RESEARCH ARTICLE OPEN ACCESS

UAV Swarm Co-Ordination and Control Using Grossberg Neural Network

Dabbiru Ravi Vikranth
Department of Computer Science,
B. R. R. Degree College, Visakhapatnam

ABSTRACT

A UAV swarm (Swarm) has multiple unmanned aerial vehicles (UAV) working in synchronization towards achieving a unified/ universal task. Swarm Co-ordination, primarily deals with communicating with each individual UAV independently from the Ground Control Station (GCS) and negotiation based [8] information exchange between UAVs themselves. The most common type of flight formations are behavioural formations. Leader – follower formation types are the most closely associated with hierarchical class schemes. In the Leader – Follower formation, it is enough to specify a path for the leader to fly as in order to dictate the path taken by formation. The architecture of a system supporting a hierarchical Leader/follower formation scheme for Swarm of Quad copters. One Quad copter acts as a leader, while other acts as follower/ (s).

***Keywords:* —** UAV, RNPG Algorithm, Navigation Point, Model Predictive Control, Formation

I. SWARM CO-ORDINATION

Swarm members work together by communicating their position and other useful information in pre-defined intervals. In order to enable such co-ordination, UAV swarm members need to communicate with each other. In case of dedicated communication infrastructure; swarm itself establishes and maintains an ad-hoc communication network. Communication in Flying Ad-hoc Network (FANET) is focused in UAV – UAV (U2U) and UAV to Infrastructure (U2I) communications. Thus, the degree of mobility of nodes is greater than Mobile Ad-hoc Network (MANET) and Vehicular Ad-hoc Network (VANET). The topology changes frequently (it needs peer to peer networking). The communication range must be greater than other networks.

.

Ad-hoc On Demand Distance Vector (AODV):-

AODV is a reactive protocol, which has same on-demand characteristics like Dynamic Source Routing (DSR) while maintaining different mechanisms of routing table. In AODV, each node stores a routing table, which contains a single record for each destination, while in DSR each node can store multiple entries in its routing table for each destination. In AODV, the source node (and also other relay nodes) stores the next hop information corresponding to each data transmission. AODV routing protocol consists of three phases viz. route discovery, packet transmitting and route maintaining. If the source node has packets to send, it initiates a route discovery process to locate the destination node and then dispatches these packets over a determined route. Discovery process enables determined routes without a loop, and it uses a sequence number to determine an up-to-date route of the destination. An expiration time is used to keep route freshness. In this process, intermediate nodes also update their routing tables. After a route-id is constructed, packets are transferred over it. As a result of mobile nodes, some link failures may occur, and this connection loss triggers a repairing process to maintain the routes.

Challenges of FANET:

Integrating UAVs into national air space is the need of the hours. This co-ordination will enable the destruction of enemy aircraft with minimal losses. At the same time, these UAVs can be used as electronic jammers and for real time video reconnaissance in enemy areas. Therefore, the collaboration of UAVs and manned air craft should be in networked environment.

A FANET uses various wireless communication bands such as VHF, UHF, L-Band, Ku-Band, C-Band etc. These bands are also used in application areas like GSM Networks, Satellite Communications etc. To reduce the frequency congestion related issues, there is a need to standardize these communication bands, signal modulation and multiplexing models.

Control of Swarm

The term Collision avoidance (CA) [2] represents the scenario where the Quad copter; try to avoid obstacles. Navigation problems of UAVs flying in formation in a free and obstacle laden environments are investigated in this paper. When static obstacles pop-up during the flight, the UAVs are required to turn around them and also avoid collisions between them. In order to achieve these goals, a new dual mode control strategy is proposed i.e. a safe mode is defined as an operation in an obstacle free environment and a danger mode is activated when there is a chance of collision or when there are obstacles in the path.

Safe Mode achieves global optimization because the dynamics of all the UAVs participating in the formation are taken into account in the control formulation. In the danger mode, a novel algorithm using a modified Grossberg Neural Network (m-GNN) [10] is proposed for obstacle/ collision avoidance. The two dimensional decentralized algorithm uses the geometry of flight space to generate optimal trajectories. In order to handle practical vehicle constraints, the lower layer of the architecture uses a Model Predictive Control (MPC) based tracking controller which tracks references generated by the upper layer.

The two dimensional collision avoidance environment investigated in [4] was used to find dynamically feasible Collision free paths. After obtaining an obstacle free path, a reactive path planner was used to avoid pop up obstacles. This generates trajectories that are not taxing for the vehicle i.e. trajectories involve less turns i.e. main idea is to track a earlier generated/ registered path and avoid pop-up obstacles as they appear, this scheme does not guarantee shortest path always.

A Schematic of the control architecture is presented in Figure – 1. Mode Selection is based on the existence of threat/ collision possibility. In both modes, the upper layer is used to generate reference trajectories and the lower layer uses an MPC based tracking controller to make the UAV follow the reference generated by the Upper layer. Compared to the most commonly used leader follower formation method [3], where the leader considers only its own path and no co-operation feedback exists [10] in safe mode, the upper layer controller uses the relative kinematics between UAVs to generate trajectories that result in optimal scheme for the entire formation.

The Upper Layer in danger mode uses a modified version of Grossberg Neural Network (m-GNN) to get the shortest distance between UAVs current position and a target outside of the danger zone that can be chosen basing on the current sensor fusion and the mission objectives. In [5], the m-GNN is used to find a Collision free path for a single robot. The entire workspace is divided into grids. One neuron is kept at each node of the grid. Each neuron state is decided by the excitation of the neighbouring neurons. All neurons achieve a stable steady state and the shortest distance path is obtained by moving along a path showing increasing activity value on neurons. The method in [6] generates the global optimal path; however the computation time is large because a neuron is place at every node point in the grid. The total number of neurons will be n_x x n_y, where n_x and n_y are the number of grid points in the x and y directions respectively. The method presented in this paper places neurons only at the vertices of the obstacles and the goal, which means that only 4n+1 neurons are required (n is the number of obstacles). 4n neurons are due to 4 vertices per obstacle and 1 neuron comes from the goal (target).

MPC is attractive for the UAV application since it changes a tracking problem into a parameter optimization problem. It can deal with changes in reference values during the course of operation and handle state and control constraints easily.

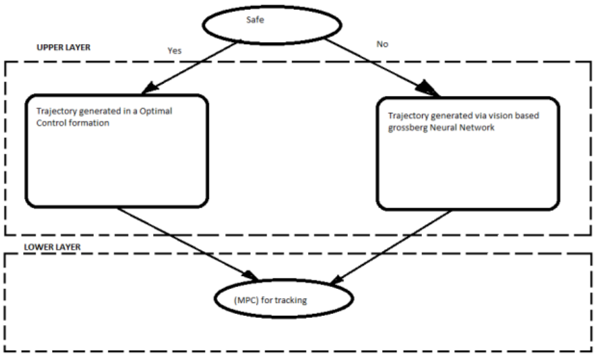


Figure – 1: TWO LAYER HIERARCHIAL STRUCTURE

II. HIERARCHICAL CONTROL DESIGN

Safe Mode:
The control commands are generated in a centralized manner. However, this process can be decentralized to a co-operative scheme with the information at a UAV limited to neighbouring or selected member of the entire UAV fleet. Relative distances and relative velocities are the upper layer system states and the relative forces are the controls. For the problems considered in this paper, the safe mode will give an optional scheme to drive the states of the relative system to the desired ones given by [7]. The Safe mode structure is given in the figure;

Upper Layer (Reference Trajectory Generation):

The RNPG Algorithm (RNPGA) [1] defines the waypoints to be developed in the given workspace. Implementing in our experimental setup involves specifying the GPS co-ordinates at all nodes of the perimeter nodes. The RNPGA is re-worked for a safe distance of 1 metre and average velocity of Quad Copter @ 22 m/s

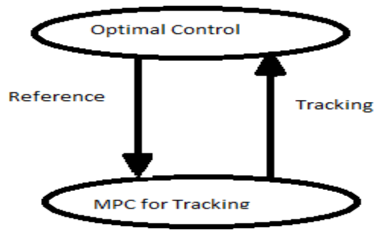


Figure 2 : Control Hierarchy in Safe Mode

Thereby giving a leverage of 100 mS for each cell using the algorithm. The following control law equations are used. The optimized path from the various way points generated is computed using Ant Colony Optimization Algorithm (ACO

Algorithm). The Pheromone level, evaporation rate are as follows;

The local minima of shortest distance (least two) are ranked as path-1 and path-2. The state representation in Earth Centered Earth Fixed (ECEF) is [x, y, z, roll, pitch, yaw, and bearing]. The transformation from one way point to another is depicted by the Direct Cosine Transformations (DCT) represented as;

δt time taken to travel from one way point to another in any given path is calculated; inter alia the battery power/ fuel reserve calculations are computed. The total time taken (T_{path}) is computed for the two paths selected. The shortest time traversal path is the optimal path for travel. The optimal path is registered in the Upper Layer (reference trajectory generation).

Lower Layer (Tracking):

After the upper layer trajectories are generated an MPC based controller calculates controls that drives each UAV to form the Square. Consider a general linear system in the discrete form;

$x(k+a) = Ax(k) + Bu(k)$ [1]

$y(k) = Cx(k)$ [2]

where,

x	: system state
u	: system control
y	: system output
A, B	: System matrices
C	: Output Matrix

In safe mode, $A \cong^{\Delta} 1 + A_w dt$; $B \cong^{\Delta} B_w dt$. 'C' is the aggregation matrix (C_{agg}) and in danger mode, A = 1+ 'Adt; B = 'Bdt and C is an identity matrix of appropriate dimension. 'dt' is the sampling interval and 'k' represents the sate or time instant. Control increment can be defined as,

$\Delta u(k) = u(k) - u(k-1)$ [3]

Assuming the current instant as step k, control at (k-1) and sates at k are known. The aim here is to calculate the control increments at k, k+1, . . . , k+hz-1 steps to match system outputs with the desired ones (i.e. reference trajectories). Note that system future output can be predicted as a function of system future control increment based on the system function. Hz is the prediction horizon. Future Hz steps prediction of system output

$Z(k) \cong^{\Delta} [(y(k+a)|k) : (y(k+hz)|k)]$ can be written as

$Z(k) = \Theta\, \Delta u(k) + f$ [4]

$f \cong^{\Delta} \Psi x(k) + \gamma u(k-1)$ [5]

where,

$\Delta u(k) \cong^{\Delta} [\Delta u(k \mid k) \ldots \Delta u(k + hz-1 \mid k)]^T$ [6]

are the future control increments. Note that s(k+T|k) represents the value of the variable s at stage (k+i) given the information upto stage k with the integer i >0

$$\breve{\Psi} = \begin{pmatrix} C_A \\ C_A^2 \\ . \\ C_A^{hz} \end{pmatrix} \qquad \breve{\Psi} = \begin{pmatrix} C_A{}^0{}_B \\ C_{AB} + C_A{}^0{}_B \\ . \\ \sum_{i=0}^{i=hz-1} C_A{}^i{}_B \end{pmatrix} \qquad \Theta = \begin{pmatrix} C_A{}^0{}_B & . & . \\ C_{AB} + C_A{}^0{}_B & C_A{}^0{}_B & \\ - & & \\ \sum_{i=0}^{i=hz-1} C_A{}^i{}_B & . & C_A{}^0{}_B \end{pmatrix}$$

define the objective function at the k^{th} stage as, [7]

$J_i = [Z(k) - W(k)]^T Q_k [Z(k) - W(k)] + \Delta u(k)^T R_k \Delta u(k)$ where Qk >=0 and Rk >0 are weight matrices with the paper dimensions. $W(k) = [\gamma(k+1|k) \ \ldots \ \gamma(k+hz|k)]^T$ are the future references. In safe mode, W(k) is given by the upper layer trajectories generated according to x_r and in danger mode, W(k) are calculated through m-GNN.

Considering the state and control constraints, the objective function can be written as

Min. $J_{\Delta u(k)} = \Delta u(k)^T H \Delta u(k) + G^T \Delta u(k) + C$ [8]

Subject to : $Z_{min.} <= Z <= Z_{max.}$ [9]

$\Delta u_{min.} <= \Delta u <= \Delta u_{max.}$ [10]

Where,

$H \cong^{\Delta} [\Theta^T Q_k \Theta + Rk].G \cong^{\Delta} Z\Theta\, Q_k(f-W(k))$ and

$C \cong^{\Delta} [(f-W(k))^T Q_k (f-W(k))]$ [11]

Zmin. And Zmax. Are the minimum and maximum values of future outputs respectively. $\Delta u_{min.}$ and $\Delta u_{max.}$ are the minimum and maximum values of control increments.

By minimizing the objective function J above; the control increments at k, k+1, … k+hz-1 instants are obtained; thus control sequences are obtained. A point need to be observed that only the first h_u steps of calculated control will be executed where $1 <= h_u < hz$ for all h_u is known as execution horizon (it is assumed that $h_u = 1$). The velocity and acceleration values are computed as mentioned in x(k+1), y(k) above.

III. DANGER MODE

Danger mode operations need to achieve collision avoidance. Presence of an obstacle is assumed to be detected using on board sensor information. It is assumed that the obstacles can be represented by convex polygons. Note that many different real life unsafe regions can be represented with such representations the shortest distance between two points in a two dimensional space and turning only at obstacle vertices i.e. the UAV changes directions only at the vertices of the polygons in space. For safety of UAVs, a safe zone is created around the obstacle and change in direction takes place on vertices of this obstacle. The Upper layer uses m-GNN. Paths can be generated for the UAVs flying in an environment with fixed obstacles and the pop up obstacles (as long as UAVs have enough time to respond to these). The control structure is as shown in figure 3.

A safe zone is created around the obstacles (or threat region) as the UAV cannot turn with infinite acceleration. Creation of a safe zone prevents a UAV from hitting the obstacle [2]. Neurons are placed at the vertices of the safe zone and target

location. In a m-GNN, each neuron receives excitation from its neighboring neuron. A neighboring neuron is defined as the vertex that the UAV can go without hitting an obstacle. Each neuron sees as its neighbor only those vertices where UAV can go without hitting/ colliding with the obstacle and thus generates an obstacle free path. An obstacle free optimal path is generated by moving from one vertex (neuron) to another vertex that has highest activity.
The following are the details of the trajectory generation algorithm;

IV. UPPER LAYER (TRAJECTORY GENERATION)

a. Vertices Creation and visibility graph formation:

Obstacles are represented by rectangles along the x, y axis. Note that the geometric shape does not restrict the applicability of this method. A small safe zone (problem dependant) is created around each article to create a safety margin to accommodate for possible tracking zeros. Four neurons are located at the vertices of the safe zone.

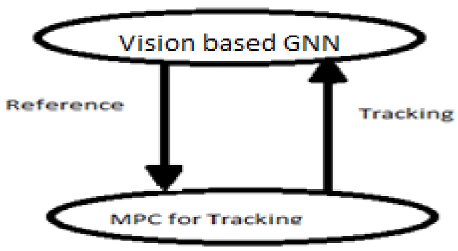


Figure 3 : Control Hieararchy in Danger Mode

A.

A visibility graph is created according to the topology of the environment namely, the geography of the flight space. A visibility graph is defined as map showing the lines joining manually visible vertices. Visibility between two vertices is obtained by joining them by a straight line and checking whether it comes as obstacles.

If this line intersects any of the obstacles (i.e. edges of the obstacle), then the vertices are considered manually invisible. A visibility graph for a simple geometric shape is shown in the figure above. The patches are the left and in the middle are obstacles that have four vertices each.

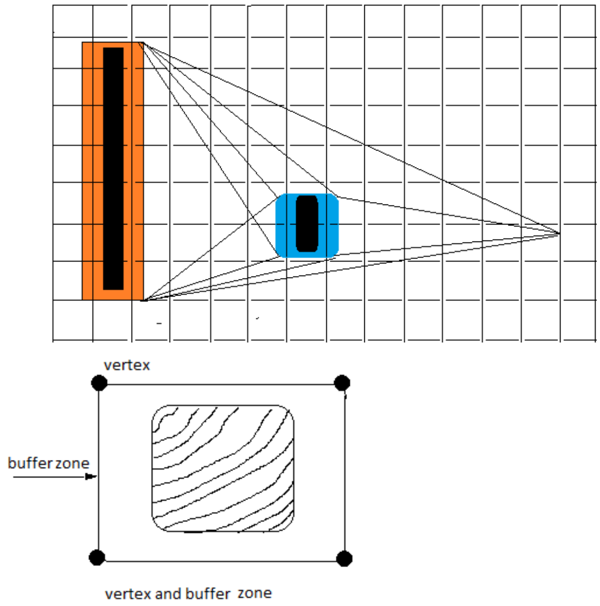


Figure – 4

b. Modified Grossberg Neural Network

Grossberg proposed a model to describe how the human vision system works. He proposed a shunting equation with neurons distributed in space. A modified version of the GNN is developed to adapt to the UAV collision avoidance problem. The dynamics of activities (x_i) of the GNN are given by,

$$\frac{dxi}{dt} = -axi + (b - xi)\left[E + \sum_{j=1}^{k} wijxj\right]$$

$$E = E1 + E2, \quad E1 = \frac{\alpha}{dper}$$

B.

$$E2 = \begin{cases} 100 & \text{if neuron sits on the destination} \\ 0 & \text{otherwise} \end{cases}$$

$W_{ij} = \mu/d_{ij}$

d_{ij} -> distance between i^{th} and j^{th} vertex.

J is the index describing the neighboring neurons, which is defined as the vertex that can be seen from the i^{th} vertex. In other words, a neighbor is a vertex where the UAV can go without colliding with any obstacle, μ is a weighing factor. E is the excitation input to each of the neurons which is comprised of two components i.e. E1 and E2. E1 is the excitation due to closeness of a vertex to the target or a goal point defined by the perpendicular distance (d_{per}) to the straight line joining the UAV and target, α >0 is also the weighing factor the second term E2 is introduced so that the target location gets a high excitation input. A value of 100 is chosen so that the destination node gets a high excitory input.

Each of the neurons activities depends on the activity of its neighboring neurons and excitation it receives. As the excitation for the target is set as a very high value, its activity is the highest, and a high activity propagates to other neurons in the network through the interconnection term

$$\sum_{j=1}^{k} wijxj$$

Therefore, the neurons closer to the target have higher activity values. Also, since E1 and

$$\sum_{j=1}^{k} wijxj$$

Are inversely proportional to d_{per} and d_{ij}, a neuron will receive a high activity if these distances are small. It is desired that the UAV move to a point that is close to the target, however, this should not cause the UAV to take large deviations. Therefore, the targets attraction and the deviation from the straight path should be weighted properly. This is done by adjusting the parameter α and μ. Note that the parameters are adjusted to weigh the shortest distance finding and least stressed path. A high value of α results in an algorithm when a UAV sees an obstacle and moves towards the vertex whose d_{per} is the lowest. However, this move may not result in the overall shortest path. Similarly, a high value of μ causes the activity level of vertices closest to the target to dominate and will cause the UAV to move to the vertex closest to the target. Also a neuron that has lesser number of neurons between itself and destination has higher activity. Hence this implies that the path chosen has less turns. This, however, may not result in the overall shortest path to the target. Any positive values for α and μ will result in an obstacle free path for the UAV. Therefore, the values of α and μ should be adjusted properly to get a path that results in smaller d_{per} values and also chooses the vertices close to the target when such path is the shortest.

The parameter α represents the time decay and can be used to modulate the rate at which steady state is reached. If value of α is high, then the activities decay fast and may not propagate throughout the network. But a high α can cause problems in numerical integration. But a low value of α is also not desirable since it may make the GNN dynamics slow in absorbing changes in the environment. Therefore, the value of α must be large enough to allow activities to propagate throughout the network and at the same time react to changes in the environment.

Once the steady state is achieved, the UAV moves towards the neighbor that exhibits the maximum activity. If the target (or goal point) is visible, the UAV goes to a point that is farthest in the sequence towards the destination among all the waypoints visible. Everytime a new obstacle is detected, new vertices can be added to the existing map and optimum obstacle free path is re-calculated. Note that, a local scheme to avoid obstacle is used when a UAV detects a new obstacle. However, the process does not vive the shortest path but an easier feasible path whereas the technique used in this study computes the shortest or optimal trajectory each time.

Activities of the neurons are generated at every instant, and therefore, the effect of changing operational environment is reflected by the change in the activity of neurons. Time taken for the neurons to attain a steady state can be understood as the reaction time of the network that reflects the change in the environment. As an example, this time was 0.1seconds with 7 obstacles in the environment. The simulations were carried on a 3.2 GHz, 1 GB RAM Dell Desktop. It must be noted that the effect of changes in the environment in the previous stages is not carried forward for the future path planning because the activities are generated every step. Also, since the initial conditions do not affect the steady state value, effect of previous environment configuration is not carried over to later stages. Note that irrelevant obstacles (not in desired path) do not affect the path planning.

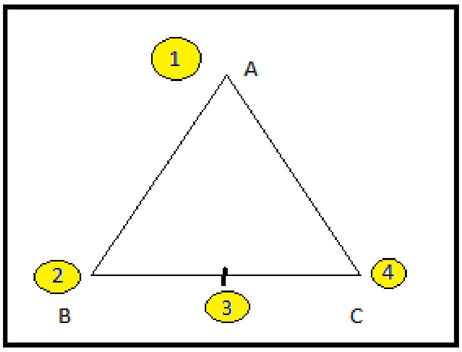
Lower Layer (Tracking)
Control for dynamically feasible trajectory paths are generated in the lower layer with an MPC scheme. Similar, to the safe mode operation.

V. FORMATIONS

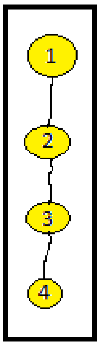
Optimal utilization of resources is the necessity in swarm based missions. Only two no. of formations [9] are considered in this paper viz. Delta Formation (Triangle Shaped) and Line Formation (Straight line).

Delta Formation: This formation is effective when spatial dispersion or more width is available for the acrobatics. This category falls under one of the attack formation. The following is the control laws assumed for the formation;

a. Sum of all Angles in the Formation is 180^0
b. Angular separation of Points B and Point C is 180^0
c. UAV_2, UAV_3, UAV_4 at points Point 2, Point3, Point4 are all equidistant from each other.

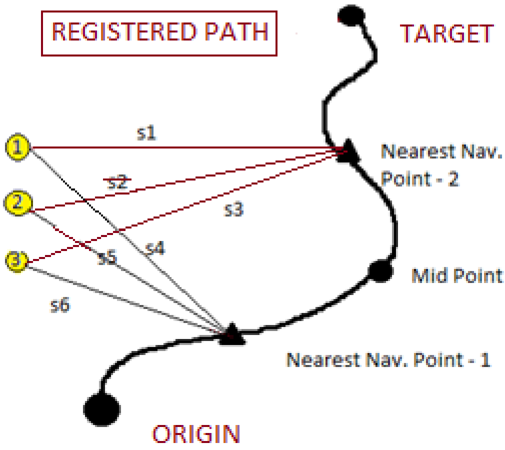


DELTA Formation



Line Formation

Line Formation: In this formation, the follower follows the leader roughly in a straight line. In either of the formations, the minimum distance between any two UAVs is double the safe zone distance [RNPG algorithm] as discussed earlier.



The point of reference of formation is known as the UAV point of Formation. The mid - point of the route is usually considered as the probable point of reference. The nearest Navigation points in the registered route/ flight plan are considered i.e. one above the mid - point and one below the mid – point. The Expected time of arrival (ETA) between UAV current position to the points of reference are computed. The average ETA from the Nav. Point – 1 ETA slots, thus computed is taken and compared with the other set of Average ETA of Nav. Point – 2. The shortest ETA of both reference nav. Points is only taken into consideration.

The time slots of all UAVs in this reference nav. Points is stored. The shortest individual ETA becomes the leader and positions in left – right reference pane are occupied by UAV with ETA in descending order. Though this method gives the rough approximation of the leader; the following are the assumed control laws for UAV Leader viz.

a. Must be agile or maneuverable enough to travel the route
b. Should have long endurance limits
c. On board sensors must be able to detect possible threats, if any in hostile environment
d. Must be able to communicate with its peers or colleagues even in hostile environment.

VI. CONCLUSION

Realization and successful implementation of Swarm Co-ordination & Control Mechanism is one stepping stone towards achieving greater autonomy. Integration of UAV based entities to national air space could give boost to cheaper, efficient, timely business opportunities like Pizza delivery by drones, send basic first aid/ medical supplies to nearby hospital/ nursing homes avoiding local transportation problems etc.

ACKNOWLEDGMENT

I personally indebted and with deep gratitude thank Dr. D. Gowri Sankara Rao, Correspondent, B. R. R. Degree College for his support, guidance. I would like to thank every member of my College and department for assisting me and coming up with this proposal.

REFERENCES

[1] D. Ravi Vikranth "Random Navigation Points Generation in Constrained Environment", International Journal of Computer Science Trends and Technology (IJCST), ISSN 2347-8578, Vol. 5, Issue 2, 2017 pp 462-466

[2] D. Ravi Vikranth and S. Karthik "Autonomous Air Traffic Controller – Collision Avoidance System for UAV using MDP", IRACST – International Journal of Computer Science and Trends, Vol. 6 Issue No. 2, 2016, pp. 437 – 445.

[3] Ahmad Drak, Md. Hejase, et.al. "Autonomous Formation Flight Algorithm and Platform for Quadcopter UAVs", International Journal of Robotics and Mechatronics, Vol. 2, Nov' 2014

[4] Zhang F, Grocholsky B and Kumar V, "Formation for Localizations of Robot Networks", Proceedings of IEEE – International Conference on Robotics and Automation, April' 2004.

[5] Simon X Yang, Max Meng et. Al., "A Bilogical Inspired Neural Network Approach to Real Time Collision free Motion planning of a Non – holonomic car like robot", Proceedings of 2000 IEEE/ RST International Conference on Intelligent Robots and Systems, 2000, pp 239-244

[6] A. Das, R. Fierro et. al., "A Vision based formation Control Framework", IEEE Transactions on Robotics and Automation, Vol. 18, Issue 5, pp. 813 – 825, 2002

[7] C. Yu, B. Fidan, BD Anderson, "Persistence Acquisition and Maintenance for autonomous formation", Proceedings of Workshop on Unmanned Aerial Vehicle, 2nd. International Conference on Intelligent Sensors, Melbourne, Australia, Dec.' 2005, pp. 379-384.3

[8] Laurent Jacolin and Robert F Stengel, "Evaluation of a Co-operative Air Traffic Management Model using Principled Negotiation between Intelligent Agents", American Institute of Aeronautics and Astronautics, 1998, AIAA-98-4103.

[9] DBEdwards, TA Bean et.al., "Leader Follower Algorithm for Multiple AUV Formations", Autonomous Underwater Vehicles, IEEE/ OES, 2004, pp. 40-46.

[10] S.Grossberg, "Non Linear Neural Networks: Principles, Mechanisms and Architecture", Neural Networks, Vol. 1, 1998, pp 17-61.