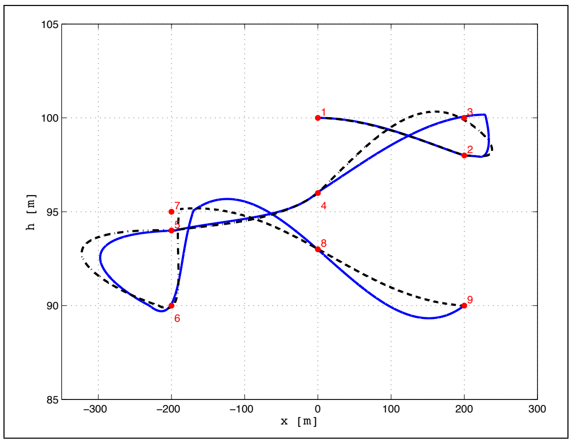

Figure 4: Random Trajectory: View in Vertical Plane.

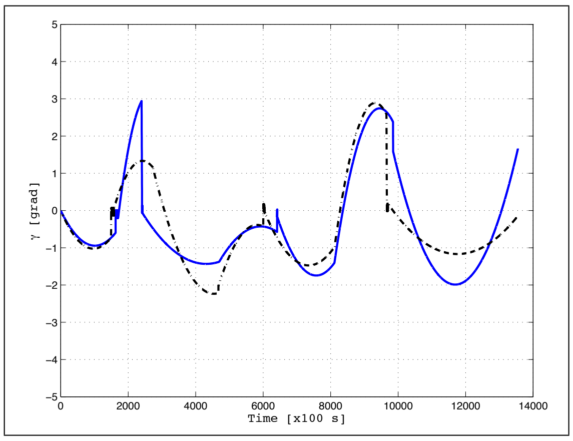

Figure 5: Random Trajectory: Climb Angle.

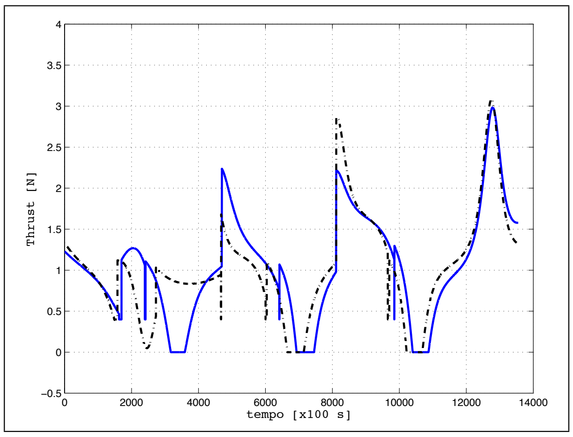

Figure 6: Random Trajectory: Thrust.

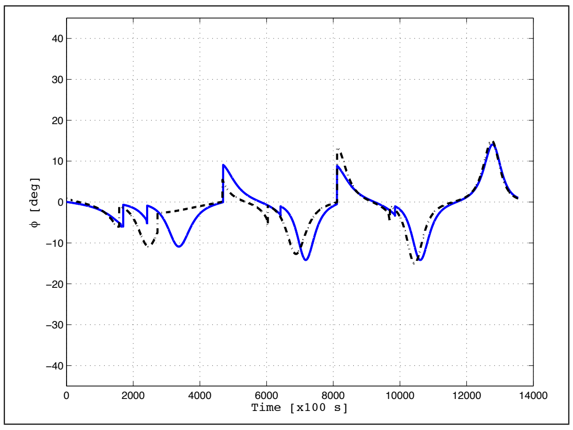

Figure 7: Random Trajectory: Roll Angle.

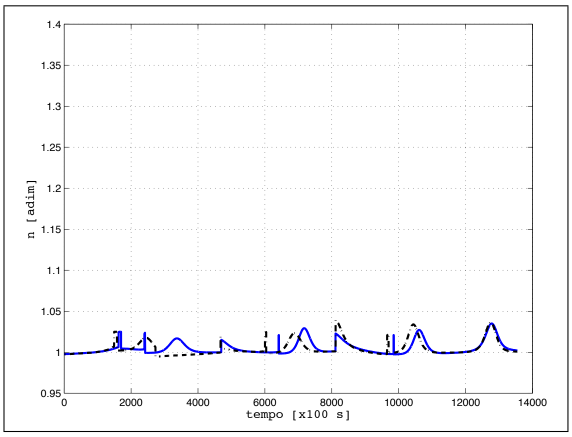

Figure 8: Random Trajectory: Load factor.

## 5.2 Climb Turn Trajectory

With this type of waypoint distribution we want to force the algorithm in the error procedure. When the vehicle will be in the waypoint number 2, its bank angle is negative and the spline for waypoints 2, 3 and 4 is not directly available. The error procedure will lead the platform to start a right turn so that the spline can be calculated.

| | **E. S. C.** | **T. P. C.** |
|---|---|---|
| **V** | 10 m/s | 10 m/s |
| $\chi_{in}$ | -30 deg | -30 deg |
| $\gamma_{in}$ | 10 deg | 10 deg |
| $w_1$ | $10^8$ | $10^6$ |
| $w_2$ | $10^5$ | $10^4$ |
| $w_3$ | 1 | 10 |
| $w_4$ | 1 | $10^4$ |
| $w_5$ | 1 | $10^4$ |

Table 3: Input File Parameters Value Used in Climb Turn Trajectory Simulation.

Figures from 9 to 15 refer to this particular trajectory. If we choose to pay more attention to the energy saving we have to give up for the precision of the trajectory. The variation of $\gamma$ and $T_e$ are more aggressive than in the precision task even if the time required to conclude the trajectory is reduced. When the energy saving is required the $T_e$ time history shows one high peak but subsequently the idle setting is kept for longer time.

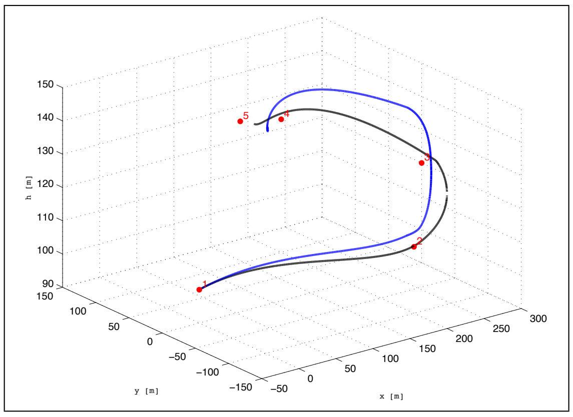

Figure 9: Climb Turn Trajectory: 3D View

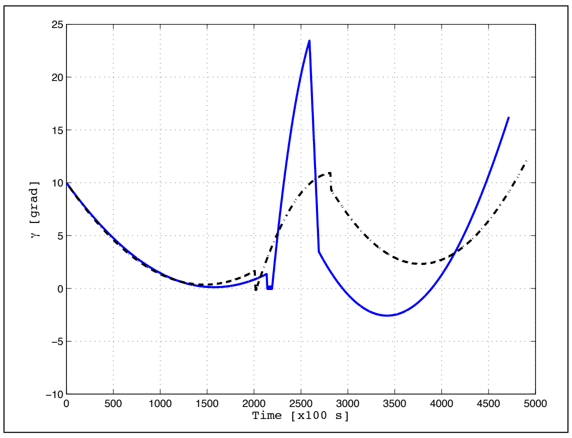

Figure 12: Climb Turn Trajectory: Climb Angle.

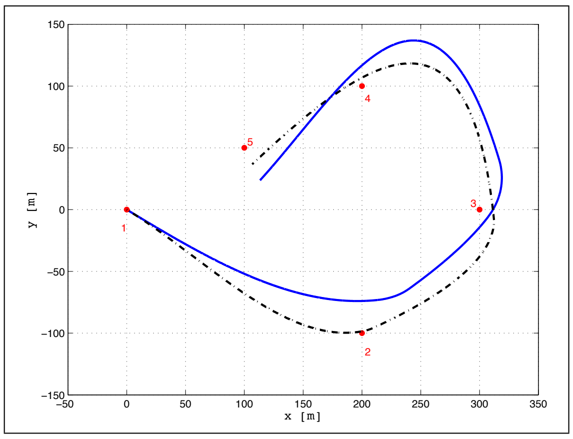

Figure 10: Climb Turn Trajectory: View in Horizontal Plane.

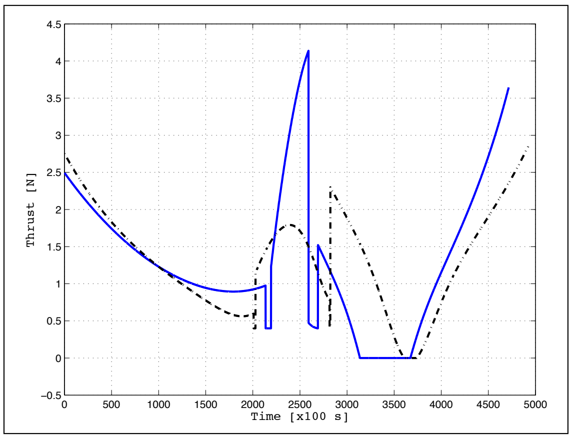

Figure 13: Climb Turn Trajectory: Thrust.

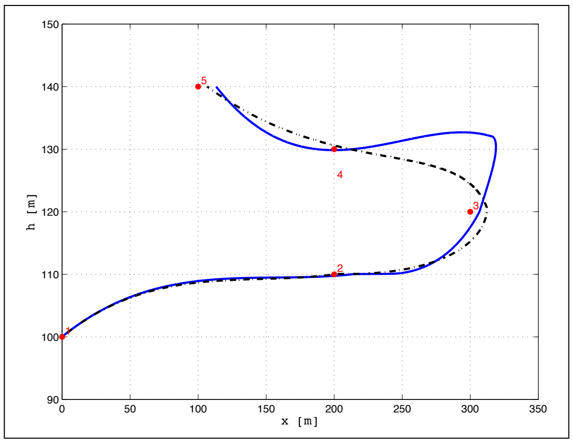

Figure 11: Climb Turn Trajectory: View in Vertical Plane.

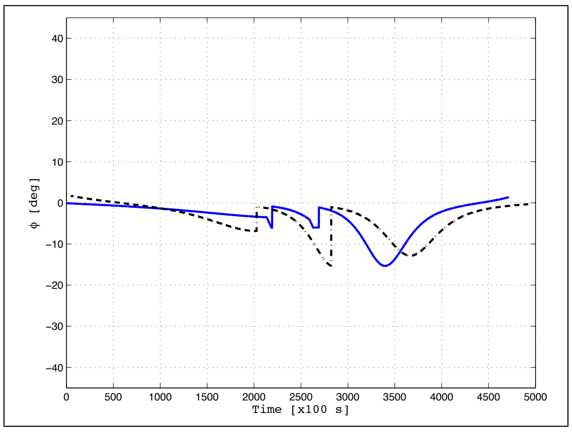

Figure 14: Climb Turn Trajectory: Roll Angle.

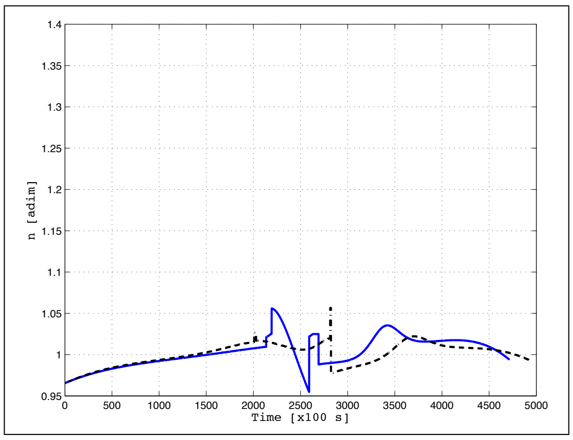

Figure 15: Climb Turn Trajectory: Load Factor.

## 5.3 "Butterfly" Trajectory

This case represents the most limiting waypoint distribution in which an UAV can operate (see from Fig. 16 to Fig. 21) as heading from crosstrack corrections are continuously updated. The task must also be sequentially repeated.
In this case no substantial difference between the two separate optimization procedures is highlighted. The total time necessary to complete the whole task is the same.

| | **E. S. C.** | **T. P. C.** |
|---|---|---|
| **V** | 10 m/s | 10 m/s |
| $\chi_{in}$ | -45 deg | -45 deg |
| $\gamma_{in}$ | 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 | $10^4$ |

Table 4: Input File Parameters Value Used in Butterfly Trajectory Simulation.

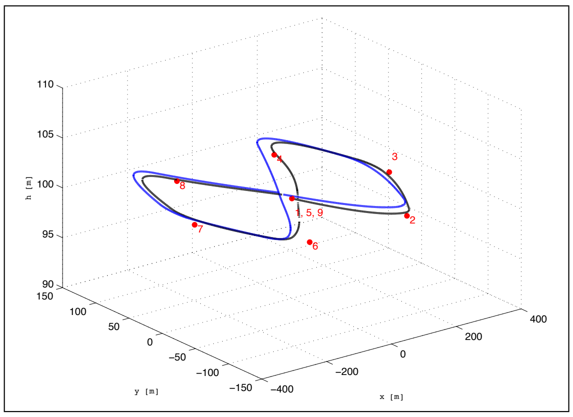

Figure 16: Butterfly Trajectory: 3D View.

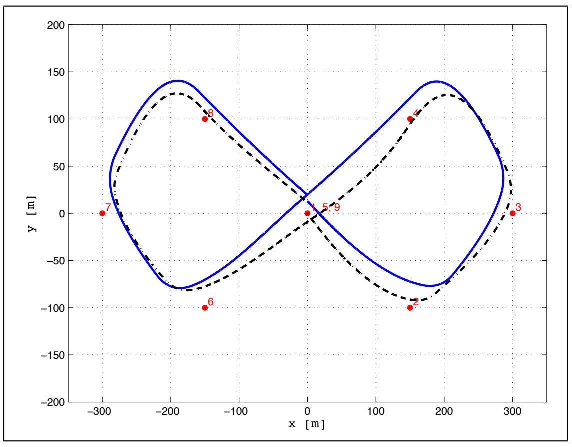

Figure 17: Butterfly Trajectory: View in Horizontal Plane.

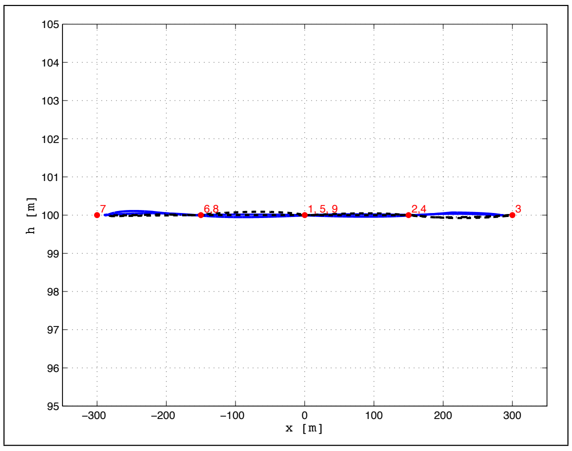

Figure 18: Butterfly Trajectory: View in Vertical Plane.

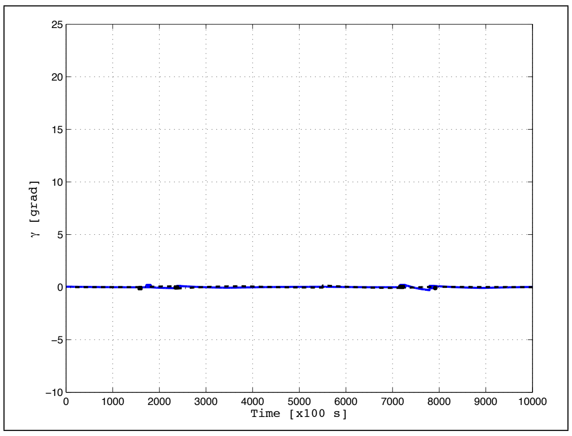

Figure 19: Butterfly Trajectory: Climb Angle.

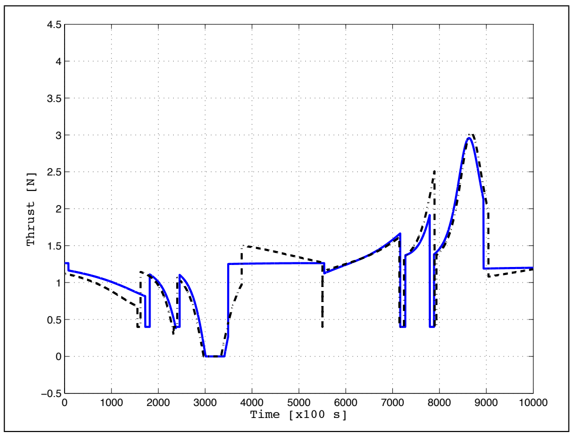

Figure 20: Butterfly Trajectory: Thrust.

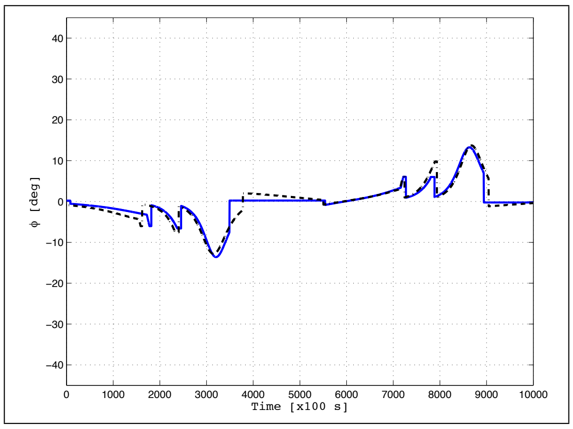

Figure 21: Butterfly Trajectory: Roll Angle.

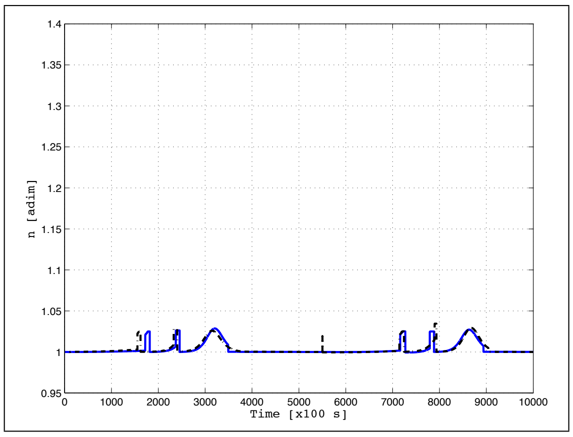

Figure 22: Butterfly Trajectory: Load Factor.

## 6 Conclusions

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

The algorithm is fast enough to be applied to pre-flight trajectory definition. An extension to real time is potentially possible with a minimal degradation in term of fitness for the best individual (reduce the number of generations). An alternative way to provide faster runs is the reduction of intermediate waypoints (plan limited to corner waypoints). Nevertheless, if the number of waypoints given for a trajectory plan is too low the optimization may be weak (undersampling). Some numerical exercise is required to tune the minimum number of waypoints in order to obtain a feasible trajectory.

The micro-GA implementation is beneficial as random search is performed on a very limited number of individuals (small population).

Optimal trajectories are affected by the type of strategy, either energy saving or precision tracking. The user is expected to decide which of the two should fit the application.

As expected, if the weights for trajectory precision are high the optimal radius of waypoints decreases. Differently, enhancing the energy balance - despite the precision - the algorithm provides trajectories with smoother changes. The effect of spline assembly is also reduced (corner smoothing) depending on proximity of waypoints. The size of inputs is also affected by the design strategy: in the energy save case the average control inputs are small and, occasionally, controls may show moderate “jumps”.

## References

[1] B. Pralio, G. Guglieri, F. Quagliotti, “Design and Performance Analysis of a Micro Aerial Vehicle Concept”, *2nd AIAA Unmanned Unlimited Systems, Technologies and Operations Conference*, San Diego, USA, 2003.

[2] B. Pralio, G. Guglieri, F. Quagliotti, “Flight Control System Design foa a Micro Aerial Vehicle”, *Aircraft Engineering and Aerospace Technology*, Vol. 78, 2006, pp. 87-97.

[3] R. Fantinutto, G. Guglieri, F. Quagliotti, “Flight Control System Design and Optimisation with a Genetic Algorithm”, *Aerospace Science and Technology*, Vol. 9, 2005, pp. 73-80.

[4] L.E. Kavraki, F. Lamiraux, and C. Holleman, “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces”, *IEEE Transaction on Robotics and Automation*, 12(4), pages 566-580.

[5] S.M. LaValle, J.J. Kuffner, “Randomized Kinodynamic Planning”, *IEEE International Conference on Robotics and Automation*, July 1999.

[6] S.J. QinAn, Department of Chemical Engineering The University of Texas, Austin, T.A. Badgwell, Department of Chemical Engineering Rice University Houston “Overview of Industrial Model Predictive Control Technology”

[7] A. Bemporad, “Reference Governor for Constrained Nonlinear Systems”, *IEEE Transaction on Automatic Control*, vol. AC-43, no.3, pages 415-419,1998.

[8] A. Bemporad, A. Casavola, E. Mosca, "Nonlinear Control of Constrained Linear System via Predictive Reference Management", *IEEE Transaction on Automatic Control*, vol. AC-42, no.3, pages 340-349, 1997.

[9] R. Murray, Control and Dynamical Systems, California Institute of Technology, Pasadena; J.Doyle, J. Marsden, Calthec Control and Dynamical System; G. Balas, University of Minnesota, "Robust Nonlinear Control Theory With Application to Aerospace Vehicles".

[10] D.E. Goldberg, "Algorithm in Search, Optimization and Machine Learning", *Addison Wesley, Reading*, USA, 1989.

[11] David L. Carroll, University of Illinois, "Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers", *Developments in Theoretical and Applied Mechanics*, vol. XVIII, eds. H. Wilson

[12] David L. Carroll, "Chemical Laser Modeling With Genetic Algoritms", *AIAA Journal*, Vol. 34, N.2, pp.338-46