Towards Autonomous Patrol Behaviours for UAVs

John Oyekan and Huosheng Hu,
School of Computer Science and Electronic Engineering,
University of Essex, Wivenhoe Park, Colchester CO3 4SQ, United Kingdom.
jooyek@essex.ac.uk; hhu@essex.ac.uk

***Abstract*—Most Unmanned Aerial Vehicles (UAVs) require a human in the loop during the decision process of carrying out duties such as surveillance, reconnaissance, search and rescue and security patrol missions. This human link introduces a weakness into the entire system since humans are easy to get tired, bored or lose concentration after prolong periods of operation. This weakness is further compounded by the type of UAV platform used. In this paper, we address this problem by implementing various behaviours on Security Surveillance UAVs. These behaviours would enable UAVs to carry out security patrol missions autonomously. The behaviours developed were tracking, hovering, landing, trajectory following, waypoint, and obstacle avoidance behaviours. The commercially available DraganFlyer QuadRotor was used as the UAV platform in this investigation. Some experimental results are presented to demonstrate the feasibility and performance of the proposed system.**

***Index Terms*—UAV, Autonomous behaviours, Security patrol, DraganFlyer.**

I. INTRODUCTION

Within the last decades, the use of Unmanned Aerial Vehicles (UAVs) in military operations has increased tremendously, including surveillance, reconnaissance and even search and destroy missions because of the significant advantages over manned aerial vehicles. For example, small size of an UAV enables it to penetrate an enemy's radar defences while maintaining a low radar signature. Its low noise level gives it a low profile when stalking or collecting status data of an object of interest. This prevents it from being detected easily and hence shot down. An UAV also carries more payload than a manned aerial vehicle since the space that is occupied by a human pilot and life support systems can be filled up with equipment. In addition to these advantages, if an UAV were shot down, no human life would be lost as the operator would not be on-board.

Furthermore, a team of Unmanned Aerial Vehicle could be equipped with co-operative algorithms that would enable them to form a dynamic communication network and reconfigures itself based on needs or circumstances leading to constant communication coverage of a battle space. This would eliminate the need to reconfigure an existing satellite's mission to power itself over the desired area hence saving its precious limited fuel supply [1][2][3]. However, most of UAVs require a human operator to remotely control it [4]. This re-introduces the weakness of the human factor into the control loop as humans may lose concentration, get tired and bored after extended periods of time leading to mistakes. Furthermore, the probability of making mistakes increases with an increase with the degree of complexity required to fly the UAV. This was made more obvious with the amount of effort required to manually fly the platform used in this project. This problem was addressed in our investigation by introducing intelligent (capable or adequate for the task at hand) behaviours into the UAV platform to assist the UAV operator.

Our results suggest that an UAV can be used to effectively perform autonomous security surveillance patrol tasks repeatedly after it has been programmed. By using waypoint behaviours, a series of defined waypoints can be patrolled every hour. A tracking behaviour could be used to stalk and investigate a moving object of interest. If the object becomes stationary, a hovering behaviour can be used to maintain a fixed position for continuous observation. Obstacle avoidance behaviour could be used to avoid obstacles in the flight path of the UAV while the trajectory behaviour could be used to maintain the flight path or trajectory precisely. By combining all these behaviours together, an UAV capable of constant autonomous patrol of waypoints can be achieved. In this paper, we present the results obtained during the development of the behaviours mentioned above. In addition, a landing behaviour was also developed for the landing phase of an UAV.

The rest of the paper is organised as follows. Section II reviews some related research work on unmanned aerial vehicles done by other researchers. Section III introduces the DraganFlyer Quadrotor platform used in this research, as well as its PID control algorithm. The Behaviour development is discussed in Section IV while results are presented in Section V. Finally, a brief conclusion and future extension are outlined in Section VI.

II. RELATED WORK

There are many research projects that have carried out investigations into controlling Un-manned Aerial Vehicles autonomously. The Multiple Agent Intelligent Coordination and Control (MAGICC) lab at the Brigham Young University used fixed-wing Unmanned Aerial Vehicles during their outdoor investigation of autonomous tracking of waypoints [5]. They used the force field vector method to fly UAVs to a desired waypoint. These fixed-winged UAVs were also launched by hand. The OATs project of the University of Oxford implemented autonomous visual tracking of intelligent ground targets and way points using a commercial airframe [6].

The Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) project [7] uses a QuadRotor platform to investigate autonomous multi-agent control of numerous QuadRotors in Real-world scenarios. The platform was developed so that it carried sufficient computing power in order to carry out computations in real-time. The UAVs in this project followed a trajectory autonomously in order to reach a desired way-point [8]. The Vanderbilt Embedded Computing Platform for Autonomous Vehicle (VECPAV) project at Vanderbilt University aimed to develop an autonomous intelligent control system that replaces a human operator in the flight control process of an UAV [9].

In contrast, the Raven project of the Massachusetts Institute of Technology, investigates the development of a test bed for the rapid prototyping of UAV technologies. It uses a multi-vehicle platform comprising of autonomous ground and aerial vehicles. It aimed to develop mission-level algorithms [10]. This project made use of a motion-capture system during its investigations. The Bear Aerobot team of the Berkeley University use a rotary winged platform to investigate obstacle avoidance and trajectory path planning in an urban environment [11].

The Carnegie Mellon Robotics Institute used a rotary winged helicopter to investigate autonomous flight and have achieved autonomous takeoff, trajectory following and landing [12]. The Georgia Institute of Technology have developed a visual target designation and tracking system algorithm for an Autonomous Helicopter platform [13]. The WITAS Laboratory at the Linkoping University aims to develop algorithms that would enable an UAV to be autonomously deployed to monitor traffic networks. This would enable the UAV to navigate in such environment and identify vehicles and their behaviours and react accordingly [14].

However, according to present knowledge, none of the above mentioned projects or researchers has ever used a behaviour-based approach to develop and combine various behaviours together on an UAV platform to achieve autonomous patrol of waypoints.

III. PLATFORM AND ITS CONTROLLER

A. *UAV platform:*

The platform used in during this investigation was a DraganFlyer Quadrotor as shown in the Fig. 1. This platform was chosen because it required an electric supply for its power requirements and not fossil fuel. This made it possible to conduct experiments in a space constrained indoor facility- University of Essex Robotics Arena- in all outdoor weather conditions without any health risk. In addition, it is easy to maintain and repair-things that are difficult to do on other types of rotary platforms that use mechanical linkages.

Fig. 1. The DraganFlyer [15].

However, controlling this platform remotely is not a trivial task as the human operator has to keep making minute adjustments to make the platform hover in a location precisely and follow a direct path leading to tiredness and sore thumbs [16][17]. As result, we were not able to fly this platform manually during our experiments. Nonetheless, by using a behaviour based approach and a *PID* low level controller under computer control we were able to achieve hovering at a precise location and also direct motion. Our approach also enabled the user to fly the platform using a simple human-machine interface in the form of a Graphical User Interface. The Graphical User Interface enabled the user to choose the behaviours needed at the moment while the computer took care of the flying of the UAV. Our control algorithm converted the user requests into a series of lower level commands using a designed *PID* controller Module to fly the DraganFlyer. A *PID* control scheme was chosen because of its robustness as observed in [18][19]. Even though [20] observed a disappointment with it due to strong perturbations, this was not observed in this investigation.

B. The PID Controller Module:

The *PID* controller Module was made up of five *PID* controllers- Height/Altitude *PID* controller, Yaw *PID* controller, Pitch *PID* controller, Roll *PID* controller and Position maintain *PID* controller.

The Yaw, Pitch and Roll *PID* controllers shown in the Fig. 2 were used to decouple the highly coupled dynamics of the DraganFlyer in order to control the yaw, pitch and roll angles respectively. The Altitude and position maintain *PID* controllers shown in Fig. 3 and Fig. 4 respectively were used to achieve autonomous hovering of the DraganFlyer at a precise location and height.

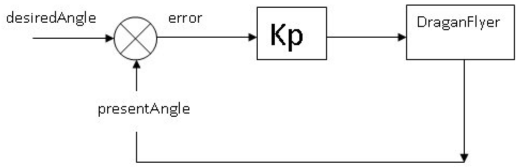

Fig. 2. Yaw, Roll and Pitch PID controller structure (*kp* [roll] = 3.0), (*kp* [pitch] = 2.0), (*kp* [Yaw] = 2.0).

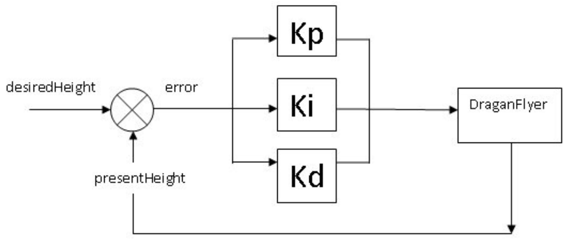

Fig. 3. Altitude PID controller structure (*kp*, *ki*, *kd*) = (0.0323, 0.00025, 2.5).

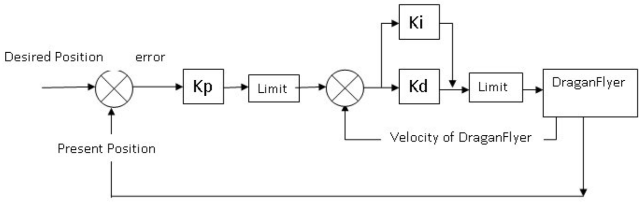

Fig. 4. Position maintain PID controller structure (*kp*, *ki*, *kd*) = (1.25, 2 $*$10e-6, 0.005).

Position, altitude and attitude information were obtained by using a VICON motion capture system operating at a frequency of 50Hz and connected to a Dell Precision PWS490 dual core computer with an Intel Xeon CPU 5150 running at a speed of 2.66GHz. It has a 2 Gigabytes of RAM. The control algorithms shown in Algorithm. 1 were also ran on this computer.

Algorithm 1 Pseudocode for the Position Maintain PID controller.

```
while not stop do
    calculate velocity
    calculate error in position
    calculate integral error
    calculate proportional error
    if proportional error >MAX DISTANCE then
        proportional error = MAX DISTANCE
    end if
    calculate velocity needed
    calculate desired Attitude angle
    if desired Attitude angle >MAX ANGLE then
        desired Attitude angle = MAX ANGLE
    end if
    old position = present position
    vicon delay
end while
```

IV. BEHAVIOUR DEVELOPMENT

The hovering behaviour is the fundamental behaviour of our system. With it, it was possible to maintain a location precisely. The hovering behaviour was achieved by using the *PID* controller structure shown in Fig. 3 to maintain the altitude of the DraganFlyer and the *PID* controller structure shown in Fig. 4 to maintain the position of the DraganFlyer.

A. Landing Behaviour:

The landing behaviour was needed for autonomous landing after completing a mission. A different set of *PID* parameter values were used for the *PID* control loop during the landing phase. The *P* parameter in the landing *PID* Controller was reduced from a value of 0.0323 to a value of 0.00923. This was done to reduce the overshoot caused by the *P* parameter when close to the ground. When the *P* parameter was too large, it resulted in the DraganFlyer crashing into the ground resulting in damage.

B. Waypoint Behaviour:

This behaviour was needed to move from one waypoint to another. In developing this behaviour, limits were introduced so that if the desired distance was above a certain maximum value, the maximum value was passed to the position maintain *PID* Controller. This was to prevent the overloading of the position maintain *PID* Controller. It also served to reduce over shoots at the end waypoint.

C. Trajectory Behaviour:

The Trajectory Behaviour was needed to follow a trajectory precisely. This behaviour was developed by using minute waypoints to reach the final waypoint. It was done with the aim that it might improve performance by reducing the overshoots produced by the *PID* positional maintain Controller during control in the x and y direction.

D. Obstacle Avoidance Behaviour:

In order to avoid collision with an obstacle, the obstacle avoiding behaviour was programmed to aim at a position of 1000 mm away from the obstacle. Aiming at a distance of 1000 mm away from the obstacle enabled the position maintain PID Controller recover from any overshoots (if there were any) so that the DraganFlyer would be able to fly without hitting the obstacle. This specification was modified from [17]. In implementing the final overall obstacle avoidance behaviour, triangle fuzzy logic membership functions as shown in Fig. 5 were used to combine the outputs of the trajectory following behaviour and the obstacle avoidance behaviour together as is shown in Algorithm. 2.

E. Tracking Behaviour:

For the tracking behaviour, the red RC remote control car was used as the chase object. A model for the car was built in the VICON MX software. The coordinates of the car were then obtained from the VICON system, and then introduced to the position maintain *PID* Controller. This made the DraganFlyer followed the car.

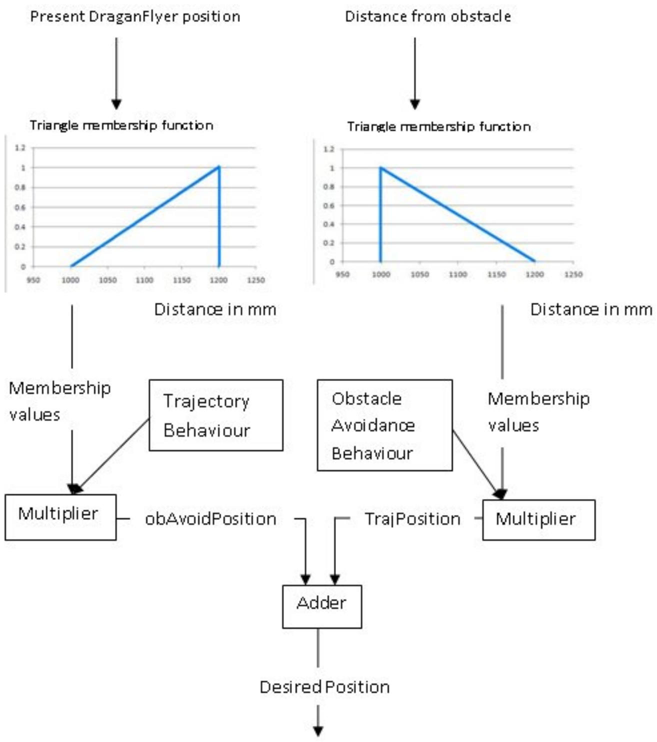

Fig. 5. Obstacle Avoidance behaviour implementation.

Algorithm 2 Fuzzy Logic Algorithm for Obstacle Avoidance behaviour implementation.

```
while not stop do
    obtain Trajectory Behaviour output
    obtain Obstacle Avoidance Behaviour output
    pass outputs to Triangle fuzzy membership functions
    obtain fuzzy membership values
    modify Behaviour outputs based on fuzzy membership values
    add both modified Behaviour outputs together.
    pass values to the position maintain PID controller alogrithm.
end while
```

V. EXPERIMENTAL RESULTS

A. Hovering Behaviour:

By using *PID* controller structures in Fig. 3 and Fig. 4, it was possible to achieve the results shown in Fig. 6, Fig. 7 and Fig. 8 when the DraganFlyer was commanded to hover at (x, y, z) = (-500, 0, 1020) mm for approximately 100 seconds. These results show that the Draganflyer was able to maintain the y position to within 300 mm of the requested *y* position and 700 mm of the requested *x* position. The offset observed in the results could be eliminated by tuning the *PID* controller structure further or adding appropriate *P*, *I*, *D* elements to it. However, these results are comparable with the results obtained in [10], [20] and [21].

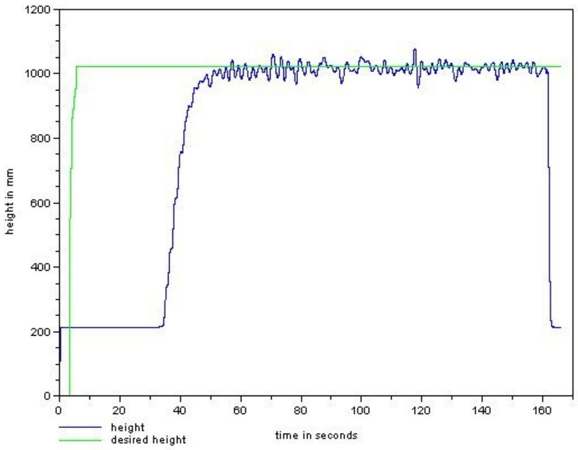

Fig. 6. Graph showing performance of Height PID Controller.

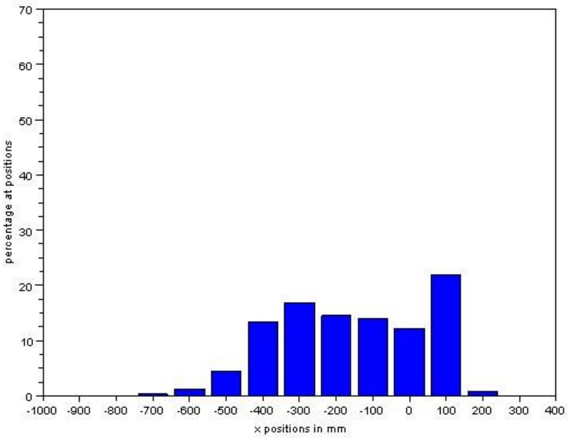

Fig. 7. Graph showing percentage of time the DraganFlyer maintained x position.

B. Landing Behaviour:

Fig. 9 shows the output of the landing behaviour. During the landing phase, landing was performed in stages- from 1020 mm to 320 mm. This was to make sure that the DraganFlyer landed in a controlled fashion. When the DraganFlyer was within 600 mm of the *x* and *y* home position and at a height of 320 mm, the rotors were put into a reduced throttle value so that the thrust generated by the rotors were slightly below the weight of the DraganFlyer. This made the DraganFlyer reduce altitude from 320 mm to ground level in a gracefully manner and also within 600 mm of the home position.

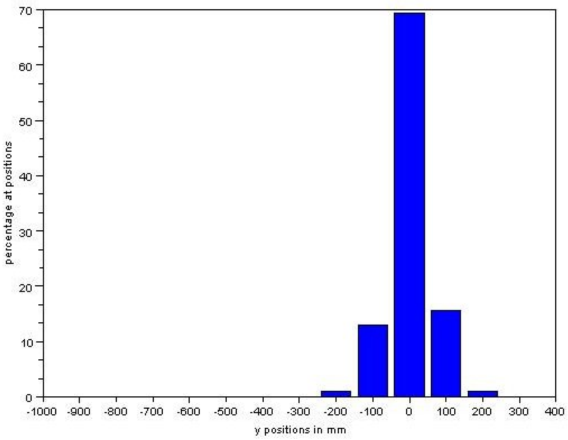

Fig. 8. Graph showing percentage of time the DraganFlyer maintained y position.

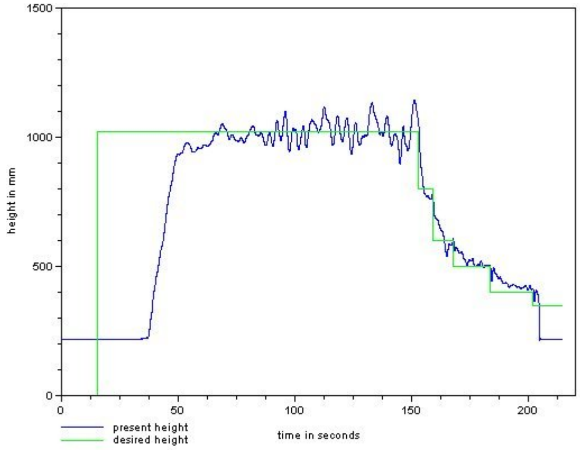

Fig. 9. Graph showing performance of Landing Behaviour.

C. Waypoint Behaviour:

An investigation into this behaviour was carried out by commanding the DraganFlyer to fly from the home waypoint (*x*, *y*, *z*) = (500, 1600, 1020) mm to a desired waypoint (*x*, *y*, *z*) = (-500,-1600, 1020) mm and back. This was maneuver was performed three times during a run to gain a proper understanding of the behaviour. The results are shown in Fig. 10.

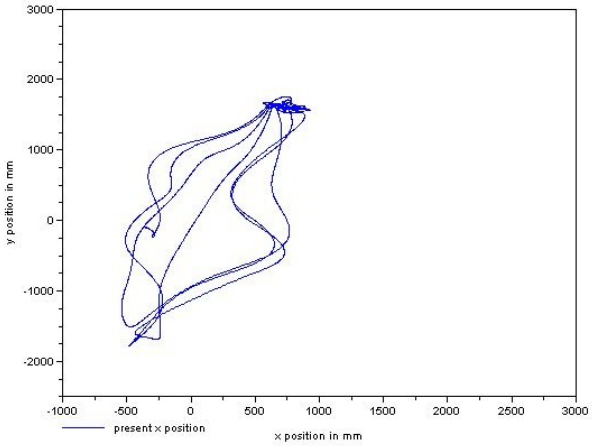

Fig. 10. Graph showing performance of Waypoint Behaviour.

D. *Trajectory Behaviour:*

The Trajectory Behaviour was investigated by simulating a straight path going from point (x, y) = (500, 1600) mm to (-500, -1600) mm. This was represented by a straight-line equation of $y = 0.312x$. By incrementing a variable every 5 seconds (This time was chosen to give the DraganFlyer time to settle at a minute waypoint before moving to the next minute waypoint) by 100 mm, the x positions needed to get the y position were generated. This made the DraganFlyer move in discrete steps towards a waypoint. The results are shown in Fig. 11. During this experiment, the DraganFlyer was commanded to go to the desired waypoint (x, y) = (-500, -1600) mm and then back to the home waypoint (x, y) = (500, 1600) mm.

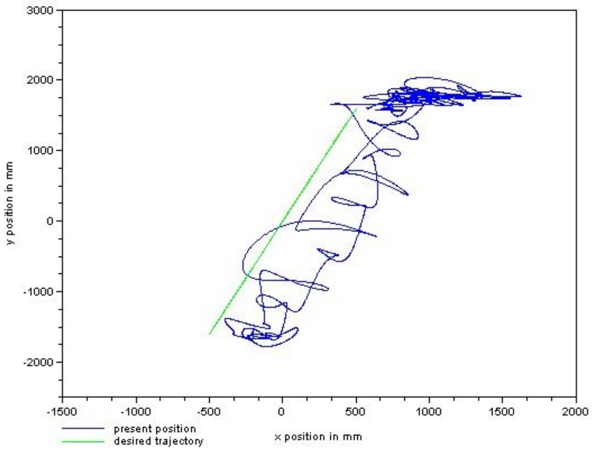

Fig. 11. Graph showing performance of Trajectory Behaviour.

E. *Obstacle Avoidance Behaviour:*

By using fuzzy logic, the outputs for these two behaviours were combined to produce a smooth output as shown in Fig. 12. An obstacle was simulated by using coordinates (x, y) = (0, 0) mm. The DraganFlyer was commanded to go from the home waypoint (x, y, z) = (500, 1600, 0) mm to a destination waypoint of (x, y, z) = (-500, -1600, 1020) mm and then back home. This was repeated three times to prove that behaviour output was effective in avoiding the obstacle.

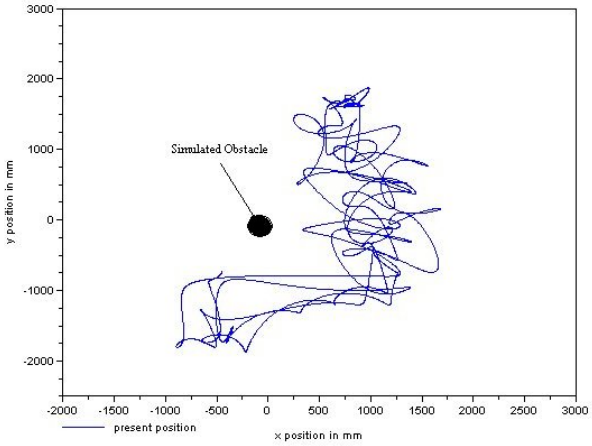

Fig. 12. Graph showing performance of Obstacle Avoidance Behaviour.

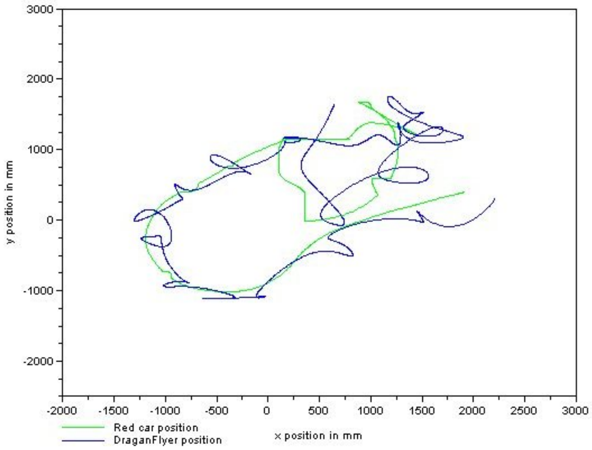

Fig. 13. Graph showing performance of Tracking Behaviour.

F. *Tracking Behaviour:*

The result of this behaviour can be seen in Fig. 13. From the result, it can be seen that the DraganFlyer was able to track the Red car.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have shown the results of using behaviour based approach in controlling a platform that is difficult to fly manually. We have also been able to show that our control approach makes it easy for a novice user to fly the DraganFlyer. The problem of stabilizing the DraganFlyer during flight was addressed by using a Proportional Integral and Derivative controller. Our investigations reveal that the use of a behaviour-based approach comprising of the various behaviours mentioned above would lead to a Military Security Surveillance Unmanned Aerial Vehicle capable of Autonomous Patrol of designated waypoints.

Our future work would be to fuse the output of the behaviours using a fuzzy logic controller to accomplish a fully autonomous patrol of waypoints. Work has already begun on this using a state machine, i.e. subsumption architecture, to achieve more autonomy on the part of the UAV and less work for the user of the system. We also plan to extend the work to control a swarm of both UAVs and UGVs for autonomous multiple target tracking, continuous surveillance and coverage of an area.

ACKNOWLEDGEMENTS

We would like to thank Renzo De Nardi for all his technical help with the VICON motion capture system.

REFERENCES

[1] Northrop Grumman, *North Grumman demonstrates Expanded Capabilites of system that delivers video imagery to Military Forces in Urban Combat*, **http://www.irconnect.com/noc/press/pages/news_releases.mhtml?d =106913** (06 March 2008).

[2] Northrop Grumman (2006), *Northrop Grumman to use a Fire scout Unmanned Aerial Vehicle to Enhance Network- Centric Capabilities for the warfighter*, **http://www.irconnect.com/noc/press/pages/news_releases.mhtml?d =96366** (06 March 2008).

[3] H. Hu, *Co-operation of Multiple Mobile Robots*,**http://dces.essex.ac.uk/staff/hhu/research.html#Topic04**(14 August 2008).

[4] CNN.com (11 March 2004), *Flying without a pilot*, **http://edition.cnn.com/2004/WORLD/asiapcf/03/11/unmanned.aerial. vehicles/index.html**(07 March 2008).

[5] D.R. Nelson, D.B. Barder, T.W. McLain, and R.W. Beard, *Vector field path following for small unmanned air vehicle*, in Proc.2006 American Control Conference, Minneapolis, MN, 2006, pp.5788-5794.

[6] H. Helble and S. Cameron, *OATS: Oxford Aerial Tracking System*, Oxford University Computing Laboratory.

[7] STARMAC, *stanford Test bed of Autonomous Rotorcraft for Mult-Agent Control*, **http://hybrid.stanford.edu/starmac/overview** (06 March 2008).

[8] G. Hoffmann, D.G. Rajnarayan, S.L. Waslander, D. Dostal, J.S. Jang and C.J. Tomlin, *The Stanford Testbed of Autonomous Rotorcraft for MultiAgent Control (STARMAC)*, Stanford University, Stanford, CA.

[9] RCHelicopter.com, *DraganFlyer RC Helicopters in VECPAV Autonomous Control System At Vanderbilt University*, NV, US, **http://www.rchelicopter.com/2007/11/22/draganflyer-uav-vanderbilt-university-drone/** (06 March 2008).

[10] J.P. How, B. Bethke, A. Frank, D. Dale, and J. Vian (April 2008) , *Real-Time Indoor Autonomous Vehicle test Environment*,IEEE Control Systems Magazine.

[11] D.H. Shim, H. Chung, S. Sastry and H.J. Kim (2005), *Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles*, American Institute of Aeronautics and Astronautics.

[12] O. Amidi, T. Kanade and R. Miller, Carnegie Mellon Robotics Institute (1998), *Vision-base autonomous helicopter research at Carnegie mellon robotics institute 1991-1997*, American Helicopter Society International Conference, Heli, Japan.

[13] D.E. Cardoze and R.C. Arkin, *Development of Visual Tracking Algorithms for an Autonomous Helicopter*, Mobile Robot laboratory, college of Computing, Georgia Institute of Technology, Atlanta, Georgia, 30332.

[14] P. Doherty, G. Granlund, K. Kuchcinske, E. Sandewall, K. Nordberg, E. Skarman and J. Wiklund, *The WITAS Unmanned Aerial Vehicle Project*, ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence, IOS Press, Amsterdam, 2000.

[15] The DraganFlyer, Remote Control Helicopter, **http://www.otherlandtoys.co.uk/dragfaq.htm** (19 March 08).

[16] N. Sacco, *How the Draganflyerflies*, Rotary Magazine, **http://www.rctoys.com/pdf/draganflyer3rotorymagazine.pdf**

[17] P. McKerrow (2004), *Modelling the DraganFlyer Four-rotor helicopter*, Proceed-ings of the IEEE International Conference on Robotics and Automation,26 April-1 May 2004, 4, 3596-3601. Copy right IEEE 2004.

[18] S. Bouabdallah (2007), *design and control of quadrotors with application to autonomous flying*, Lausanne, EPFL.

[19] D. Gurdan, J. Stumpf, M. Achtelik, K. Doth, G. Hirzinger and D. Rus (10-14 April 2007),*Energy efficient Autonomous Four rotor Flying Robot Controlled at 1khz*, IEEE International Conference on Robotics and Automation, Roma Italy.

[20] S. Bouabdallah, A. Noth and R. Siegwart, *PID vs LQ Control techniques Applied to an Indoor Micro Quadrotor*, Autonomous Systems Laboratory Swiss Federal Institute of Technology.

[21] P. Castillo, A. Dzul and R. Lozano (July 2004), *Real-Time Stabilization and Tracking Four-Rotor Mini Rotorcraft*, IEEE Transactions on Control Systems Technology, vol 12, No 4.Transactions on Control Systems Technology, vol 12, No 4.