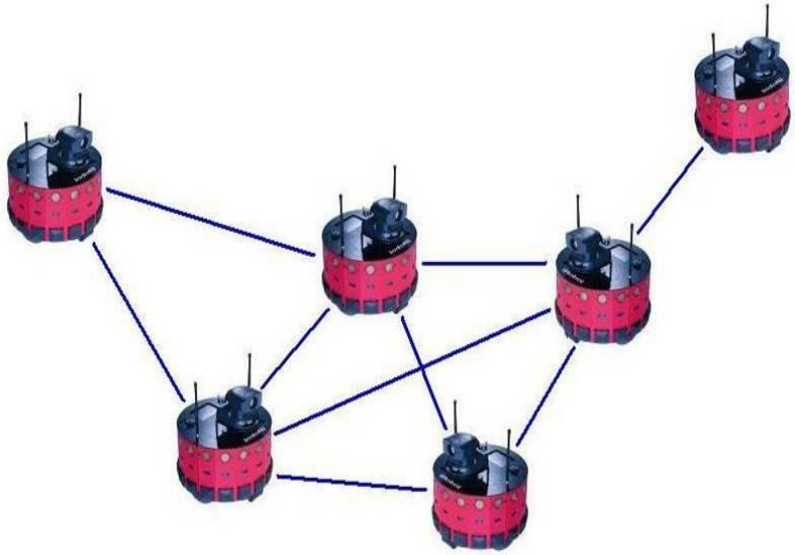
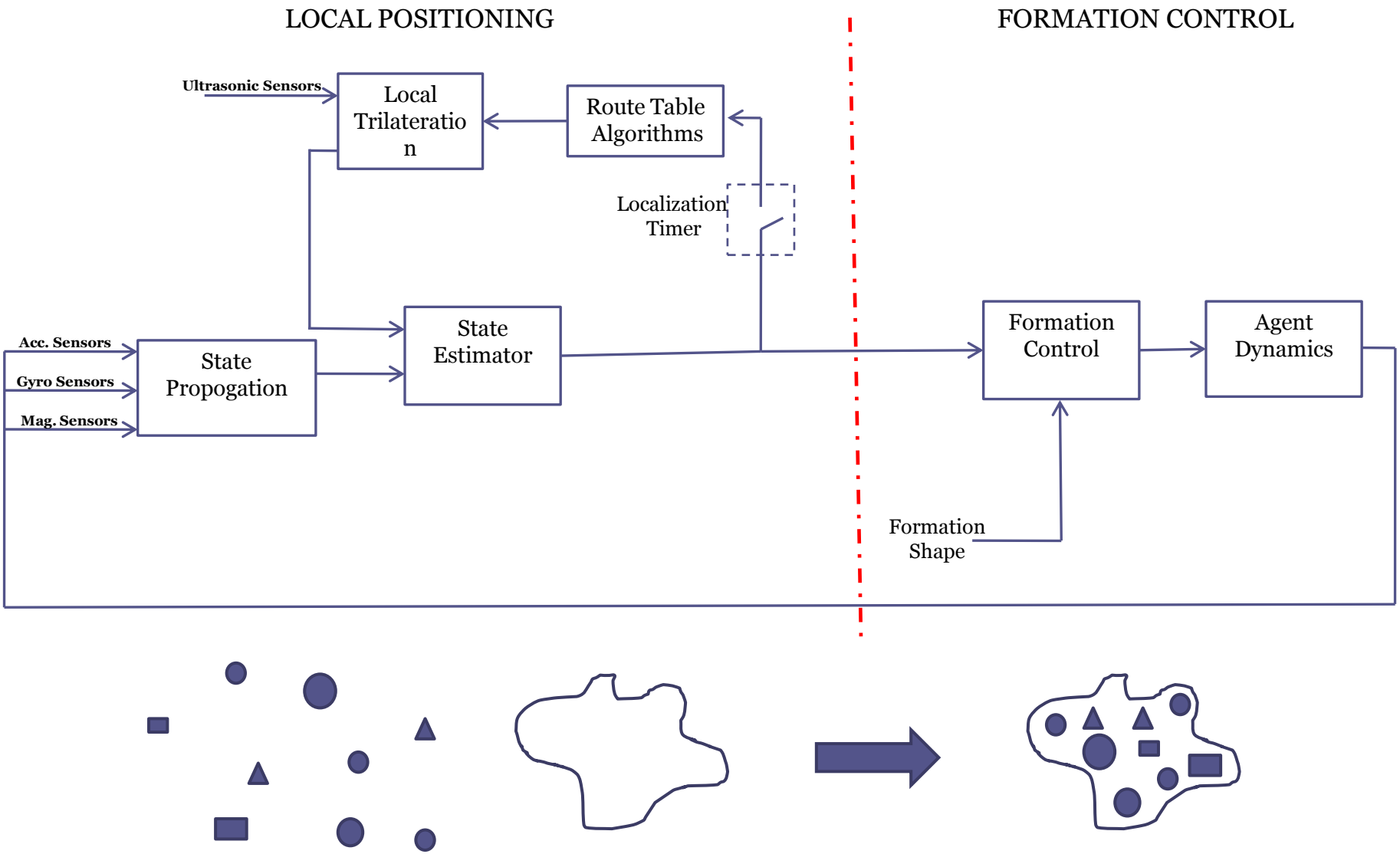
DYNAMIC FORMATION CONTROL WITH HETEROGENEOUS MOBILE ROBOTS



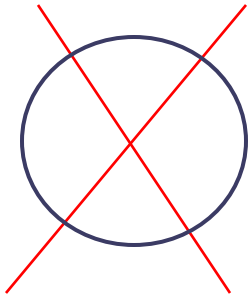
SYSTEM DESIGN

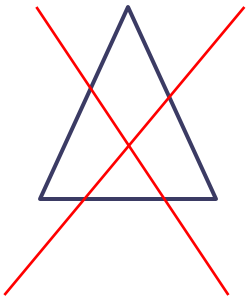


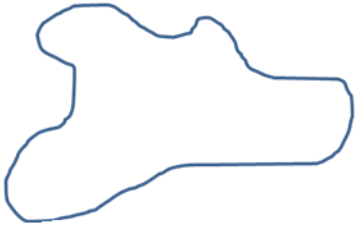
Problems & Requirements

In this project, it is aimed to create a method of formation control for a given complex shape with heterogenous agents.

1) Complex geometric shapes will be requested from the swarm



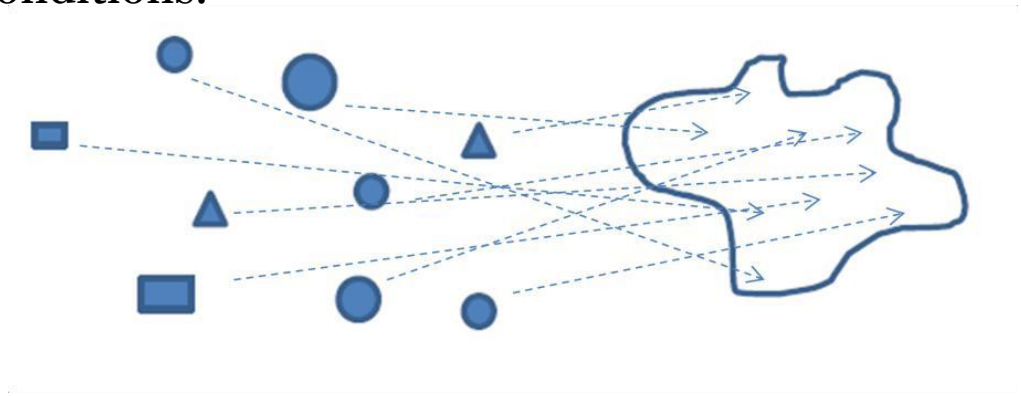




Problems & Requirements

2) Agents will have volumes, they may collide with the other ones and the obstacles in the environment

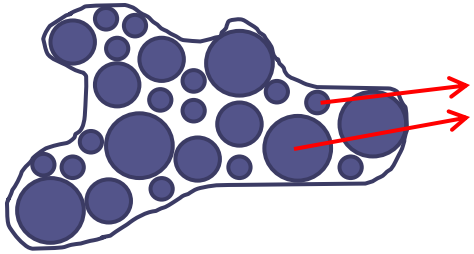
3) Each agent must decide the location in which it will be positioned in the formation by taking into account energy consumption issues, obstacles in the environment, other agents' decisions/conditions.



Project Breakdown Structure

General formation control problem can be divided into two main parts.

1) Partitioning complex formation shape into goal states.
In this context, potential goal states positions must be determined in the given complex shape to cover the whole formation homogenously.



2) Decision Process Between Possible Goal States
During decision process, the cost of reaching different goal states will be the main criteria for each agent. It is obvious that each agent will try to choose a goal state with minimum cost according to its position and the orientation in the environment. But the cases in which two or more agents are willing to reach the same goal point must be handled to optimize the overall utility of the swarm.

SHAPE PARTITIONING WITH ARTIFICIAL FORCES

- This process is used to detect the positions of the goal states in a given complex shape with an offline simulation.
- Two artificial forces, resistive and intermember forces will be used to provide a uniform distribution of the agents in the given formation.

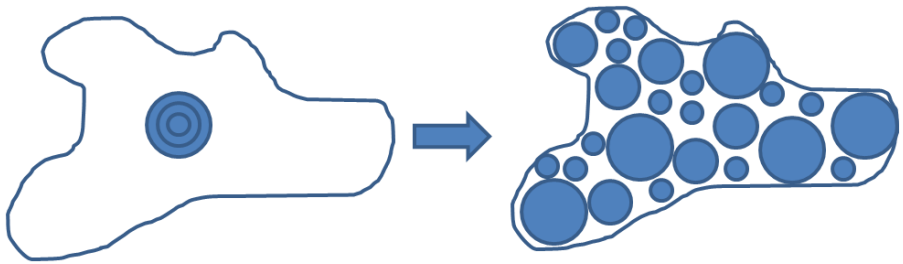
- Repulsive Forces

$$F_{i,r,x} = k_f \sum_{j=1}^{n} \left(\frac{x_i - x_j}{d_{ij}} \frac{1}{(d_{ij} - r_i)^2} \right)$$

- Intermember Forces

$$F_{i,m,x} = k_m \sum_{j=1, j \neq i}^{n} \left(\frac{x_i - x_j}{d_{ij}} \frac{1}{(d_{ij} - d_o)^2} \right)$$

where $d_o = r_i + r_j$



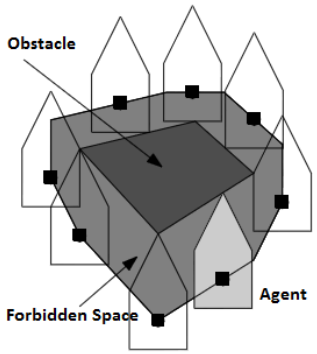
DECISION OF GOAL STATES

At this point, the complex formation control problem is reduced down to a problem in which every agent is expected to decide individually where to position in a given set of possible goal states $g_i \in G$

1)Calculation of Free Configuration Space

$$C(R_i) = C_{free}(R_i, S) + C_{forb}(R_i, S)$$

Forbidden Space : $S_1 \oplus S_2 := \{p + q : p \subset S_1, q \subset S_2\}$



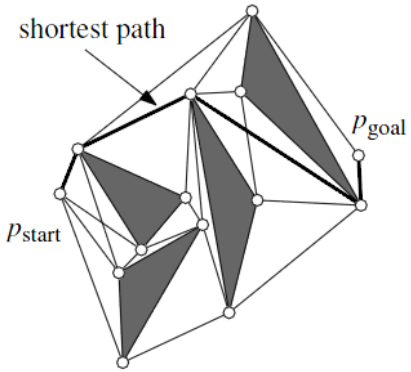
Forbidden Space

DECISION OF GOAL STATES

2)Visibility Graphs

•Consider set of obstacles in the environment is augmented with the Minkowski Sums described in the previous chapter. The shortest path between p_start and p_goal among a set S of augmented polygonal obstacles consists of arcs of the visibility graph

$$\gamma_{vis}(S^*) \text{ where } S^* := S \cup \{p_{start}, p_{goal}\}$$



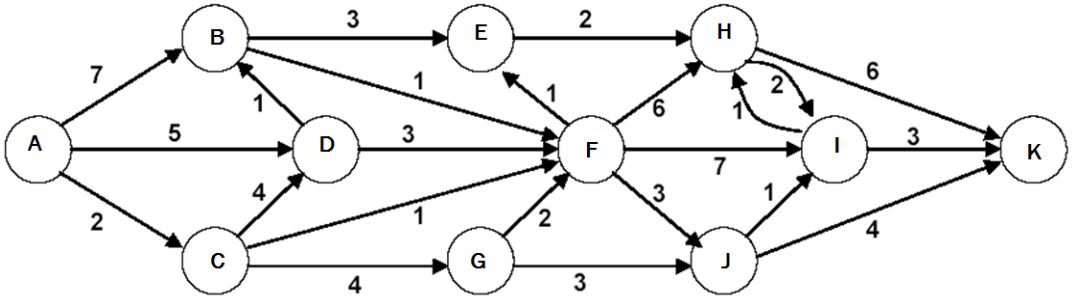
Visibility Graph

DECISION OF GOAL STATES

3)Dijkstra's Algorithm

- Dijkstra's algorithm is a tree search algorithm for finding the shortest paths between nodes in a graph

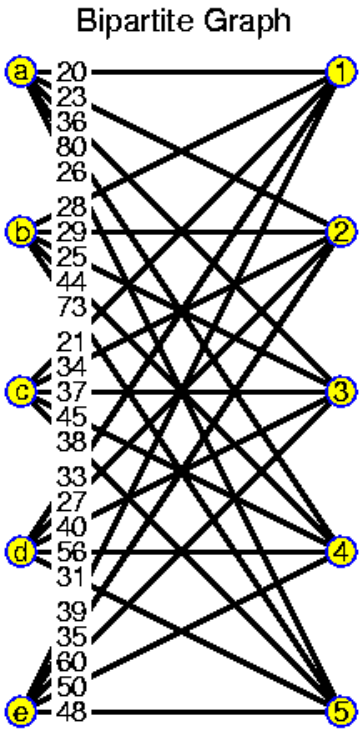
Up to now, we have calculated the all possible paths to the given goal states for an agent by using visibility graphs. With the help of Dijkstra's algorithm , shortest paths and costs to these goal states will be calculated.



Calculation of Minimum Shortest Path From A to K

DECISION OF GOAL STATES

4)Hungarian Algorithm (Munkres Assignment Algorithm)



	Clean Bathroom	Sweep Floors	Wash Windows
Jim	$3	$3	$3
Steve	$3	$2	$3
Alan	$3	$3	$2

- The Hungarian method is a combinational optimization algorithm that solves the assignment problem in polynomial time. Worst case time complexity : O (n^3)
- Optimality is guaranteed
- Time complexity can be made better with some dynamic and incremental implementations.

VELOCITY CONTROLLER

State feedback with LQR controller;

System is augmented with an artificial error state,

$$\begin{bmatrix}\dot{v}\\ \dot{e}\end{bmatrix} = \begin{bmatrix}-b/m & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}v\\ e\end{bmatrix} + \begin{bmatrix}1/m\\ 0\end{bmatrix} F_{net} \qquad y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}v\\ e\end{bmatrix}$$

Q and R matrices used in solving Riccati equations,

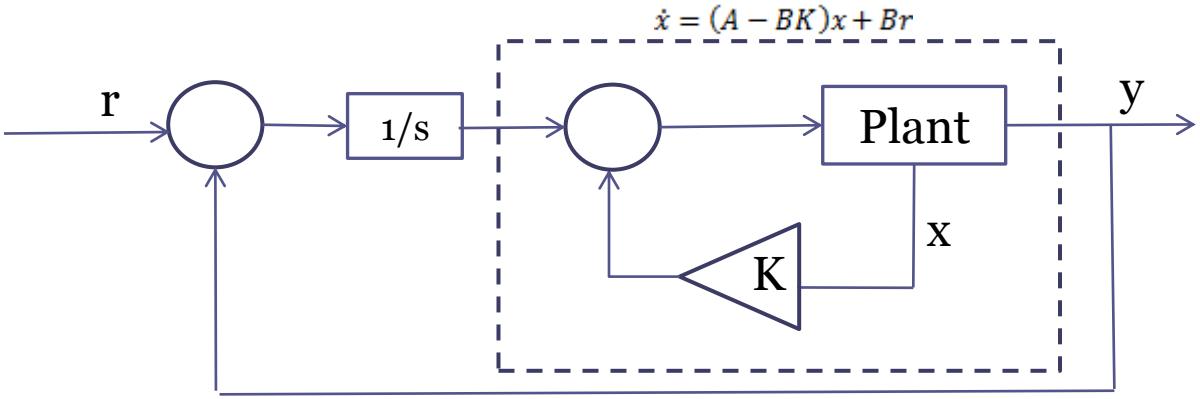
$$Q = \begin{bmatrix}q_1 & 0\\ 0 & q_2\end{bmatrix};\ R = \rho r_1 \qquad q_1 = \frac{1}{t_{s_1}(x_{1max})^2};\ q_2 = \frac{1}{t_2(x_{2max})^2}\ and\ r_1 = \frac{1}{(u_{1max})^2}$$

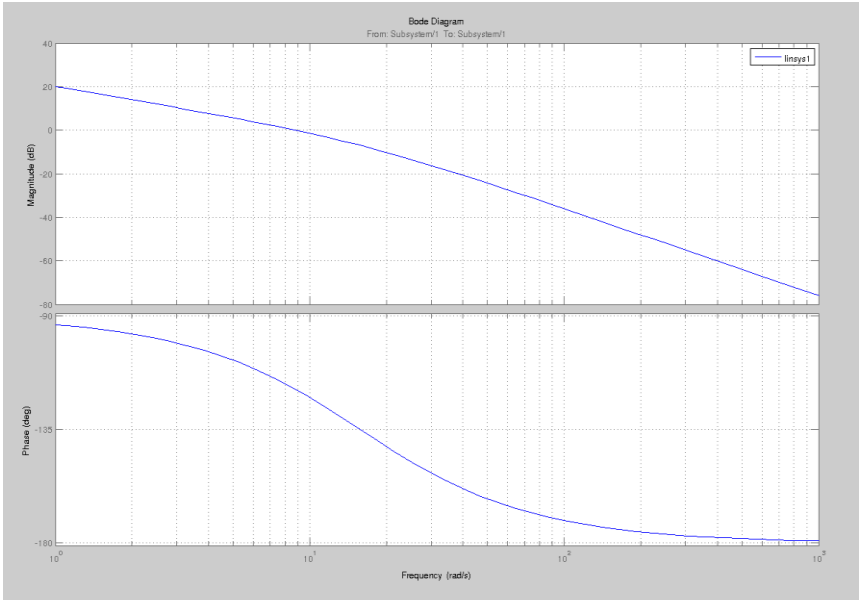
where,

t_{s_i}: *desired settling time for* x_i

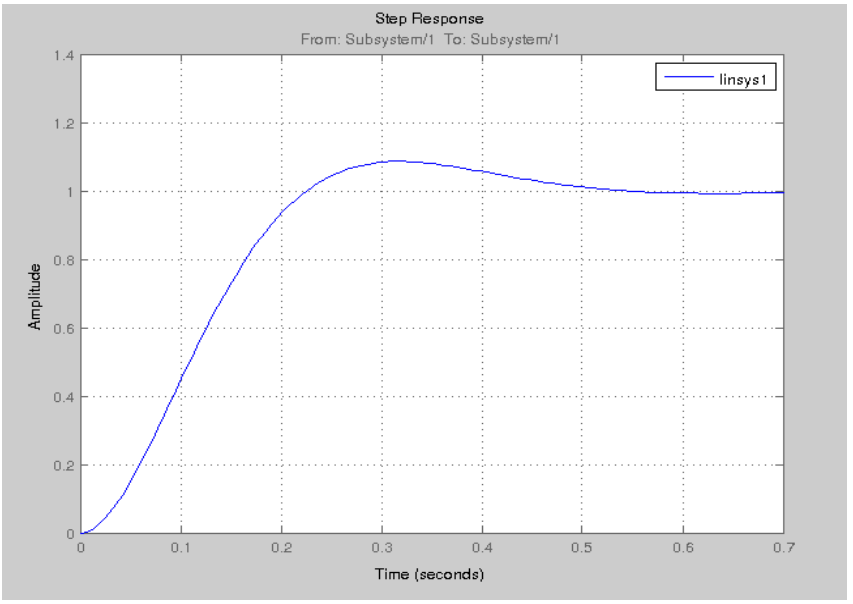
ρ: *tradeoff regulation vs control effort*

VELOCITY CONTROLLER





Open loop Bode plots



Step response

SIMULATION ENVIRONMENT