

# ADAPTIVE EVOLUTIONARY SEARCH ALGORITHM WITH OBSTACLE AVOIDANCE FOR MULTIPLE UAVS

**Barry Carruthers, Euan W. McGookin, David J. Murray-Smith**

*Centre for Systems and Control,  
Department of Electronics and Electrical Engineering,  
University of Glasgow, Glasgow, U.K.  
Telephone: +44-141-330-6137  
Email: {b.carruthers, e.mcgookin, d.murray-smith}@elec.gla.ac.uk*

**Abstract:** Unmanned air vehicles (UAVs) can be used for surveillance and searching, usually with a specified goal i.e. a given area to search or, person or object to find. In these cases, multiple UAVs can be more effective for searching than a single vehicle due to the greater area of coverage that they offer. This paper outlines a search methodology, based on evolutionary algorithms (EAs), which guides a swarm of UAVs throughout a search mission. The autonomously navigating vehicles also avoid unforeseen obstacles via reactive obstacle sensing. *Copyright © 2005 IFAC*

**Keywords:** Autonomous vehicles, obstacle avoidance, search methods.

## 1. INTRODUCTION

Unmanned air vehicles (UAVs) are robotic aircraft that can fly with either a remote input from a ground-based operator, or can fly autonomously without human intervention. UAVs and their potential applications have become increasingly prominent in recent years. In a military application, multiple specialized vehicles may be more effective, and cheaper, than a single UAV which must undertake all search and attack tasks by itself, see (Jin, *et al.*, 2003). In civilian terms, UAVs can be used for emergency tasks such as searching for survivors in dangerous environments or locating spills or leaks of dangerous chemicals etc., see (Flint, *et al.*, 2002; Bruemmer, *et al.*, 2002).

Controlling multiple vehicles has also become a growing area of research. Due to obvious limitations when working with multiple aircraft, i.e. cost of equipment and experimentation, previous research has used virtual simulation, see (Fierro and Das, 2003; Reynolds, 1987; Koo and Shahruz, 2001), land-based vehicles, see (Fredslund and Matarić, 2002) and virtual missiles (Hughes, 2002). Some work in this field has used the “solution searching”

principle of Genetic Algorithms (GAs), see (Srinivas and Patnaik, 1994), as the basis for searching of physical areas. As the algorithms imitate the natural Darwinian selection process, it is conceptually simple to see how such algorithms can be applied to searching physical space within which there are high interest or high “fitness” areas. This has prompted research using genetic and evolutionary algorithms for search, see (Goldberg, 1989; Raphael and Smith, 2003), as well as vehicle route planning, see (Hughes, 2002; Kim, *et al.*, 1997; Creaser and Stacey, 1999). Particularly relevant to this paper and future progress, is work that uses GAs to search multidimensional space, see (Grevera and Meystel, 1990), UAV guidance by GA rules, see (Marin, *et al.*, 1999), and the use of GAs for guidance and control of multiple searching UAVs, see (Enns, *et al.*, 2002).

## 2. THE MISSION SYSTEM

This paper focuses on the problem of performing a blind search i.e. with no a-priori information about the search area. The paper presents a system to control a swarm of UAVs to perform the search task.

As the swarm has no knowledge of targets or obstacles, the system must guide the UAVs throughout the search area, whilst individuals must also be able to react to obstacles that they may encounter during their search.

The overall system can be described by a block diagram of the form shown in Fig. 1.

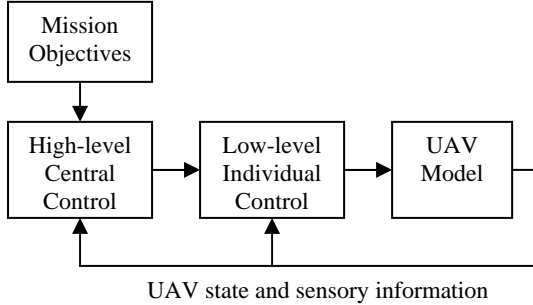


Fig. 1. The overall system architecture.

The control hierarchy is divided into high-level central control and low-level individual control. The high-level central control assumes ideal and full communication with each member of the swarm, which includes information gathered from each aircraft's relevant sensors. Proximity sensors indicate the possibility of a collision with an obstacle or another aircraft while a primary sensor (camera, IR sensor, etc.) indicates whether or not a target has been found. This allows central control to make intelligent decisions regarding the collective motion of the swarm and the searching process.

The low-level control block comprises a line-of-sight autopilot, flight control system and obstacle avoidance procedure. Low-level control is constructed as shown in Fig. 2. The flight control system and line-of-sight autopilot work together to allow each vehicle to navigate autonomously between waypoint  $x$ - $y$  coordinates. However, during the aircraft's flight between waypoints, unexpected obstacles may be encountered e.g. trees, power lines, flocks of birds, etc. which must be avoided. Obstacle avoidance is employed to use information from the aircraft's proximity sensors and subsequently generate an appropriate change in heading.

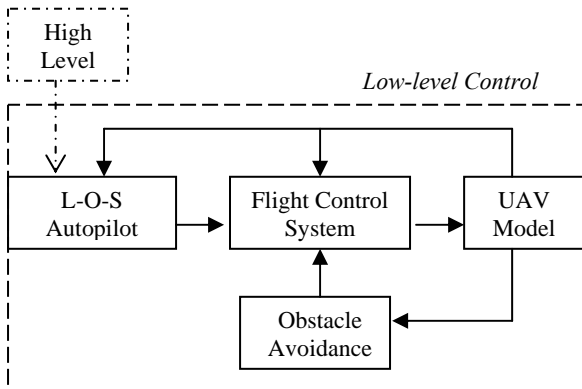


Fig. 2. Low-level control architecture.

## 2.1 The UAV Mathematical Model

This research uses a small perturbation, linear representation of a single main-rotor and tail rotor helicopter. For convenience, the model used is that of a Westland Puma helicopter but, although this is not typical of a vehicle that might be used for a swarm of UAVs, it is believed that principles of this control task can be investigated in a valid fashion using such a model. The model is given in standard state-space form as shown in Eqn. (1),

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} \quad (1)$$

where  $\mathbf{x}$  represents the state variables ( $u, v, w, p, q, r, \phi, \theta, \psi$ ),  $\mathbf{u}$  represents the helicopter control inputs ( $\theta_{0(1)}, \theta_{1S(1)}, \theta_{1C(1)}, \theta_{0(2)}$ ),  $\mathbf{A}$  is the system matrix and  $\mathbf{B}$  is the input distribution matrix. The system and input distribution matrices correspond to a steady-level flight trim-state. At this trim-state, the aircraft's speed,  $U_E$ , is 81.8 knots ( $42.08 \text{ ms}^{-1}$ ) and the altitude,  $Z_E$ , is 4031 ft (1228.65 m). In conjunction with the Euler transformation matrix, these values allow analysis of the aircraft's motion relative to the Earth-fixed axes ( $X_E, Y_E, Z_E$ ).

## 2.2 The Flight Control System

The flight control system design is based on a simple *PID* control method, see (Dorf and Bishop, 2001). The control system for each vehicle consists of four independent *PID* controllers that control small perturbations in surge velocity  $u$ , altitude  $z$ , yaw angle  $\psi$  and lateral (sway) velocity  $v$ . Each controller has the structure shown in Fig. 3.

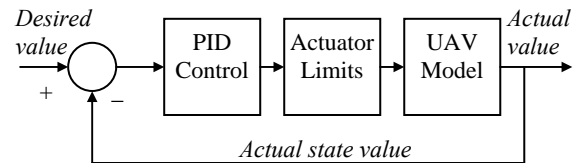


Fig. 3. The general structure for each *PID* controller.

## 2.3 Line-Of-Sight Autopilot

The four *PID* controllers provide sufficient overall flight control of the linearised Puma model such that the aircraft can autonomously navigate a course of pre-defined waypoints. The UAV navigates to each waypoint using simple trigonometry between current and desired coordinates, as shown in Fig. 4 and Eqn. (2).

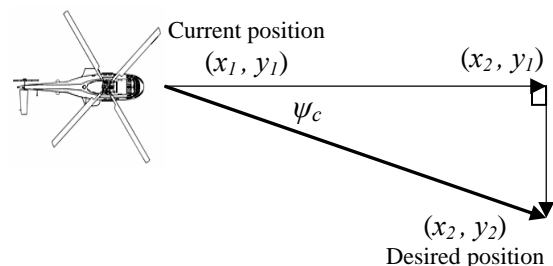


Fig. 4. Line-Of-Sight heading calculation.

$$\psi_c = \tan^{-1} \left( \frac{(y_2 - y_1)}{(x_2 - x_1)} \right) \quad (2)$$

## 2.4 Obstacle Avoidance

The final element of the lower-level individual vehicle control is provided by reactive obstacle avoidance. In this study, the mission assumes no a-priori information about the search area and this includes no knowledge of obstacles that a vehicle may encounter. Therefore each vehicle cannot plot a course around known obstacles but instead must react to proximity sensor readings supplied by sensors onboard each aircraft.

In this paper, each vehicle is assumed to have two forward-looking proximity sensors that indicate the presence of an object within a certain range and heading angle. The configuration of these sensors, shown in Fig. 5, is such that the aircraft knows whether the obstacle is starboard or port of its current heading. This simple configuration is employed with practical implementation considered. Assuming the size and shape of obstacles are unknown, an evasion manoeuvre cannot be planned.

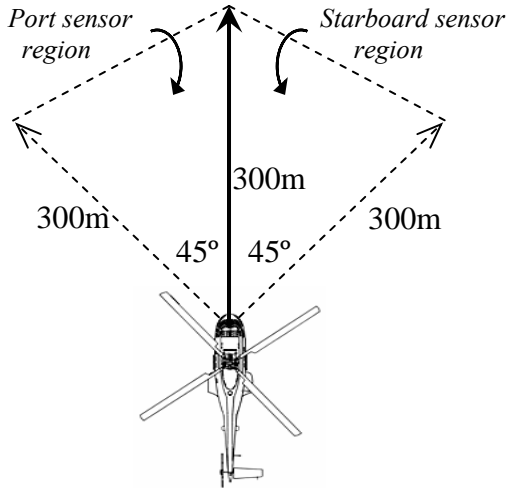


Fig. 5. Proximity sensor configuration.

If an obstacle is detected by either sensor i.e. an obstacle enters either of the sensor-sensitive regions, then an alert flag is raised. When one of the flags signals an obstacle, the aircraft control system adds or subtracts  $90^\circ$  to the previously desired heading angle. As the heading command is passed through a smoothing filter before becoming the desired heading command in the heading controller, this change is not seen as a step input. Instead, it is added as a first-order increase or decrease in heading command that results in a gradual change in heading, thus avoiding the obstacle. As the obstacle moves out of the sensitive region, the aircraft returns to an appropriate heading which will allow it to successfully intercept the next desired waypoint. The sensors do not distinguish between moving and static obstacles. This means that the obstacle

avoidance will also attempt to avoid moving objects such as a flock of birds.

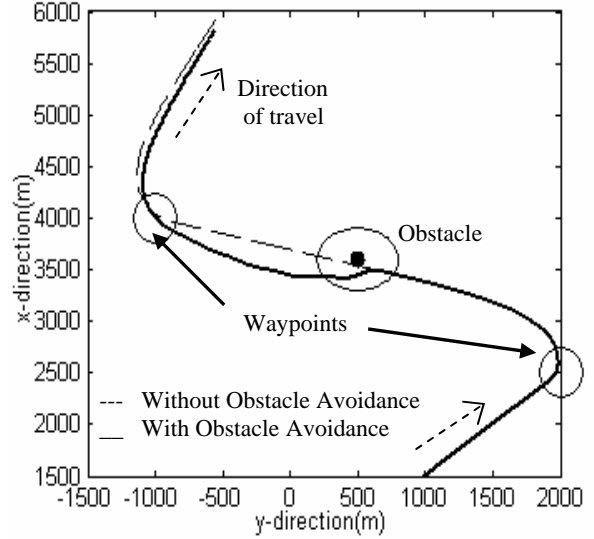


Fig. 6. The successful operation of the complete low-level control system.

With every aspect of the low-level control section working in tandem, each aircraft can autonomously navigate a course of waypoints whilst also avoiding any obstacles that they encounter. This is illustrated in Fig. 6.

This  $x$ - $y$  plot shows the flight path of the UAV as it navigates a course of waypoints. There is also an obstacle (shown as a black dot) placed directly between two waypoints (shown as circles). The ellipse around the obstacle indicates the point at which the UAV first detects the obstacle. This is actually a circle with a radius of 300 metres. The dashed flight path shows the flight path of the UAV without the aid of obstacle avoidance. The solid line shows how the addition of the obstacle avoidance system successfully directs the UAV around the obstacle as well as successfully intercepting the next waypoint. Note that a vehicle must only pass within an “acceptance radius” of each waypoint. This is effectively a sphere around each waypoint with a radius of 200 metres.

## 2.5. Central Swarm/ Mission Control

High-level control is provided by a central control system which is concerned primarily with the mission-specific task elements. This includes using any information given by the user in order to define mission objectives as indicated in Fig. 1. Although the simulations presented here feature UAVs flying in trim-state, this central block would ideally control each vehicle from take-off to mission completion and landing.

The central control system implements a two-stage strategy in order to carry out the blind search. As the UAV model is not suitable for modes of operation such as take-off and landing, simulations at this stage are concerned with the collective movement of the

UAV swarm to a search area, and the subsequent search of that area. For these two tasks, central control implements a Migration Mode and a Search Mode, respectively. The task of collective migration between search areas is presented in previous work by Carruthers and McGookin (2004). Therefore this paper focuses on the search technique incorporating a “taboo” element, see (Glover, 1989), as well as obstacle avoidance.

The Search Mode is responsible for guiding each UAV as the swarm enters the designated search area. The searching process is dictated by the creation of  $x$ - $y$  coordinates by Search Mode algorithms. The Search Mode comprises two algorithms for two different search conditions. The first is active when no target has been found, even partially, and therefore the search mode relies on an evolutionary-based algorithm to generate random  $x$ - $y$  coordinates within the search area. The second condition is that a target has been identified, although perhaps not pin-pointed. At this stage, the search process relies on an adaptive-evolutionary algorithm to generate subsequent  $x$ - $y$  coordinates. The definition of these conditions, and thus the selection of an appropriate algorithm, relies on target identification.

In order to identify targets, each UAV is equipped with a primary sensor (e.g. camera, IR detector, etc.) that continually scans the ground below, see Fig. 7. This sensor scans a circle directly below with a radius of 250 metres. For ease, the swarm is looking for a target circle of the same radius.

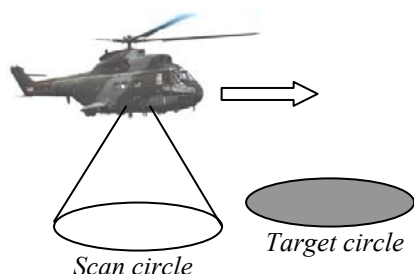


Fig. 7. Each UAV uses a downward-facing primary sensor to search for a target directly below.

As each UAV travels to a desired  $x$ - $y$  coordinate, it continually evaluates the ground below, thus recording  $x$ - $y$  coordinates with associated interest or “fitness” values. It is this fitness value that dictates the particular algorithm to be used throughout the remainder of the search. The fitness value can be any value between 0 and 1. A fitness value of ‘0’ indicates that no target is detected at that coordinate. However, if fitness is greater than 0, then a target has been identified to some extent (fitness equal to 1 would indicate that the target is completely covered by the scan circle). Eqn. (3) and Fig. 8 show how the overlapping of the scan circle and target circle is used to generate a fitness value.

$$\text{Fitness}(x_{UAV}, y_{UAV}) = \frac{A_O}{A_T} \quad (3)$$

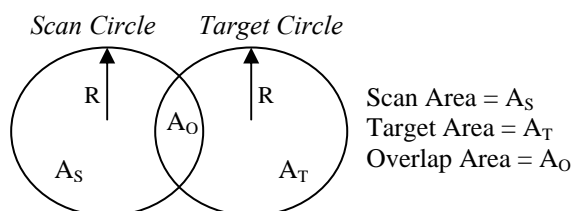


Fig. 8. Fitness evaluation via percentage overlap.

With each  $x$ - $y$  coordinate having an associated fitness value, the appropriate search algorithm can be selected. Both the evolutionary and adaptive-evolutionary algorithms use binary chromosomes as a basis. An initial chromosome comprising 8 binary bits is generated – 4 bits for the  $x$  component and 4 bits for the  $y$  component. These 8-bit chromosomes are decoded into the 2D coordinates which each UAV must intercept.

While a target has not been found (Fitness=0), the evolutionary algorithm is chosen to generate the search coordinates as shown in Fig. 9. As a target has not been found, the mutation rate between successive chromosomes is set at maximum i.e. it is possible that all 8-bits may change. This is to ensure that no region of the search area is unduly focused upon. In order to ensure that the swarm search as much of the area as quickly as possible, a ‘taboo’ element is introduced. This means that as a new coordinate is generated by the evolutionary algorithm, it is checked against a list of locations that have already been generated and searched. If the new point is within a 200m radius of a previously searched location, a new chromosome is created and subsequently checked. This continues until a target is found, the search area has been fully searched, or the search is terminated.

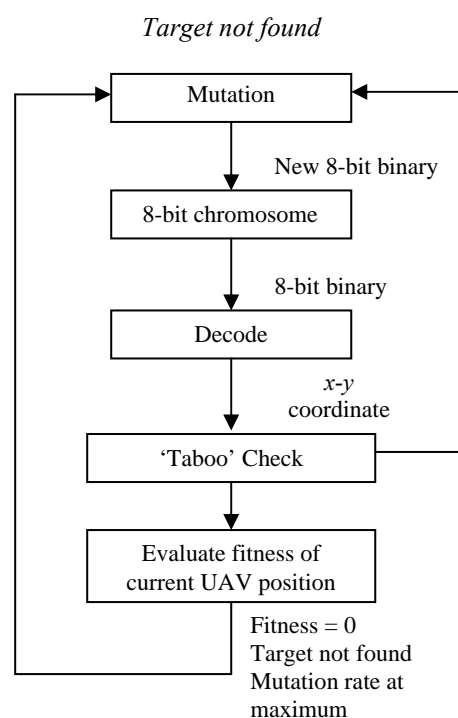


Fig. 9. The mechanics of the evolutionary algorithm.

However, if a target has been found ( $Fitness > 0$ ), the mechanics of the search algorithm are changed. The search coordinates are now generated by an adaptive search algorithm, as illustrated in Fig. 10.

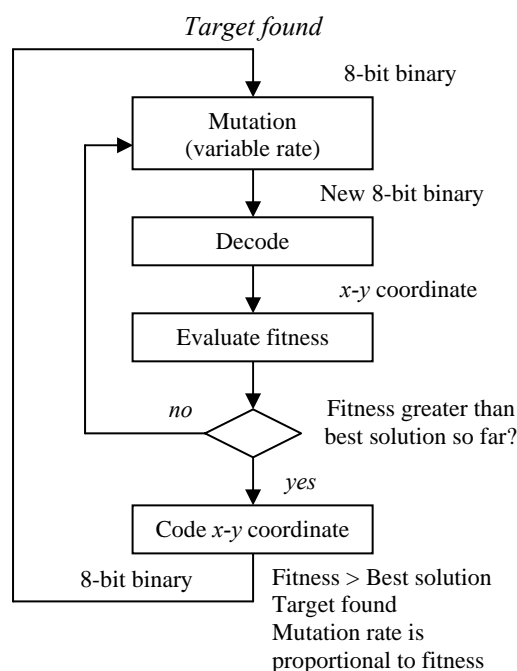


Fig. 10. The mechanics of the adaptive-evolutionary algorithm featuring variable mutation rate.

This algorithm is based on a “best guess so far” principle where the instant that a UAV first registers a fitness value greater than 0, the  $x$ - $y$  coordinate of the UAV is recorded with its associated fitness. The  $x$ - $y$  coordinate is coded into the closest possible 8-bit binary representation and this chromosome becomes the basis for the next batch of search coordinates. The associated fitness value is used to define the mutation rate used during the subsequent mutation process. As fitness increases, mutation rate decreases, therefore less of the “best guess” chromosome is altered. This has the effect of making successive search coordinates converge towards the target centre, thus allowing the swarm to pin-point and record the exact location of the target.

### 3. MISSION STRATEGY & RESULTS

To illustrate the unification of the low-level and high-level control hierarchies, a swarm comprising four UAVs has been used. The simulated mission is not concerned with group migration between search areas and instead focuses on the Search Mode. The searching process within the search area must also be conducted with the presence of obstacles.

Within the search area, the four UAVs search independently and follow  $x$ - $y$  coordinates initially generated by the evolutionary algorithm. Once a target has been identified however, three UAVs follow the  $x$ - $y$  coordinates generated by the adaptive-evolutionary algorithm in order to pin-point the centre of the target. The fourth UAV continues to search via the evolutionary algorithm. This means

that whilst 3 UAVs pin-point an identified target, at least 1 UAV continues to search the area to find additional targets. These are recorded in a “to do” list which the other 3 UAVs will subsequently pin-point.

The implementation of the described mission strategy is illustrated in Fig. 11a - 11d.

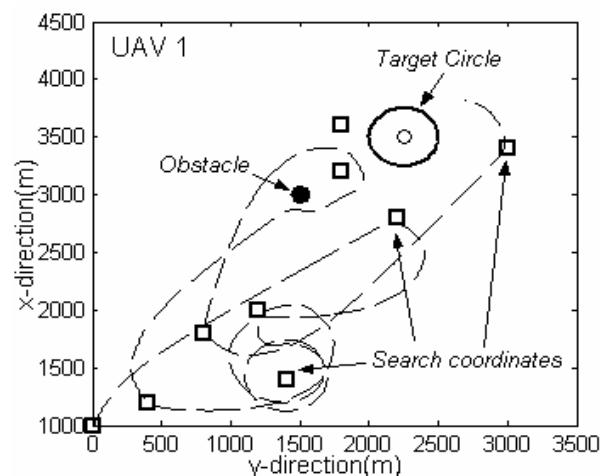


Fig. 11a. The flight path of UAV 1 during the search mission.

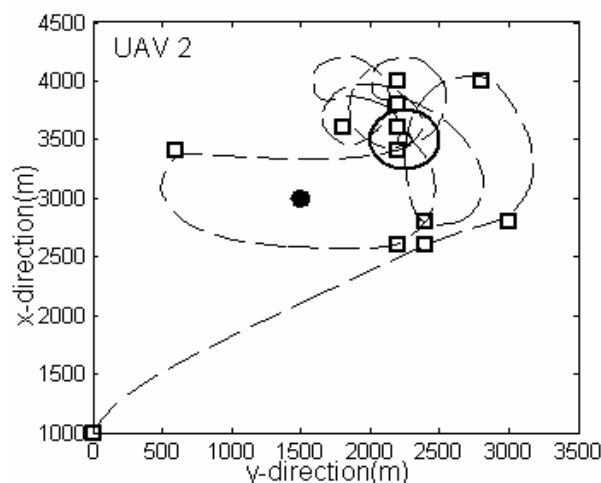


Fig. 11b. The flight path of UAV 2.

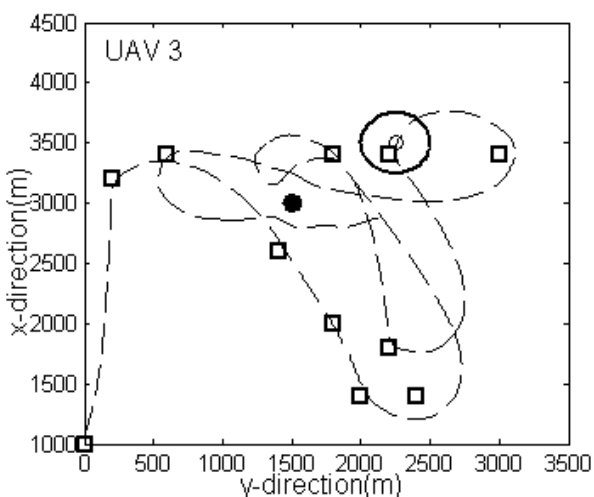


Fig. 11c. The flight path of UAV 3.

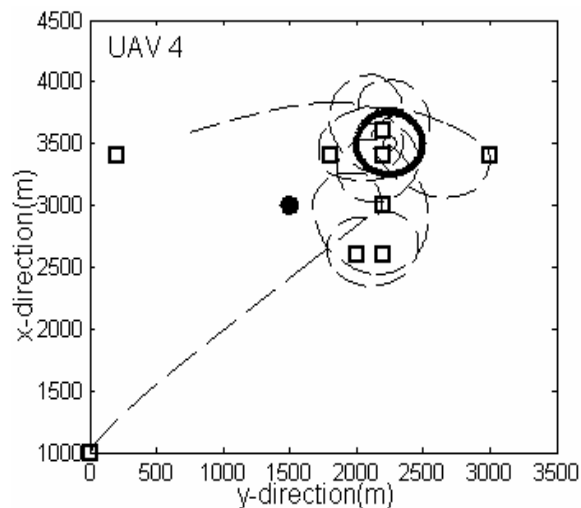


Fig. 11d. The flight path of UAV 4.

The flight path of each vehicle is plotted individually for clarity. The search area contains one target and one obstacle as labelled in Fig. 11a. Each UAV enters from the bottom left-hand corner of the  $x$ - $y$  plot. The flight paths of UAV1 and UAV3 demonstrate the operation of the reactive obstacle avoidance system. Each aircraft successfully avoids the obstacle whilst also intercepting the next waypoint. Fig. 11b is a particularly good example of successive search coordinates converging towards the target.

#### 4. CONCLUSIONS

This paper presents a system to control multiple UAVs as they perform a blind search of an area. Simulation results show the successful implementation of evolutionary-based algorithms as the basis of the search method. This search method uses an evolutionary-based algorithm as well as an adaptive version of an evolutionary algorithm in order to guide the swarm of vehicles throughout the search. Through a low-level individual control system, each vehicle is also able to avoid unexpected obstacles. The most important aspect is the continued development of a mission framework that will allow further research into areas such as swarm control methods and collision avoidance.

#### ACKNOWLEDGEMENTS

The authors would like to thank Dr. Stewart S. Houston, University of Glasgow, for the mathematical model of the Puma helicopter and EPSRC for their funding of this research.

#### REFERENCES

Bruemmer, D.J., D.D Dudenhoefter and J. Marble (2002). A Robotic Swarm for Spill Finding and Perimeter Formation. *Proc. of the Int. Conf. on Nuclear and Hazardous Waste Management, Spectrum 2002*, Reno, Nevada, U.S.A., p.CD.  
Carruthers, B. and E.W. McGookin (2004). Swarm Control for Multiple Autonomous Vehicles.

*Proc. of the Int. Conf. on Advanced Engineering Design 2004*, Glasgow, U.K., p.CD.  
Creaser, P.A. and B.A. Stacey (1999). Generation of Dual Missile Strategies Using Genetic Algorithms. *Proc. of the Congress on Evolutionary Computation*, **1**, pp. 117-124.  
Dorf, R.C. and R.H. Bishop (2001). *Modern Control Systems – 9<sup>th</sup> Edition*. Prentice Hall, Inc., New Jersey.  
Enns, D., D. Bugajski and S. Patt (2002). Guidance and Control for Cooperative Search. *Proc. of the American Control Conf.*, **3**, pp. 1923-1929.  
Fierro, R. and A.K. Das (2003). Hybrid Control of Reconfigurable Robot Formations. *American Control Conf.*, **6**, pp. 4607-4612.  
Flint, M., M. Polycarpou and E. Fernandez-Gaucherand (2002). Cooperative Control for Multiple Autonomous UAVs Searching for Targets. *Proc. of the 41<sup>st</sup> IEEE Conf. on Decision and Control*, **3**, pp. 2823-2828.  
Fredslund, J. and M.J. Mataric (2002). Huey, Dewey, Louie, and GUI – Commanding Robot Formations. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, **1**, pp. 175-180.  
Glover, F. (1989). Tabu Search: Part 1. *ORSA J. of Computing*, **2**, No.1, pp.190-206.  
Goldberg, D.E (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.  
Grevera, G. J. and A. Meystel (1990). Searching in a Multidimensional Space. *Proc. of the 5<sup>th</sup> IEEE Int. Symp. on Intelligent Control*, **2**, pp.700-705.  
Hughes, E.J. (2002). Evolutionary Guidance for Multiple Missiles. *IFAC World Congress Conf.*, Paper 1801, pp. 21-26.  
Jin, Y., A.A. Minai and M.M. Polycarpou (2003). Cooperative Real-Time Search and Task Allocation in UAV Teams. *Proc. of the 42<sup>nd</sup> IEEE Conf. on Decision and Control*, **1**, pp. 7-12.  
Kim, Y., W. Han and T. Kuc (1997). An Intelligent Missile Autopilot Using Genetic Algorithms. *IEEE Int. Conf. on Computational Cybernetics and Simulation*, **2**, pp. 1954-1959.  
Koo, T.J. and S.M. Shahruz (2001). Formation of a Group of Unmanned Aerial Vehicles (UAVs). *Proc. of the American Control Conf.*, **1**, pp. 69-74.  
Marin, J.A., R. Radtke, D. Innis, D.R. Barr and A.C. Shultz (1999). Using a Genetic Algorithm to Develop Rules to Guide Unmanned Aerial Vehicles. *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, **1**, pp. 1055-1060.  
Raphael, B. and I.F.C. Smith (2003). A direct stochastic algorithm for global search, *Applied Mathematics and Computation*, **146**, pp. 729-758.  
Reynolds, C.W. (1987). Flocks, Herds, and Schools: A Distributed Behavioural Model. *Computer Graphics*, **21**(4), pp. 25-34.  
Srinivas, M. and L.M. Patnaik (1994). Genetic Algorithms: A Survey. *Computer*, **27** (6), pp. 17-26.