# PSO method for the control of a fleet of UAV

**Abstract:**

To control approach of a fleet of Unmanned Aerial Vehicles based on a leader of the fleet. The optimization method used will decide the virtual leader. The end goal is to find optimal positions of each UAV present in the fleet at every instant. This is based on the task of minimizing a predefined objective function. For simulation purposes a 2-D plane is being considered with weightless objects (particles of swarms). While achieving the end goal particles are not supposed to collide with each other and maintain safe distance which is governed by a function. Recently, various applications of UAVs are emerging in various sectors. Using multiple vehicles in a fleet can help one achieve a task in lesser duration and maximizing the output of task being done as well, but this requires coordination between members of the fleet. Particle Swarm Optimization is a meta-heuristic algorithm that helps optimize the given function.

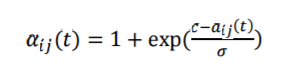
*The cost function of each UAV in the fleet at a given time t is defined as :*

C(t) = ρ(||Pd(t) – (xi(t)+h|| - aip(t)) + ij(t) (||xj(t) – (xi(t) + h)|| - aij(t))

where and h 2 and ρ >> 1

The main goal is to minimize h for each UAV ,i , thus minimizing the cost function Ai(t).

*In order to introduce this condition in the cost function ΛB (t), we define new functions ij(t) between each agent i and its neighbors j such that*



where value of aij(t) depends upon distance of 2 UAV with target and safety distance c and aij(t) < c and converges to value 1 where aij(t) >> c.

**PSO Algorithm:**

**In simple words the algorithm will have these steps –**

1. Random generation of an initial population.
2. Reckoning of a fitness value for each particle and this will directly depend on the distance to the optimum or the target value/space.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop. Otherwise go back to 2

**Pseudo Code:**

The pseudo code of the procedure is as follows  
  
For each particle   
    Initialize particle  
End  
  
Do  
    For each particle   
        Calculate fitness value  
        If the fitness value is better than the best fitness value (pBest) in history  
            set current value as the new pBest  
    End  
  
    Choose the particle with the best fitness value of all the particles as the gBest  
    For each particle   
        Calculate particle velocity  
        Update particle position  
    End   
While maximum iterations or minimum error criteria is not attained

