An Evolutionary Computation Approach to Three-Dimensional Path Planning for Unmanned Aerial Vehicles with Tactical and Kinematic Constraints

J. Kok [1], T. Bruggemann [1], and F. Gonzalez [1]

[1] *Cooperative Research Centre for Spatial Information and the Australian Research Centre for Aerospace Automation, Brisbane Airport, Queensland, 4008, Australia*

Abstract

This paper presents a novel evolutionary computation approach to three-dimensional path planning for unmanned aerial vehicles (UAVs) with tactical and kinematic constraints. A genetic algorithm (GA) is modified and extended for path planning. Two GAs are seeded at the initial and final positions with a common objective to minimise their distance apart under given UAV constraints. This is accomplished by the synchronous optimisation of subsequent control vectors. The proposed evolutionary computation approach is called synchronous genetic algorithm (SGA). The sequence of control vectors generated by the SGA constitutes to a near-optimal path plan. The resulting path plan exhibits no discontinuity when transitioning from curve to straight trajectories. Experiments and results show that the paths generated by the SGA are within 2% of the optimal solution. Such a path planner when implemented on a hardware accelerator, such as field programmable gate array chips, can be used in the UAV as on-board replanner, as well as in ground station systems for assisting in high precision planning and modelling of mission scenarios.

Keywords: Evolutionary computation, genetic algorithm, path planning, trajectory generation, unmanned aerial vehicles.

Introduction

Many civilian applications of UAVs (such as aerial surveillance, search and rescue or asset inspection) require the UAVs to execute precise manoeuvres whilst taking into account the field-of-view of on-board camera and sensor payload in five-dimensional (position and orientation) state space. Recently, optimal five-dimensional state space UAV path planning which considers tactical and kinematic constraints has been solved by [1]. Their work extends Dubin's shortest path theory [2] from a two-dimensional plane into five-dimensional state space. The authors show that the optimal path problem can be solved by using either geometric method or numerical approach. The geometric method can produce optimal paths rapidly (a few seconds) but due to its analytical and deterministic nature, it lacks the flexibility to include obstacles and terrain constraints necessary in practical UAV applications. The numerical approach required a large computational effort and processing time (a few minutes) to find the optimal solution, thus they are not competitive against near-optimal and rapid path planning techniques in view of an on-board path planning system.

Genetic algorithm (GA) is a powerful problem-generic population-based metaheuristic optimisation method associated to the internationally acclaimed field of evolutionary computation [3]. In the past, GAs have been used as an evolutionary computation approach to solving the path planning problem by optimising various path representations, such as B-Spline curves [4] and transitional waypoints [5]. Another key advantage of applying GAs in path planning is that they may be readily applied in hardware accelerators, such as field programmable gate arrays (FPGAs), for rapid real-time parallel computation of flight paths on-board UAVs [6]. Implementing numerical or analytical based approaches on FPGA's requires much engineering effort and due to the nature of the algorithms cannot take advantage of the parallel computing power provided by FPGAs [6]. Thus far, the direct application of evolutionary computation approaches to solving a UAV control optimisation problem with tactical and kinematic constraints has not been well explored.

For this purpose, this paper presents a GA for optimising a sequence of control inputs for a simplified non-linear UAV system dynamics with the additional consideration of tactical and kinematic constraints. These constraints are necessary as simplified models are not always tangible in real-world applications, particularly when the UAV has on-board downward or forward facing cameras/sensors which are orientation dependent [7], [8]. The simulated flight experiments show that our proposed evolutionary computation approach is seen to be effective in rapidly generating a near-optimal flight path when the two points are sufficiently far apart (at least four times the minimum turning radius). Note that, this limitation has minor implications as most real-world applications often cover a large expanse. Additionally, the resulting path plan exhibits no discontinuity when transitioning from curve to straight trajectories.

Problem Definition

The state equations are represented by first order differential constraints on the system dynamics:

$$\dot{q} = f\big(q(t), u(t)\big) \tag{1}$$
$$q(0) = q_i \tag{2}$$
$$q\big(t_f\big) = q_f \tag{3}$$

where $q(t) = \big(x(t), y(t), z(t), \psi(t), \gamma(t)\big)^T$ denotes the state vector and $u(t) = \big(\eta(t), \mu(t)\big)^T$ denotes the control vector. Here, $x(t), y(t), z(t) \in \mathbb{R}$ denotes the x coordinates, y coordinates, and z coordinates, respectively, of the instantaneous position, $\psi(t) \in \mathbb{R}$ denotes the heading angle measured counter clockwise from the x-axis in the x-y plane, and $\gamma(t) \in \mathbb{R}$ denotes the flight path angle measured in reference to the horizontal x-y plane.

The mission tactical constraint is subjected to given initial and final (goal) configurations specified by position and orientation vectors:

$$q_i = (x_i, y_i, z_i, \psi_i, \gamma_i) \tag{4}$$
$$q_f = \big(x_f, y_f, z_f, \psi_f, \gamma_f\big) \tag{5}$$

f

The UAV kinematic model used to characterise the dynamic motion of an UAV in 5 dimensional state space is as follows [9]:

$$\dot{x} = V \cos\psi \cos\gamma \tag{6}$$
$$\dot{y} = V \sin\psi \cos\gamma \tag{7}$$
$$\dot{z} = V \sin\gamma \tag{8}$$
$$\dot{\psi} = \eta \tag{9}$$
$$\dot{\gamma} = \mu \tag{10}$$

Here V denotes the magnitude (constant) of the UAV velocity vector $V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$.

The UAV kinematic constraint is subjected to a maximum curvature c_{max}:

$$c(t) \leq c_{max} \tag{11}$$

where the instantaneous path curvature $c(t)$ is given by [1],

$$\begin{aligned} c(t) &= \sqrt{\dot{\psi}^2(t)\cos^2\gamma(t) + \dot{\gamma}^2(t)} \\ &= \sqrt{\eta^2(t)\cos^2\gamma(t) + \mu^2(t)} \end{aligned} \tag{12}$$

The path planning problem is defined as to obtain a minimum flight path between the given initial position and final position by optimising the control inputs $\eta \in \mathbb{R}$ and $\mu \in \mathbb{R}$ which are the time rate of change in heading angle and flight path angle, respectively. The length of the flight path is given by,

$$J(q_i, q_f, u) = \int_0^{t_f} dt \tag{13}$$

where q_i and q_f denote the initial and final states for equations in Eqn 1 to Eqn 5, which are given in terms of the vectors $q_i = (x_i, y_i, z_i, \psi_i, \gamma_i)$ and $q_f = (x_f, y_f, z_f, \psi_f, \gamma_f)$. u denotes the control sequence from q_i to q_f. To solve this problem we introduce an approach using genetic algorithms in the following section.

Genetic Algorithm

Genetic algorithm (GA) is the most popular type of EA. Fundamentally, it is a heuristic optimisation method based upon mechanisms of biological evolution: selection, reproduction, and replacement. A pseudo code describing the working principle of a GA is illustrated in Fig. 1.

```
Genetic Algorithm

Initialise Population

WHILE (Termination criteria NOT MET)
      Selection
      Reproduction
      Replacement
END-WHILE
```

Fig. 1: Pseudo code of a genetic algorithm

Initialise Population

The first generation of population is initialised randomly as the generality of GAs does not require *a priori* knowledge of the problem.

Termination

The generational process is iterated until termination criteria are met. Common criteria used are maximum generation, desired solution, maximum computational run-time, convergence plateau, or any combinations of the above.

Selection

The objective of the *selection* operator is to establish a selection pressure in which fitter solutions in a population have better chance of survival as evolution progresses. Some reputable methods include tournament selection, roulette wheel selection, ranking selection, and truncation selection [10], [11].

Reproduction

The objective of the *reproduction* operator is to establish a balance between exploitation (crossover) and exploration (mutation). The crossover operation exchanges portions of information to create possibly better solutions, whereas the mutation operation randomly alters a portion of the solution to maintain diversity in the population. A range of reproduction operators can be found in [12].

Replacement

The objective of the *replacement* operator is to establish an update scheme for the offspring population with the parent population. There are three fundamental replacement schemes commonly employed: generational replacement, in which offspring population overwrites all of parent population; environmental replacement, in which worst solutions are deleted incrementally until population reaches a predefined minimum size; and elitist replacement, in which best parent solutions are preserved [13].

These biological mechanisms of evolution are the fundamental building blocks for typical GAs. From this set of evolutionary components, different GAs are designed and implemented by the transformation of relevant features. For instance, the GA used in this work incorporates convergence plateau termination criteria, tournament selection, crossover and mutation reproduction, and generational replacement.

Synchronous Genetic Algorithm for Path Planning

In this section, the proposed synchronous genetic algorithm (SGA) for three-dimensional path planning is presented. The purpose of the SGA in the overall path generation process is to optimise instantaneous control vectors, $\eta(t)$ and $\mu(t)$, for the UAV subjected to tactical and kinematic constraints, by which path length is minimised simultaneously by initialising two GAs at the initial position and final position. The steps are as follows.

Step 1: Initial position and orientation, $q(0) = q_i$, final position and orientation, $q(t_f) = q_f$ and maximum curvature, c_{max} , are defined.

Step 2: Two GAs, GA_1 and GA_2, are initialised at the given initial position and final position, respectively, with the tactical constraints considered. GA_1 incrementally optimises the state $q(t+1)$ starting from $q(0)$, while GA_2 decrementally optimises the state $q(t-1)$from $q(t_f)$ with the kinematic constraints considered. The path length between states of the two GAs is minimised, hence the two GAs will eventually orientate towards each other.

Step 3: The subsequent states of GA_1 and GA_2 are updated according to Eqn 6 to Eqn 10. Both GAs now optimise the subsequent control vectors with path length referenced to current corresponding state vectors.

Step 3 is iterated until both GAs are orientated towards each other, which indicates a forward flight path for connecting the two control sequences. Fig. 2 shows the pseudo code of the SGA (top) and illustrates the effect of both GAs control sequences (bottom-left) and the outcome of the path generation process (bottom-right).

Synchronous Genetic Algorithm

$t = 0$
Initialise GA_1 at q_i
Initialise GA_2 at q_f
Set objective of GA_1 and GA_2 to minimise path length from $q(i+t)$ to $q(f-t)$

WHILE (orientation at $q(i+t)$) $\neq$ (180°- orientation at $q(f-t)$)
    $t = t+1$
    Set objective of GA_1 and GA_2 to minimise path length from $q(i+t)$ to $q(f-t)$
    Run GA_1 and GA_2
END-WHILE

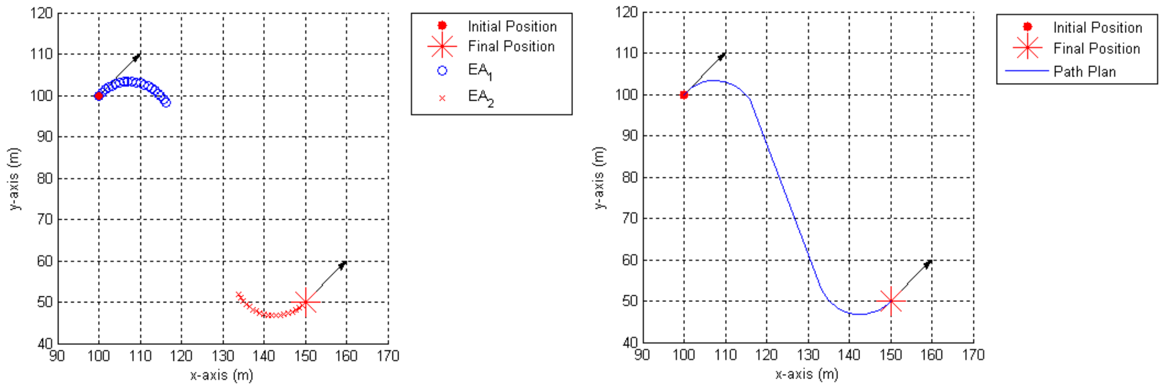

Fig. 2: (top) Pseudo code of the SGA. (bottom-left) Effects GA_1 and GA_2. (bottom-right) Overall path generated by the SGA

Experiments and Results

The experiments are conducted across four specifically chosen set of initial and final conditions for verifying the robustness of the proposed evolutionary computation approach to path planning. All four initial position and final position are identical but orientations are set to the four extreme possible configurations at 180° intervals. The case studies and results are tabulated in Table 1. The optimal path lengths for each case were calculated using the geometric method described in [1]. The maximum curvature c_{max} is 0.2m^{-1}. The velocity constant V is 1 m/s. The discrete time step ΔT is 1s. The running time in MATLAB on quad-core desktop PC with 2GB RAM was less than 10 seconds per path. Fig. 3, Fig. 4, Fig. 5 and Fig. 6 show the path generated between the initial position and final position (top-left), curvature profile (top-right), heading angle profile (bottom-left), flight path angle profile (bottom-right) by the SGA for Case 1, Case 2, Case 3, and Case 4, respectively. Note that, with the above velocity and time configuration, each second is equivalent to one metre of flight path length. Overall, these figures highlight that the generated paths satisfy the tactical and kinematic constraints, whilst achieving a near optimal result (within 2% of optimal). Additionally, the resulting path plan exhibits no discontinuity when transitioning from curve to straight trajectories.

Table 1: Case study used in this work

Case Study		Position (x, y, z) (m)	Orientation (ψ, γ) (deg)	SGA Path Length (m)	Optimal Path Length (m)
Case 1	Initial	(100,100,100)	(0,0)	88.32	87.86
	Final	(150,150,150)	(0,10)		
Case 2	Initial	(100,100,100)	(180,0)	95.26	94.17
	Final	(150,150,150)	(0,10)		
Case 3	Initial	(100,100,100)	(180,0)	102.74	100.80
	Final	(150,150,150)	(180,10)		
Case 4	Initial	(100,100,100)	(0,0)	94.43	93.35
	Final	(150,150,150)	(180,10)		

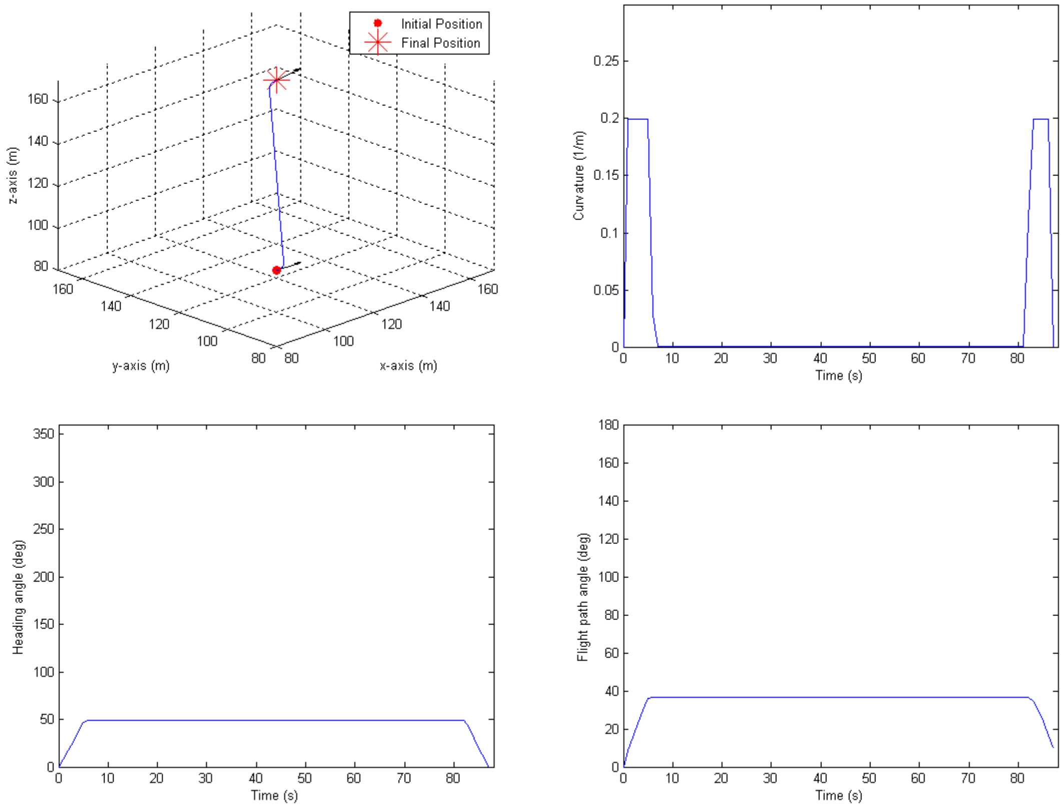

Fig. 3: Results for Case 1. (top-left) Path generated. (top-right) Curvature profile. (bottom-left) Heading angle profile. (bottom-right) Flight path angle profile.

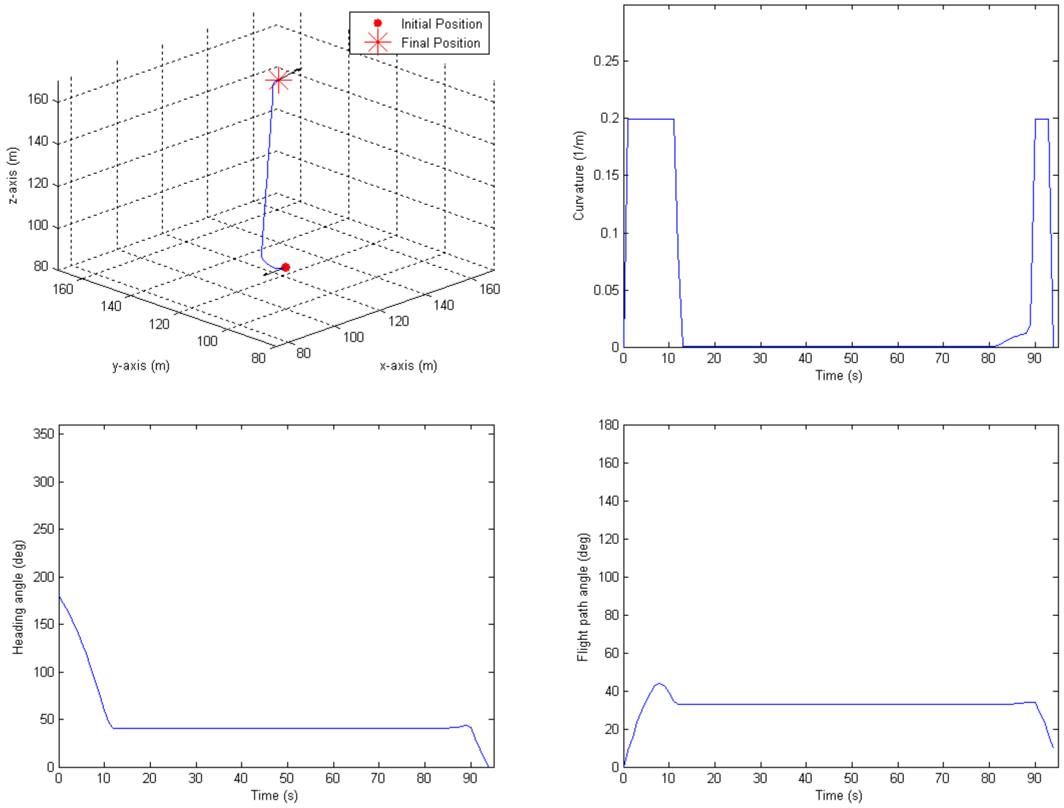

Fig. 4: Results for Case 2. (top-left) Path generated. (top-right) Curvature profile. (bottom-left) Heading angle profile. (bottom-right) Flight path angle profile.

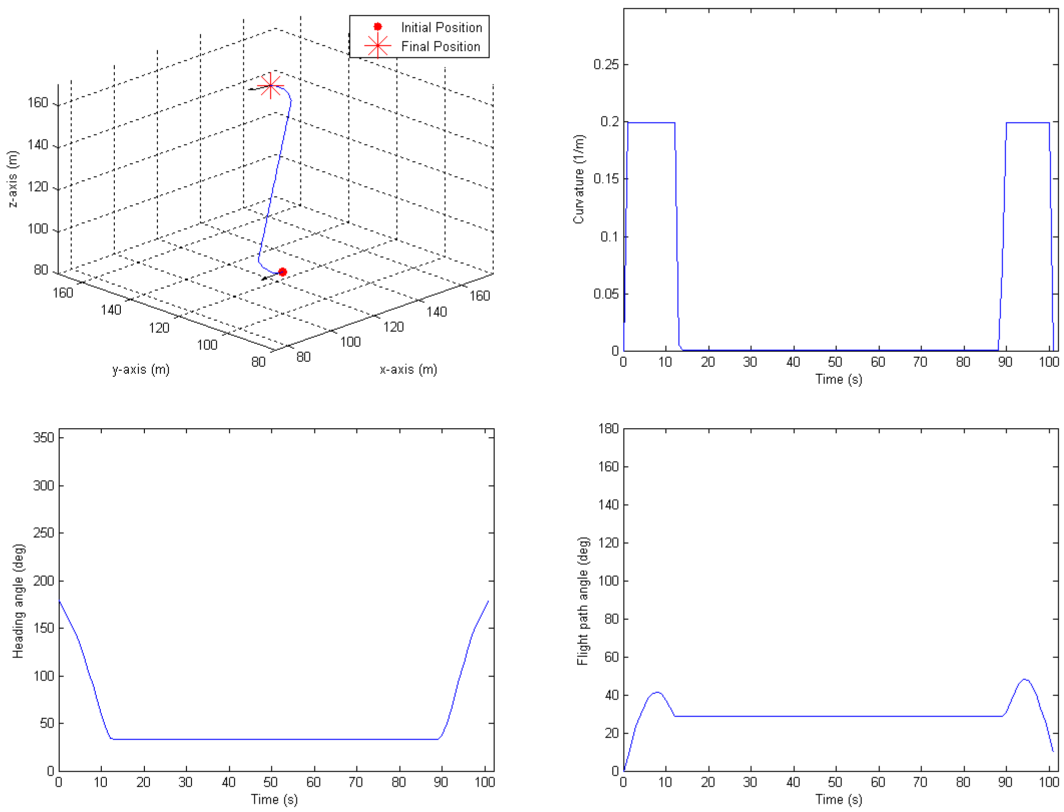

Fig. 5: Results for Case 3. (top-left) Path generated. (top-right) Curvature profile. (bottom-left) Heading angle profile. (bottom-right) Flight path angle profile.

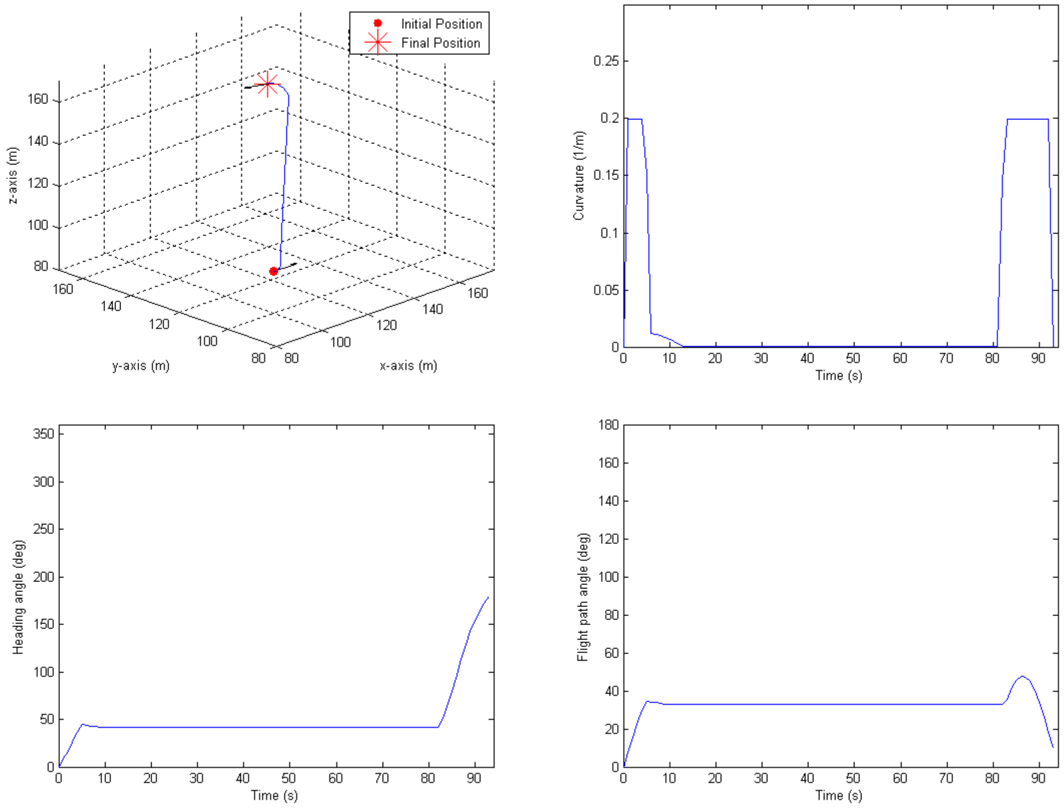

Fig. 6: Results for Case 4. (top-left) Path generated. (top-right) Curvature profile. (bottom-left) Heading angle profile. (bottom-right) Flight path angle profile.

Conclusions

In this paper, the SGA is proposed as an evolutionary computation approach to path generation for UAVs with tactical and kinematic constraints. The robustness behind the evolutionary computation approach is due to the GAs ability of subjecting candidate path solutions to the tactical orientations and UAV kinematic constraints. By formulating the overall path planning problem as a control optimisation problem, the SGA was able to optimise a sequence of control vectors for generating a path plan with near-optimal solution quality. The resulting path plan exhibits no discontinuity when transitioning from curve to straight trajectories. The proposed evolutionary computation approach is effective in optimising and generating a flight path when the two points are sufficiently far apart (at least four times the minimum turning radius). Note that, this limitation has minor implications as most real-world applications often cover a large expanse. Future work will analyse the performance of the algorithm when obstacles and terrain constraints are included into the path planning problem.

Acknowledgements

The work has been supported by the Cooperative Research Centre for Spatial Information, whose activities are funded by the Australian Commonwealth's Cooperative Research Centres Programme.

References

1. Hota, S. and Ghose D., "Optimal path planning for an aerial vehicle in 3D space", *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 4902-4907.

2. Dubins, L.E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", *American Journal of Mathematics*, Vol. 79, No. 11, 1957, pp. 497-516.

3. Michalewicz, Z., *Genetic algorithms + data structures = evolution programs*, Springer-Verlag, 3rd rev. and extended edition, 1996.

4. Mittal, S. and Deb, K., "Three-dimensional online path planning for UAVs using multiobjective evolutionary algorithms", *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore, September, 2007, pp. 3195-3202.

5. Rathbun, D., Kragelund, S., Pongpunwattana, A. and Capozzi, B., "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments", *Proceedings of the 21st Digital Avionics Systems Conference*, Vol. 2, 2002, pp. 8D2(1)-8D2(12).

6. Kok, J., Gonzalez, L.F. and Kelson, N.A., "FPGA implementation of an evolutionary algorithm for autonomous unmanned aerial vehicle on-board path planning", *IEEE Transactions on Evolutionary Computation*, 2012 (In press).

7. Bruggemann, T.S. and Ford, J.J., "Guidance of aircraft in periodic inspection tasks", *Proceedings of* the *Australian Control Conference*, University of Melbourne, Melbourne, VIC, November 10-11, 2011.

8. Saunders, J. and Beard, R., "Tracking a Target in Wind Using a Micro Air Vehicle with a Fixed Angle Camera", *IEEE American Control Conference*, Big Sky MT, 2009.

9. Ambrosino, G., Ariola, M., Ciniglio, U., Corraro, F., De Lellis, E. and Pironti, A., "Path Generation and Tracking in 3-D for UAVs", *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 4, July, 2009.

10. Goldberg, D.E. and Deb, K., "A comparative analysis of selection schemes used in genetic algorithms", *Proceedings of Foundations of Genetic Algorithms 1*, Morgan Kaufmann Publishers, 1991, pp. 69-93.

11. Blickle, T. and Thiele, L., "A comparison of selection schemes used in evolutionary algorithms", *Evolutionary Computation*, Vol. 4, No. 4, 1996, pp. 361-394.

12. Spears, W.M., *Evolutionary algorithms: The role of mutation and recombination.* Natural Computing Series. Springer, 2000.

13. Liefooghe, A., Laetitia, J., Legrand, T., Humeau J. and Talbi E., "ParadisEO-MOEO: A software framework for evolutionary multi-objective optimization", *Advances in Multi-Objective Nature Inspired Computing*, Studies in Computational Intelligence, 2010, Vol. 272, Springer Berlin / Heidelberg, pp. 87-117.