FORMATION AND CONTROL OF FLY ROBOTS FOR BORDER SECURITY

Project Reference No.: *45S_BE_4749*

College ***:*** *Ramaiah University of Applied Sciences, Bengaluru*
Branch ***:*** *Department of Electrical and Electronics Engineering*
Guide(s) ***:*** *Mrs. B K Swathi Prasad*
Dr. Jayahar Sivasubramanian
Student(S) ***:*** *Ms. Darshana Deb*
Mr. Abhiram Ramesh
Ms. Harshitha H Gowda

Keywords:
Formation; Control; Fly Robots; CAD

Introduction:

Fly robots are attracting many researchers recently, because of their superior maneuverability and aerodynamic performance, especially in the low Reynolds number flight regime. Also, due to light weight, such small robots are extensively used for surveillance in unknown terrains and environments. Because of these reasons, we attempt to develop a flapping wing fly robot that is capable of vertical take-off, hovering and landing using minimum power supply. And form a pattern or configuration using these fly robots.

Fly robots adopt flapping mechanism for generating thrust and lift during flight. These flapping vehicles have been forgotten due to the success of fixed wing aircrafts. But recently the rebirth of flapping vehicles is occurring in the form of Unmanned Air Vehicles (UAV). This rebirth is occurring mainly due to the requirement of high altitude surveillance. Due to their shape fly robots look like a natural bird and it doesn't attract enemy's attention.

Coordinated behaviour of multiple fly robots leads to a formation configuration. The configuration includes open and closed contours. Closed contour refers to ring formation in the form of a polygon. Open contour refers to chain formation in the form of a tree. These patterns are maintained by proper coordination, control and communication. The formation in any geometric shape is stabilized using control techniques. Following is a summary of few recent works that highlights the limitations in the analysis of formation in multi robot systems:

1. The centralized control strategy [1] is developed to obtain a polygon (a closed contour). Only a fixed, rigid formation can be obtained.
2. The decentralized control strategy [2] is developed to move agents collectively in the desired pattern. The rectangular lattice pattern is formed after achieving consensus by assigning identity or unique number to each agent. The agent collision is not addressed during the agent movement from one line to another.
3. The synchronous control strategy [3] is developed using transition and rotation dynamic model for agents to follow in real-time. The switching of formation from ellipse to rectangle is not well-defined. The transition of formation is not addressed while satisfying the formation constraints.

4. The neural network control [4] strategy is developed for leader-follower configuration for formation. The desired geometric pattern is not achieved while maintaining the flocking behavior.
5. The adaptive control strategy is developed for formation control of autonomous vehicles [5]. The formation transition is not addressed with the changes in the terrain geometry.
6. The adaptive control strategy is combined with the neural network for formation switching while flocking and avoiding the collision [6]. The desired pattern with avoiding collision is not obtained.
7. The predictive model is used in the pattern formation [7,8]. The predictive model in the model-based Reinforcement Learning (RL) is viewed as the constrained control strategy. Any missing information in the predictive model affects the stability of the pattern formation. The issues of model-based control strategy for pattern formation have not been addressed.

The formation error depends on the system model and changes in parameter of the system affects the performance and stability. There is a need to develop a decentralized adaptive control strategy to overcome the disadvantages of centralized control strategy and requirement of the system model. In this project a decentralized model-free based RL is developed for triangular formation.

Objectives:

The objectives of the project are intended to perform the following tasks:

a) To perform literature review on fly robots, formation control and aerodynamics of fly robots.
b) To develop an algorithm for fly robot in triangular formation.
c) To develop CAD model of fly robot with its system and subsystem.
d) To develop a fly robot model.
e) To test the fly robot for surveillance.

Methodology:

The proposed work is motivated from the deterministic cleaning-robot MDP. This deterministic problem in Figure 1 shows that a cleaning-robot must collect can and charge the battery.

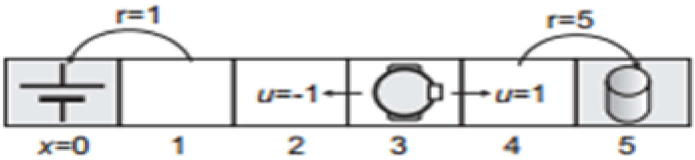

Figure 1. Deterministic Cleaning Robot [9]

In the above problem, the states-space is discrete and contains 6 states, where the robot moves to left or right depending on the optimal action chosen. In the proposed work, the same idea is projected for 3 robots and the robot transit in the states space to find its position. The proposed model is described in Fig. 2. In real – time, the geo-tag position sensor will be attached and considered as states- space. The robots are awarded or penalized based on their positions in the states – space.

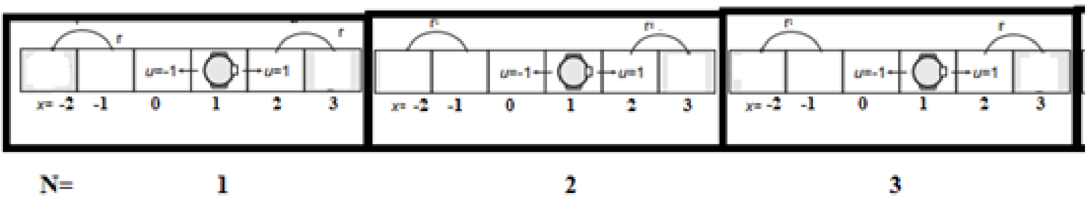

Figure 2. Proposed Model for Pattern Formation

Proposed structure for learning:

Initially first agent starts to explore the States-Space from States−Space ID: 1 till it computes its position and gets locked. Here lock indicates whether the agent should continue with the learning process or not.

Decentralized learning for multi-agents:

In this work, the decentralized control topology utilizes Q learning for making agents to learn parallel in both x and y coordinates. Q learning is used to find an optimal action-selection policy for given action-state pair. Optimal action is chosen based on deterministic Markov decision process (MDP). MDP model contains a set of states; 's', a set of possible actions; 'a', reward function $R(s,a)$, a description or task 'T'. The deterministic actions are represented using $T: S \times A \rightarrow S$. For each action-state pair, new state is computed based on the Q-learning epsilon-greedy exploitation method. This exploitation method is the policy adopted for computing the next action of an agent and is computed using Eq.1. The action space is defined as $A = \{-1,1\}$, where an agent can move either to the left ($a = -1$) or to the right ($a = 1$)

$$Action = \underset{a \in A}{argmax}\, Q(state\_id, a) \quad (1)$$

The Q function is computed using Eq.2.

$$\begin{aligned} Q(state\_id, action\_id) & \\ = Q(state\_id, action\_id) &+ alpha\ (next\_reward \\ &+ gamma * max(Q(next\_state\_id, :)) \\ &- Q(state\_id, action\_id)) \end{aligned} \quad (2)$$

The goal of the approach is to find an optimal policy that makes the return or desired target from any initial state. Here $gamma$ is the learning rate schedule, $alpha$ is the exploitation rate schedule and $reward$ is given to the agent for action taken based on the policy. The Q function is executed till the agent finds its position and locks itself without moving further. For the next agent, the initial state is considered as the previous agents' final position and performs Q learning till the agent reaches its position.

Reward computation:

The agent space contains six agents, denoted by integers 1 to 6: $N = \{1,2,3,4,5,6\}$. The next state of each agent is computed and updated based on the action taken from the optimum Q value. Each agent is defined in the discrete states-space and contains 6 distinct states, denoted by real numbers -2 to 3: $X = \{-2,-1,0,1,2,3\}$. The agent can move to the left $(a = -1)$ or to the right$(a = 1)$. The next state of each agent is updated till it finds its position in X and lies within the states space of X. The agent gets positive reward when the agent performs certain action, is penalized if the agent performs action even after finding its position and in neither case, the agent is given as lesser negative reward. The corresponding transition function for the above problem is given in Eq. 3.

$$f(x,u) = \begin{Bmatrix} x+u \;\; if -2 \le x \le 3 \\ x \;\; otherwise \end{Bmatrix} \tag{3}$$

The corresponding reward function is defined for each agent is defined independently for x and y coordinates.

1) Reward computation for x-coordinate of each agent:

The reward function for each agent is given in Eq. 4 to Eq. 6.

$$R_1(x,u) = \begin{Bmatrix} 10 & if\ x = -2\ and\ u = 1 \\ 10 & if\ x = 0\ and\ u = -1 \\ -10 & if\ x = -1\ and\ u = 1\ or -1 \\ -1 & otherwise \end{Bmatrix} \tag{4}$$

$$R_2(x,u) = \begin{Bmatrix} 10 & if\ x = -1\ and\ u = 1 \\ 10 & if\ x = 1\ and\ u = -1 \\ -10 & if\ x = 0\ and\ u = 1\ or -1 \\ -1 & otherwise \end{Bmatrix} \tag{5}$$

$$R_3(x,u) = \begin{Bmatrix} 10 & if\ x = 0\ and\ u = 1 \\ 10 & if\ x = 2\ and\ u = -1 \\ -10 & if\ x = 1\ and\ u = 1\ or -1 \\ -1 & otherwise \end{Bmatrix} \tag{6}$$

2) Reward computation for y-coordinate of each agent:

The reward function for each agent is given in Eq. 7 to Eq. 9.

$$R_1(x,u) = \left\{ \begin{array}{ll} 10 & if\ x = -1\ and\ u = 1 \\ 10 & if\ x = 1\ and\ u = -1 \\ -10 & if\ x = 0\ and\ u = 1\ or -1 \\ -1 & otherwise \end{array} \right\} \tag{7}$$

$$R_2(x,u) = \left\{ \begin{array}{ll} 10 & if\ x = 0\ and\ u = 1 \\ 10 & if\ x = 2\ and\ u = -1 \\ -10 & if\ x = 1\ and\ u = 1\ or -1 \\ -1 & otherwise \end{array} \right\} \tag{8}$$

$$R_3(x,u) = \left\{ \begin{array}{ll} 10 & if\ x = 0\ and\ u = 1 \\ 10 & if\ x = 2\ and\ u = -1 \\ -10 & if\ x = 1\ and\ u = 1\ or -1 \\ -1 & otherwise \end{array} \right\} \tag{9}$$

Design and development of fly robots:

The modelling of flapping wing UAV “fly robots” which is capable of performing vertical take-off, landing, hovering and horizontal flight was done using CATIA V5. The vehicle consists of a fuselage, wing, transverse gear system and tail. The fuselage is 27 cm long and it provides support to the transverse gears, electronics, servo motors, battery and camera. It has been made sure that the fuselage weighs light by proper weight reduction (see Figure 3). The wing has a total span of 60 cm and chord length of 20 cm (see Figure 4). A triangular tail of 10 cm base and 10 cm height is used to increase the stability and control (see Figure 5). The tail also includes two servos for pitch and roll control. A flat wing configuration with no camber has been chosen. The wing will be assembled as a single sheet flat wing with carbon fibres as the skeleton material, in order to achieve the desired flexibility and minimum weight. Carbon fibres are also used in the construction of the wings to provide a lightweight and stiff structure of spars (Figure 4). The material used for the wing cover is nylon sheet.

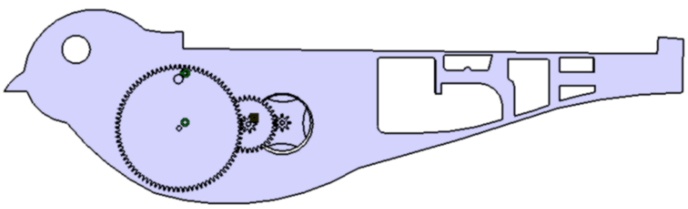

Figure 3. CAD model of the fuselage of the fly robot.

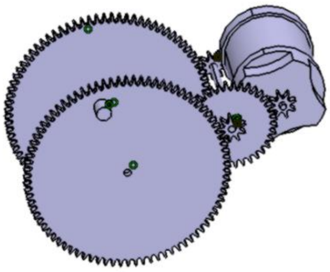

Figure 4. CAD model of the transverse gear system.

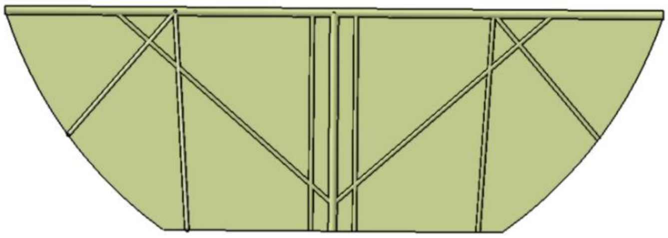

Figure 5. CAD model of the wing of the fly robot

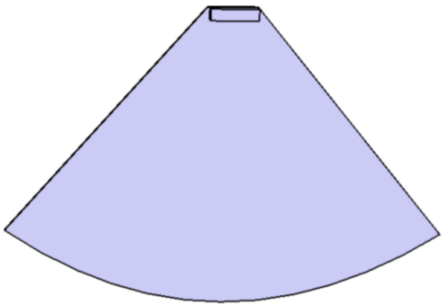

Figure 6. CAD model of the tail of the fly robot.

Results and Conclusions

The agents' transit from one state position to the other based on the optimal deterministic action till the agent finds its position and locks in that position. During transition, several episodes the agent has undergone to find its position and by using next state transition history of the agent in x and y coordinate, the pattern is obtained. The traversal episodes are listed in Table 1. The triangular formation is shown in Figure 7. After travelling through the states-space, the triangular formation is achieved.

Final CATIA model of the fly robot is shown in Figure 8. The specifications of the model were arrived at based on literature survey and preliminary analysis. Before the fabrication of the fly robot, we will conduct further structural and aerodynamic analysis to confirm the performance of the fly robot.

Table 1 Travel History of Agents

Traversal History x				Traversal History y			
States – Space ID	Next State in x	Agent ID	Episodes	States – Space ID	Next State in y	Agent ID	Episodes
1	-2	1	1	2	-1	1	1
2	-1	1	2	1	-2	1	2
1	-2	2	1	2	-1	1	3
2	-1	2	2	3	0	1	4
1	-2	2	3	2	-1	2	1
1	-2	2	4	1	-2	2	2
2	-1	2	5	2	-1	2	3
3	0	2	6	2	-1	3	1
2	-1	2	7	1	-2	3	2
3	0	2	8	2	-1	3	3
4	1	2	9				
1	-2	3	1				
2	-1	3	2				
1	-2	3	3				
1	-2	3	4				
2	-1	3	5				
3	0	3	6				

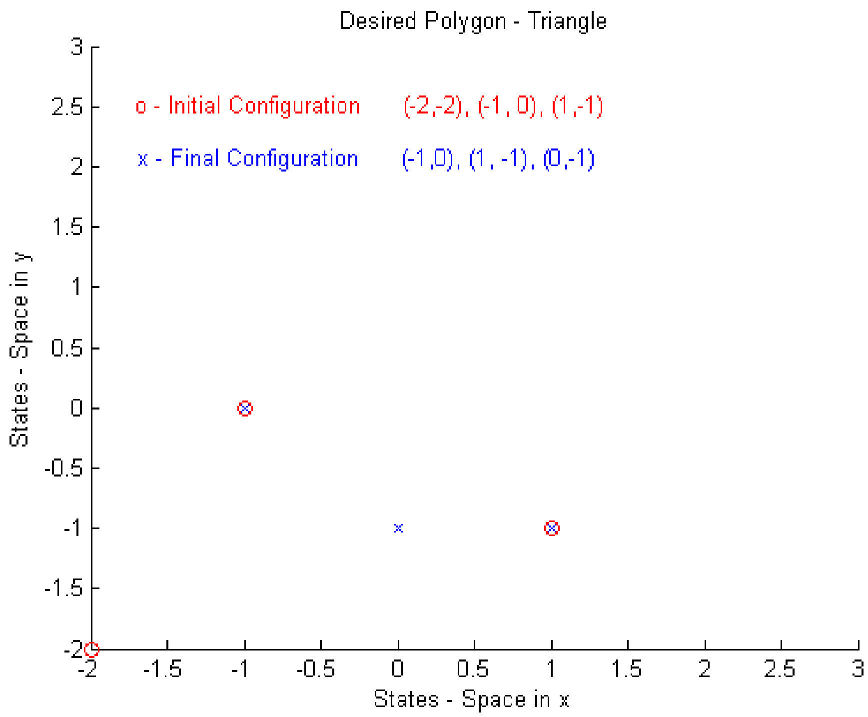

Figure 7. Triangular formation by multi robots.

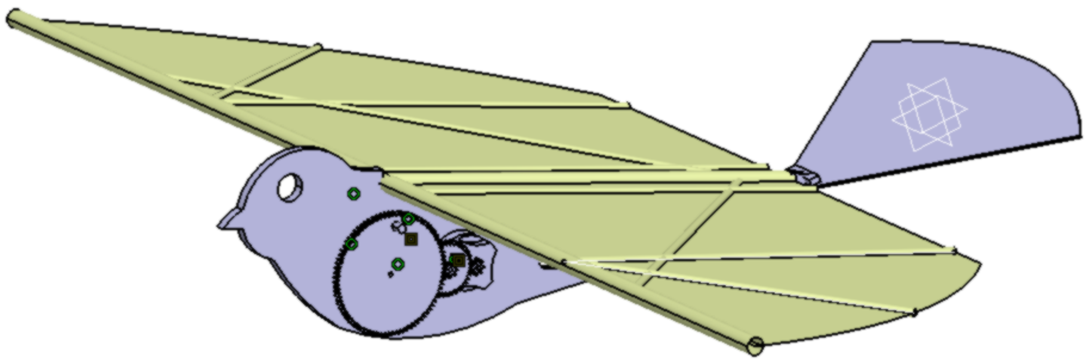

Figure 8. Final CATIA model of the fly robot.

Scope for future work:

In the next phase of the project, we will complete the fabrication of the fly robots and test the robots for surveillance.

Reference:

[1.] Jaydev P Desai, James P Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. IEEE transactions on Robotics and Automation, 17(6):905{908, 2001.

[2.] Teddy M Cheng and Andrey V Savkin. Decentralized control of multi-agent systems for swarming with a given geometric pattern. Computers & Mathematics with Applications, 61(4):731{744, 2011.

[3.] Ibrahim MH Sanhoury, Shamsudin HM Amin, and Abdul Rashid Husain. Synchronizing multi-robots in switching between different formations tasks while tracking a line. In Trends in Intelligent Robotics, Automation, and Manufacturing, pages 28{36. Springer, 2012.

[4.] Caihong Zhang, Tairen Sun, and Yongping Pan. Neural network observer-based nitetime formation control of mobile robots. Mathematical Problems in Engineering, 2014, 2014.

[5.] A. Guillet, R. Lenain, B. Thuilot, and P. Martinet. Adaptable robot formation control: Adaptive and predictive formation control of autonomous vehicles. IEEE Robotics Automation Magazine, 21(1):28{39, March 2014. ISSN 1070-9932. doi: 10.1109/MRA. 2013.2295946.

[6.] Quan Shi, Tieshan Li, Jingqi Li, CL Philip Chen, Yang Xiao, and Qihe Shan. Adaptive leader-following formation control with collision avoidance for a class of second-order nonlinear multi-agent systems. Neurocomputing, 350:282{290, 2019.

[7.] E. A. Oury Diallo and T. Sugawara. Multi-agent pattern formation: a distributed model free deep reinforcement learning approach. In 2020 International Joint Conference on Neural Networks (IJCNN), pages 1{8, 2020. doi: 10.1109/IJCNN48605.2020.9207657.

[8.] Bo Wang, Sergey Nersesov, and Hashem Ashra uon. Time-varying formation control for heterogeneous planar underactuated multi-vehicle systems. Journal of Dynamic Systems, Measurement, and Control, 2021.

[9.] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. Reinforcement learning and dynamic programming using function approximators. CRC press, 2010