ME312

OPERATIONS MODELING AND ANALYSIS

PUNCTUAL DRONES

Presented By : Group 43

Tanmay Jain
Nikita Kanwar
Srushti Kendre
Divyanshu Rathore

MOTIVATION

The Techfest's drone show fascinated us. It sparked curiosity about route planning to prevent collisions and maximize profit in us. Now, we're diving into optimizing paths for multiple drones, eager to understand the strategies behind the impressive aerial display.

PROBLEM STATEMENT

Optimal Assignment

We employ a mixed integer linear programming approach that allows computationally efficient optimal positions minimizing distance travelled.

Trajectory Generation & optimization

Collision-free trajectories are generated using sequential convex programming in the our proposed solution. For each iteration a Convex Quadratic programming problem is solved, terminating when difference is below a threshold ensuring collision avoidance and adherance to dynamic constraints.

Quadrotor Control

The Quadrotor is a Non-linear state space model with a double integrator dynamics. Quadrotor control involves reference trajectory input which is received as output from the trajectory generation class. For trajectory tracking, we employ a backstepping control algorithm ensuring the Lyapunov stability and desired tracking trajectories expressed in term of the center of mass coordinates along (X, Y, Z) axis and yaw angle, while the desired roll and pitch angles are deduced from nonholonomic constraint

FORMULATION

Linear Time Invariant Causal Double Integrator State Space Model

Assumed to simply Trajectory Generation for reference path

Linear Time Invariant Causal State Space Model:

$$\dot{X} = f(X, U) = AX + BU \tag{1}$$

Thus for discrete time steps, this becomes:

$$X[k+1] - X[k] = h \cdot (AX[k] + BU[k])$$

$$X[k+1] = FX[k] + GU[k]$$

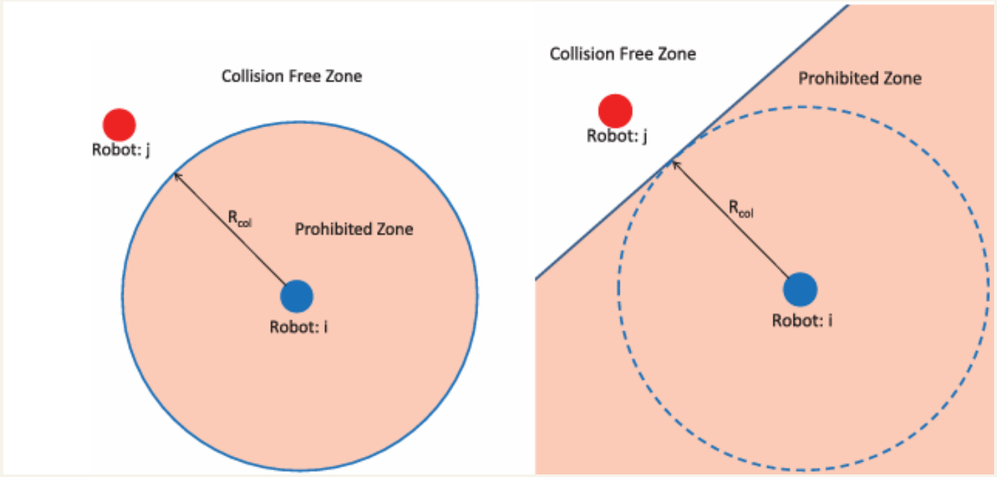
$$X[k+1] = F^{k+1}X[0] + F^{k}GU[0] + \dots FGU[k-1] + GU[k]$$
$$\text{for } k \in \{0, 1 \dots, T-1\}$$

State X: $[p, v]$ for all dimensions; p: position , v: velocity
For a 3D configuration space, we have 6 states namely (x,y,z) in pos, vel.
X: State
U: Input
$F = Id(n) + h \cdot A$: State transition matrix
$G = h \cdot B$: Input transition matrix
T_f: Final Time Step

FORMULATION

Original Decoupled Collision Avoidance constraint (Non-convex)

$$\|G(\mathbf{x}_j[k]-\bar{\mathbf{x}}_i[k])\|_2 \geq R_{\text{col}} \qquad k=k_0,\ldots,T,$$
$$i \in \{\mathcal{N}_{[j]} \cap \mathcal{P}_j\}$$



Non Convex

Convex

Decentralized Convex Problem

$$\min_{\mathbf{u}_j} \sum_{k=k_0}^{T-1} \|\mathbf{u}_j[k]\|_1 \Delta t \quad \text{subject to}$$

$$\mathbf{x}_j[k+1] = A_j[k]\mathbf{x}_j[k] + B_j[k]\mathbf{u}_j[k] + z_j[k],$$
$$k = k_0,\ldots,T-1$$
$$\|\mathbf{u}_j[k]\|_\infty \leq U_{\max} \qquad k=k_0,\ldots,T-1$$
$$\mathbf{x}_j[0] = \mathbf{x}_{j,0}$$
$$\mathbf{x}_j[T] = \mathbf{x}_{j,f}$$
$$(\bar{\mathbf{x}}_j[k]-\bar{\mathbf{x}}_i[k])^{\mathrm{T}} G^{\mathrm{T}} G(\mathbf{x}_j[k]-\bar{\mathbf{x}}_i[k])$$
$$\geq R_{\text{col}}\|G(\bar{\mathbf{x}}_j[k]-\bar{\mathbf{x}}_i[k])\|_2$$
$$k=k_0,\ldots,T, \quad i \in \{\mathcal{N}_{[j]} \cap \mathcal{P}_j\}$$

FORMULATION

Quadratic programming (our adaption)

Optimization variable: control input (U)

Equaliity constraints: Initial, Terminal constant

Inequality constraints: Collision Avoidance

Using Sequential Convex Programming Algorithm:
Cost function to optimize :

$$\text{Cost to minimize J} = min\ \{\frac{1}{2}U^T PU + q^T U\} \text{ wrt } u \tag{2}$$

subject to constraints:

$$\text{Equality Constraints: } a \cdot U = b$$

$$U_i[T_f] = 0$$

$$X_i[T_f] - F^{T_f} X_i[0] = F^{T_f - 1 - t} G U_i[t]$$

$$\text{for } t \in \{0, 1, \ldots T_f - 1\}\ ; i \in \{1, \ldots num_{robots}\}$$

$$\text{Inequality Constraints: } g \cdot U \leq h$$

$$(\overline{X_i[t]} - \overline{X_j[t]})^T * (F^{t-1-w} G)[pos] \leq -R_{col} \parallel \overline{p_i[t]} - \overline{p_j[t]} \parallel - (p_i[t] - \overline{p_j[t]})^T \overline{p_j[t]} + (p_i[t] - \overline{p_j[t]})^T (F^t X_i[0])[pos]$$

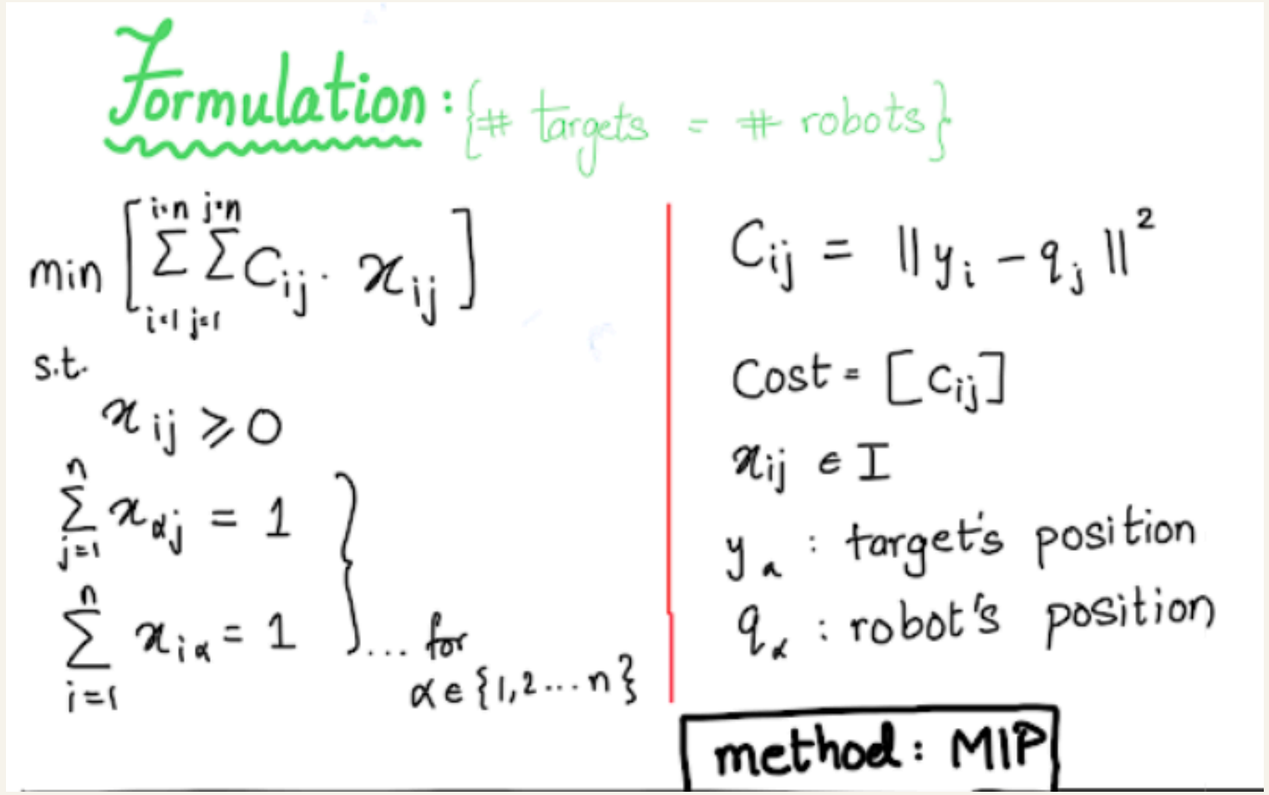
$$\text{for } t \in \{0, 1, \ldots T - 1\}\ \ ; w \in \{0, 1 \ldots t - 1\}$$

$$i, j \in \{1, 2, \ldots num_{robots}\} \text{ with } i < j$$

MIXED INTEGER LP

A mixed-integer linear program (MILP) is a problem with

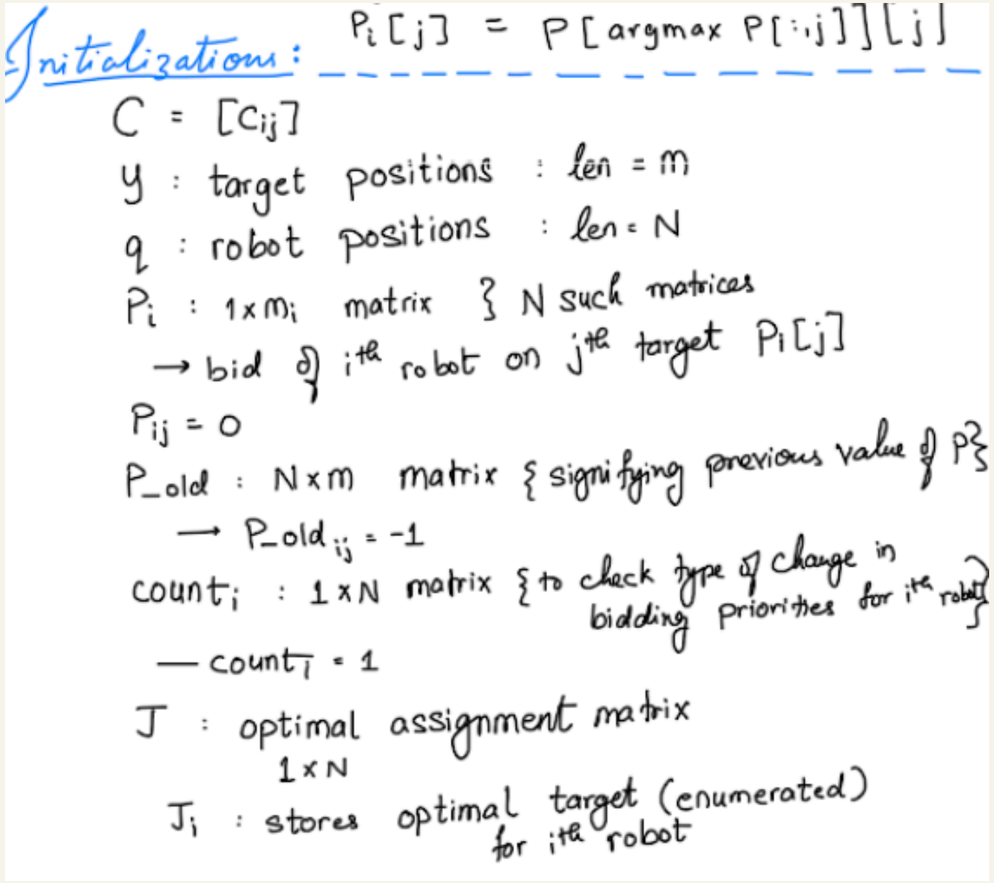
- Cost function Cij: is the l2-norm between ith and jth robots.
- The optimization variable z is a binary variable, which takes only 0,1 as values.
- The total number of robots and targets are fixed, hence one robot can only select one target and one target can only be selected once.
- MILP is NP-Hard, so it can't be solved in polynomial time unless P=NP. However, MILP can certainly be solved in exponential time by branch and bound.



AUCTION ALGORITHM

DEFINITION

The "auction algorithm" encompasses various combinatorial optimization techniques for solving assignment and network optimization problems with linear or convex/nonlinear costs, allocating resources efficiently through competitive bidding and iterative pricing mechanisms.

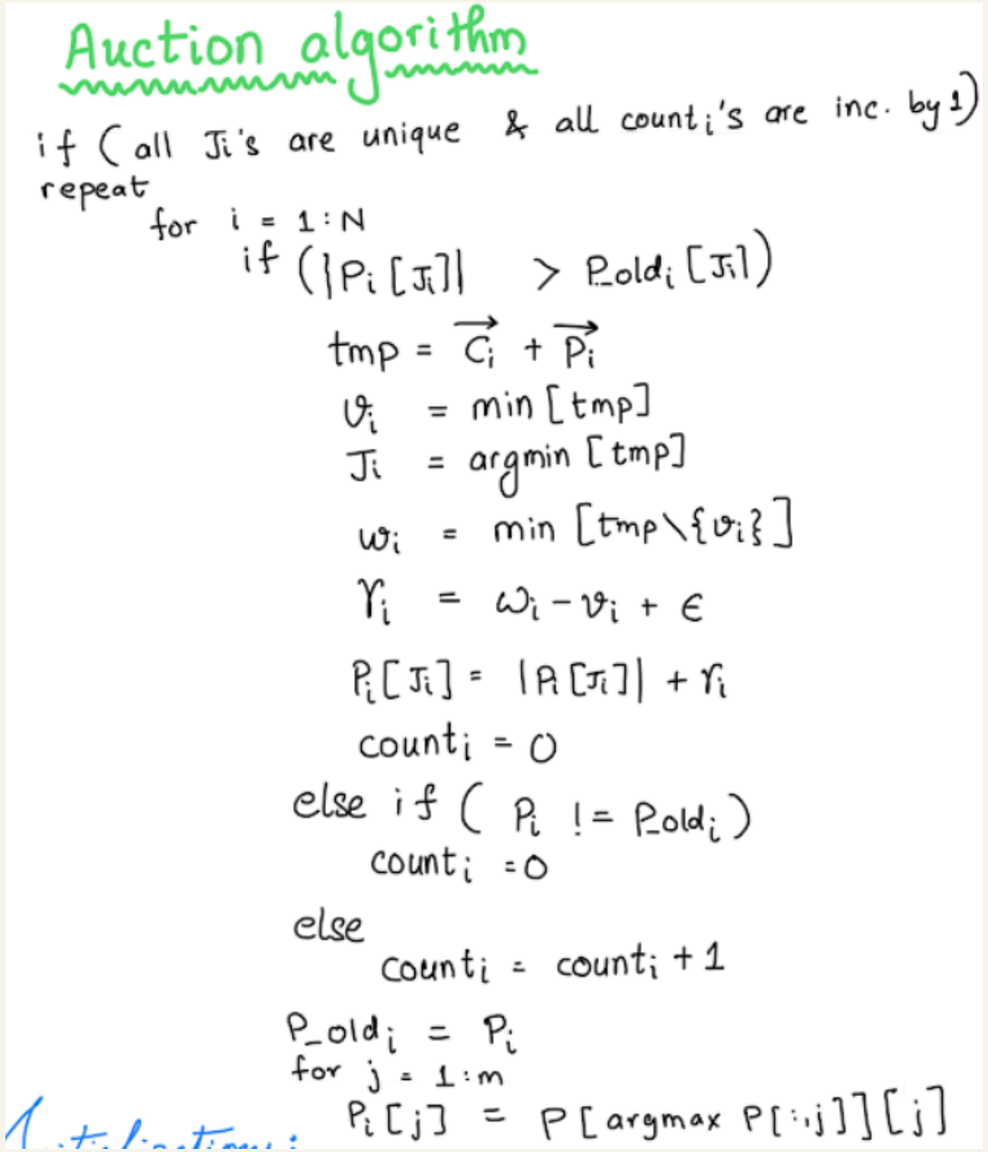


AUCTION ALGORITHM FORMULATION

What we found in application:

1. Auction was used but was infeasible for finding scaling and optimal solution.
2. Drawbacks we observed:

- Time complexity depends on spatial arrangement of nodes. (farther nodes more time)
- Reached near optimal solution if ran for a lesser number of iterations
- Found out edge cases where infeasible solution is an output (2 robots having same target)



TRAJECTORY GENERATION

Sequential convex programming (SCP) is a local optimization method for nonconvex problems that leverages convex optimization.

$$\mathcal{T}^{(k)} = \left\{ x \in \mathbf{R}^n \mid \left\| x - x^{(k)} \right\|_2 \leq \rho \right\}$$

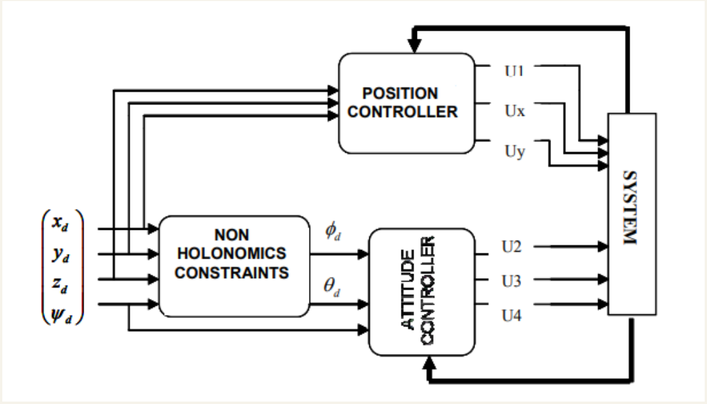
$$\mathcal{T}^{(k)} = \left\{ x \in \mathbf{R}^n \mid |x_i - x_i^{(k)}| \leq \rho_i, \ i = 1, \ldots, n \right\}.$$

Trust Regions is typically either an ℓ2-norm ball, which takes the difference of the current and nominal trajectory and iterates till the difference is within the region.

1: $\bar{\mathbf{x}}_j[k] := \mathbf{0}_{6\times 1}, \ \forall j, k$
2: $\mathbf{x}_{j,0}[k]$:= the solution to Problem 5 (decentralized convex program) with $\mathcal{P}_j = \emptyset, \ \forall j, k$
3: $\bar{\mathbf{x}}_j[k] := \mathbf{x}_{j,0}[k], \ \forall j, k$
4: Communicate $\bar{\mathbf{x}}_j[k]$ to all neighboring agents ($i \in \mathcal{N}_{[j]}$)
5: $\mathcal{K} := \{1, \ldots, N\}$
6: $w := 1$
7: **while** $\mathcal{K} \neq \emptyset$ **do**
8: **for all** $j \in \mathcal{K}$ (*run in parallel*) **do**
9: $\mathbf{x}_{j,w}[k]$:=the solution to Problem 5 (decentralized convex program), $\forall k$
10: **end for**
11: **for all** j (*run in parallel*) **do**
12: $\bar{\mathbf{x}}_j[k] := \mathbf{x}_{j,w}[k], \ \forall k$
13: Communicate $\bar{\mathbf{x}}_j[k]$ to all neighboring agents ($i \in \mathcal{N}_{[j]}$)
14: **if** $\|\mathbf{x}_{j,w}[k] - \mathbf{x}_{j,w-1}[k]\|_\infty < \epsilon_{\text{SCP}} \ \forall k$ and $\|G(\mathbf{x}_{j,w}[k] - \mathbf{x}_{i,w}[k])\|_2 > R_{\text{col}} \ \forall k, \forall i \in \mathcal{N}_{[j]} \cap \mathcal{P}_j$ **then**
15: Remove j from $\mathcal{K}$
16: **end if**
17: **end for**
18: $w := w + 1$
19: **end while**
20: $\mathbf{x}_{j,w-1}[k]$ is the approximate solution to Problem 3

QUADROTOR CONTROL

The Quadrotor dynamical state space model has 12 states; position and velocity in 3 dimensions (translational and rotational)



$$
\begin{cases}
\dot{\phi} = p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\
\dot{\theta} = q[c(\phi)] - r[s(\phi)] \\
\dot{\psi} = r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)} \\
\dot{p} = \frac{I_y - I_z}{I_x}rq + \frac{\tau_x + \tau_{wx}}{I_x} \\
\dot{q} = \frac{I_z - I_x}{I_y}pr + \frac{\tau_y + \tau_{wy}}{I_y} \\
\dot{r} = \frac{I_x - I_y}{I_z}pq + \frac{\tau_z + \tau_{wz}}{I_z} \\
\dot{u} = rv - qw - g[s(\theta)] + \frac{f_{wx}}{m} \\
\dot{v} = pw - ru + g[s(\phi)c(\theta)] + \frac{f_{wy}}{m} \\
\dot{w} = qu - pv + g[c(\theta)c(\phi)] + \frac{f_{wz} - f_t}{m} \\
\dot{x} = w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\
\dot{y} = v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - w[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] + u[c(\theta)s(\psi)] \\
\dot{z} = w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)]
\end{cases}
$$

QUADROTOR CONTROL

IIT Bombay | 2024

Backstepping control is a nonlinear control technique used to stabilize systems that can be represented as a cascade of subsystems. The method involves designing a control law for each subsystem while ensuring Lyapunov stability at each step ensuring decoupled dynamics for all states which are independent, hence "backstepping" through the cascade.

Integrator Backstepping

- Theorem of Integrator Backstepping:

$$\dot{x} = f(x) + g(x)z$$ — Nonlinear System

$$\dot{z} = u$$ — Integrator

If the nonlinear system satisfies certain assumption with z Є R as its control then

- The CLF

$$V_a(x,z) = V(x) + \frac{1}{2}[z - \alpha_s]^2$$

depicts the control input u

$$\boldsymbol{u = -c\big(z - \alpha_s(x)\big) + \frac{\partial \alpha}{\partial x}[f(x) + g(x)z] - \frac{\partial V}{\partial x}}$$

- renders the equilibrium point x=0, z=0 is GAS.

RESULTS

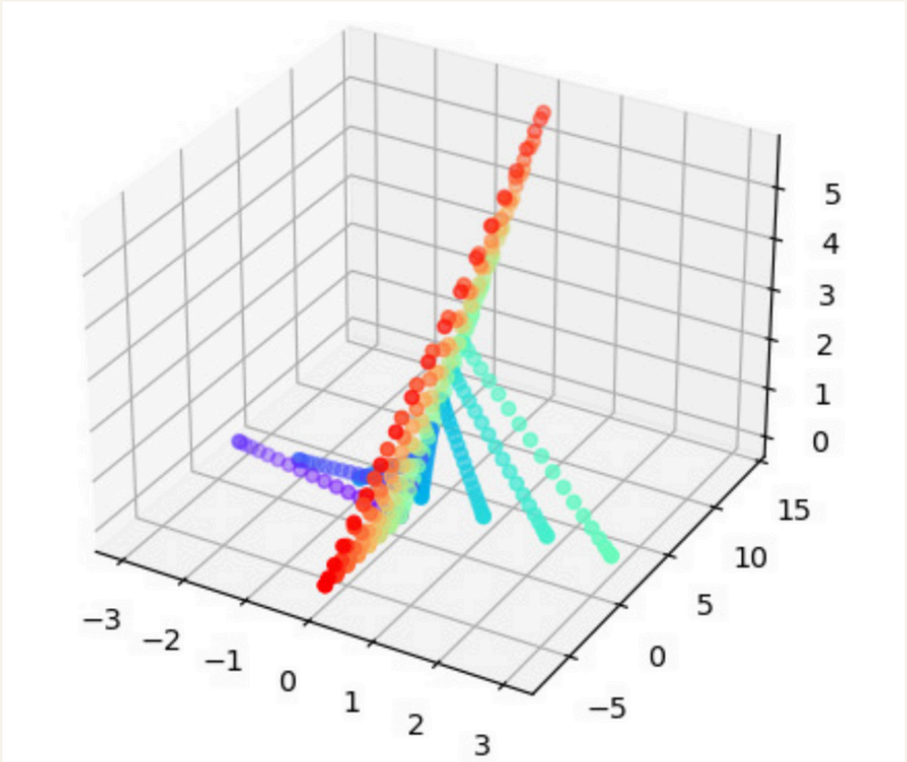
FORMATION OF LETTERS

- **We used 15 target coordinates.**

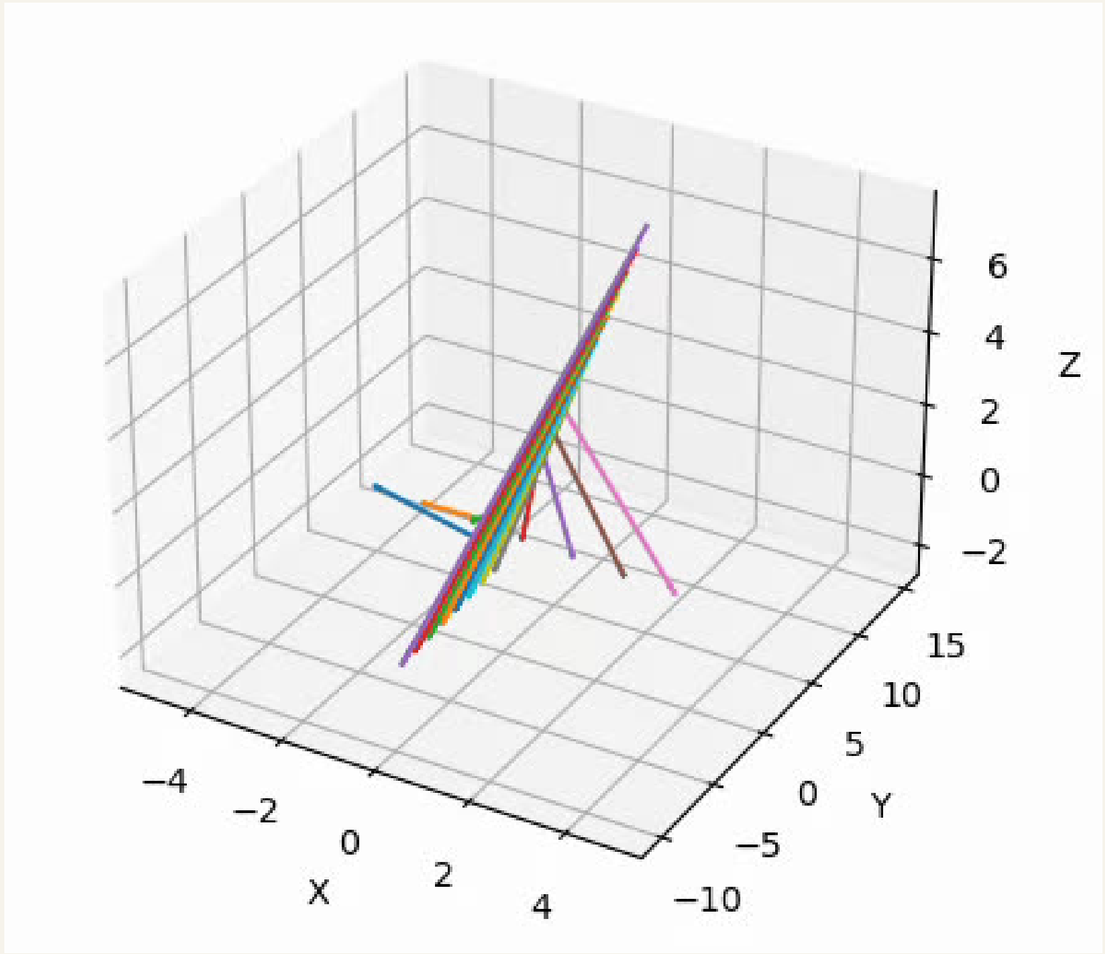
- **Initial positions that we gave was an array defined below:**

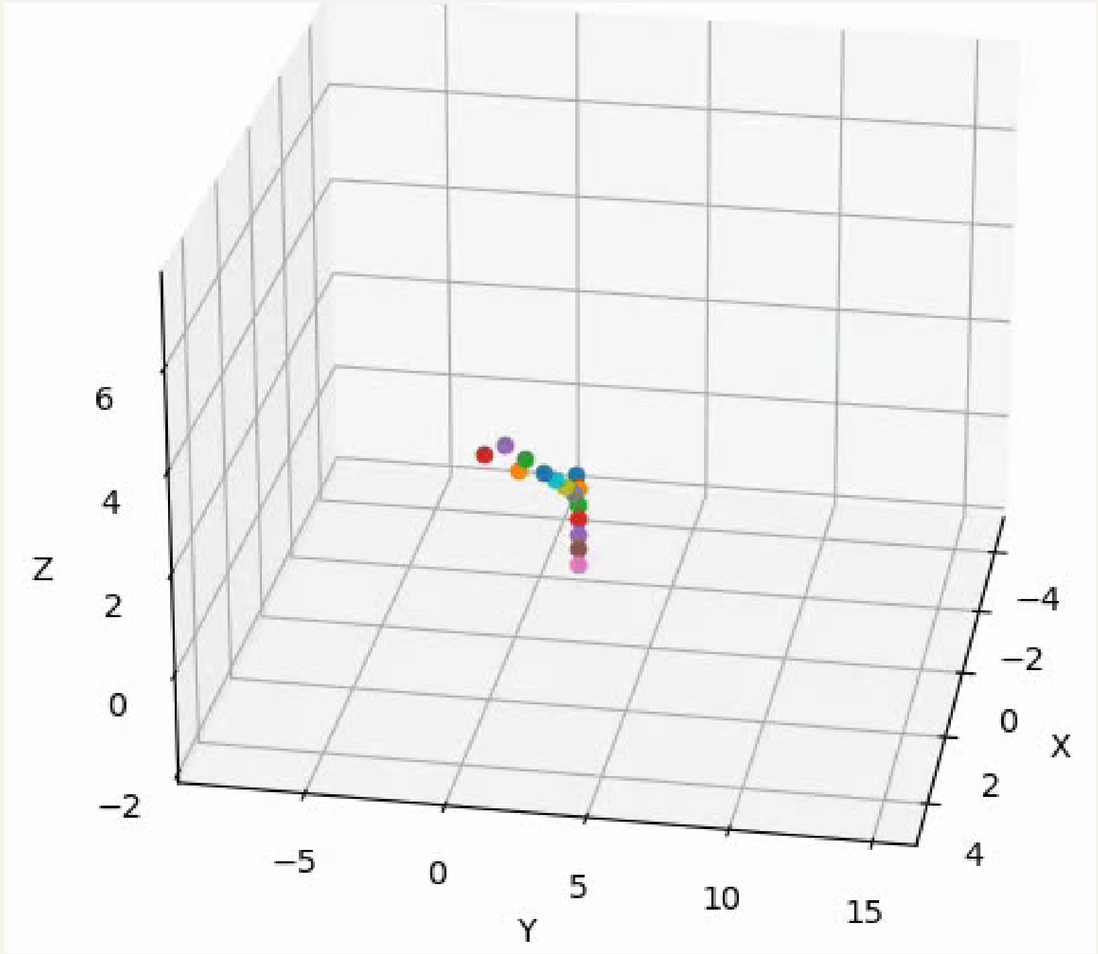
```
self.number_robots = 15
self.INIT_POSE = np.array([(0, i, 2*i/5) for i in range(self.number_robots)])
```

- **The final positions were given in 3-D coordinate system for each point**

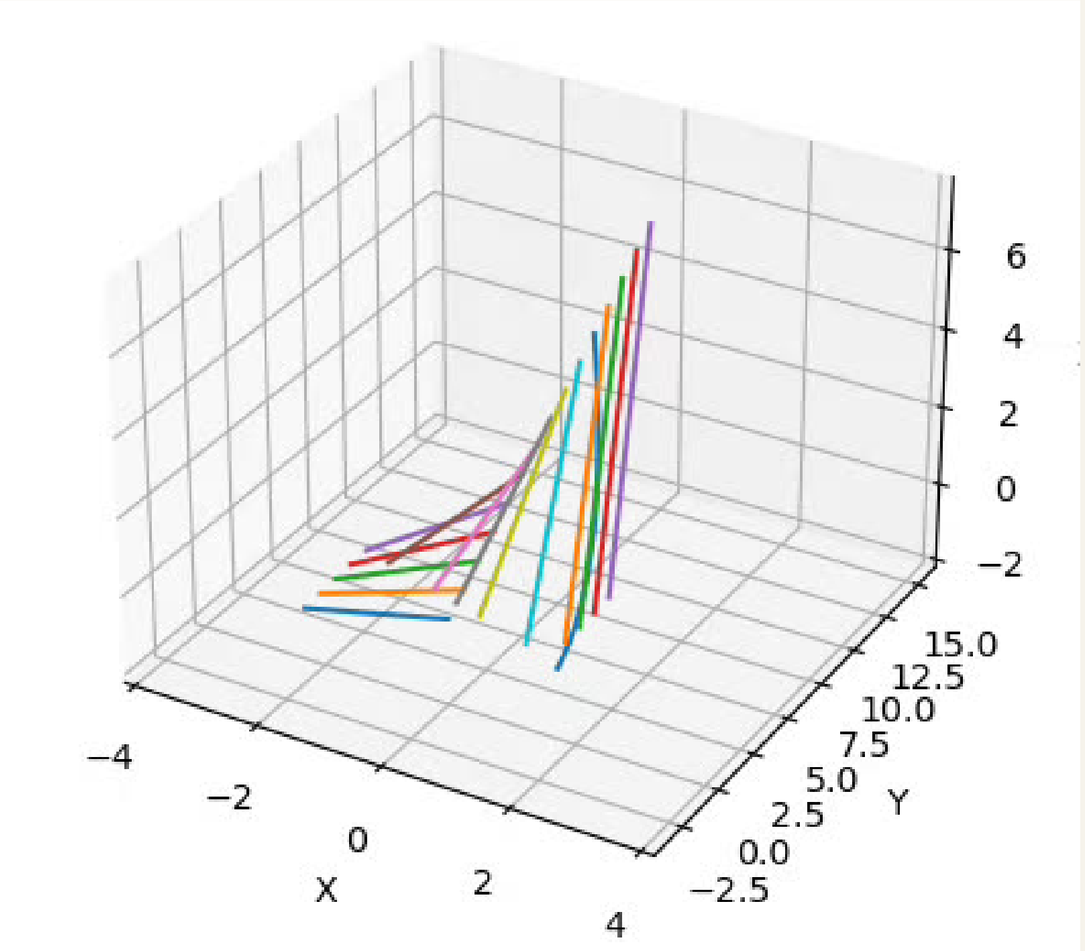


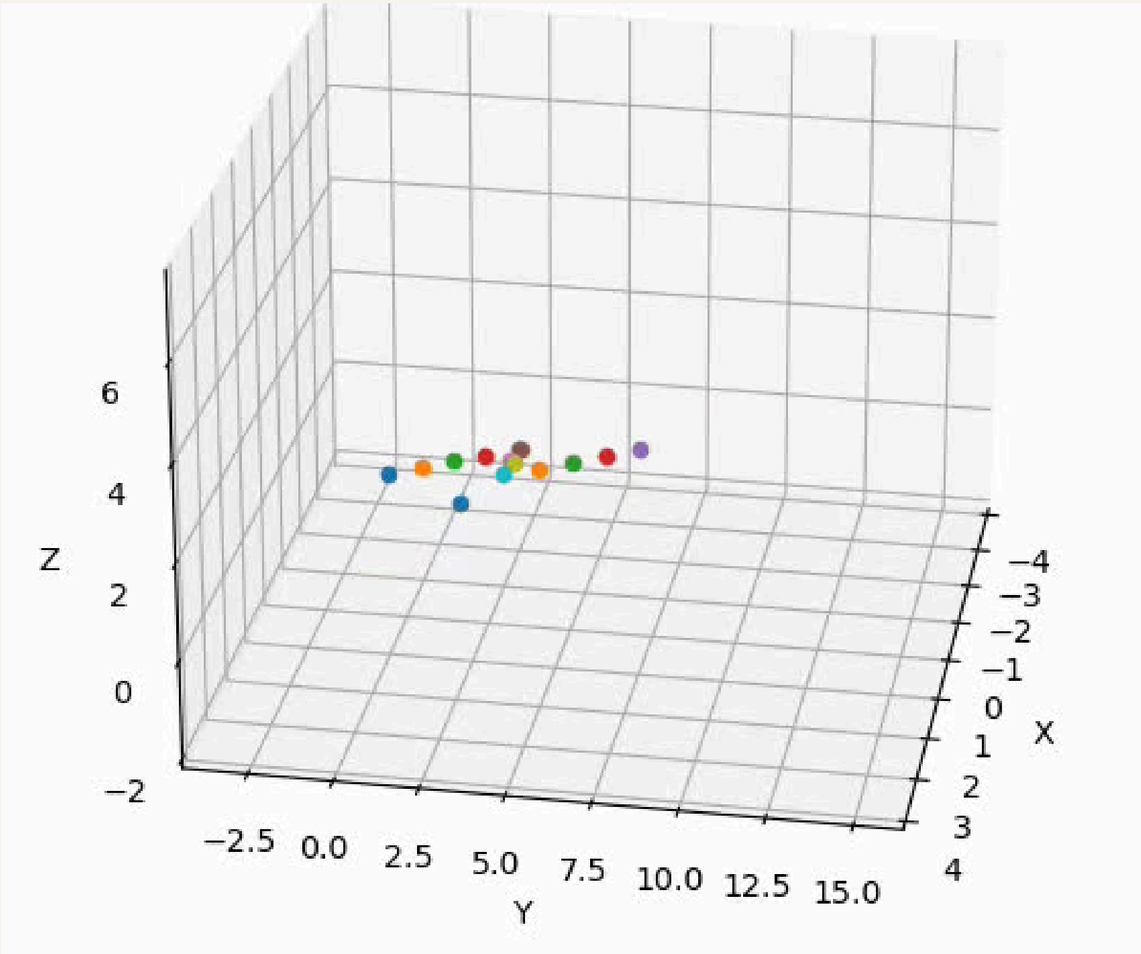
FORMATION OF LETTER ‘T’





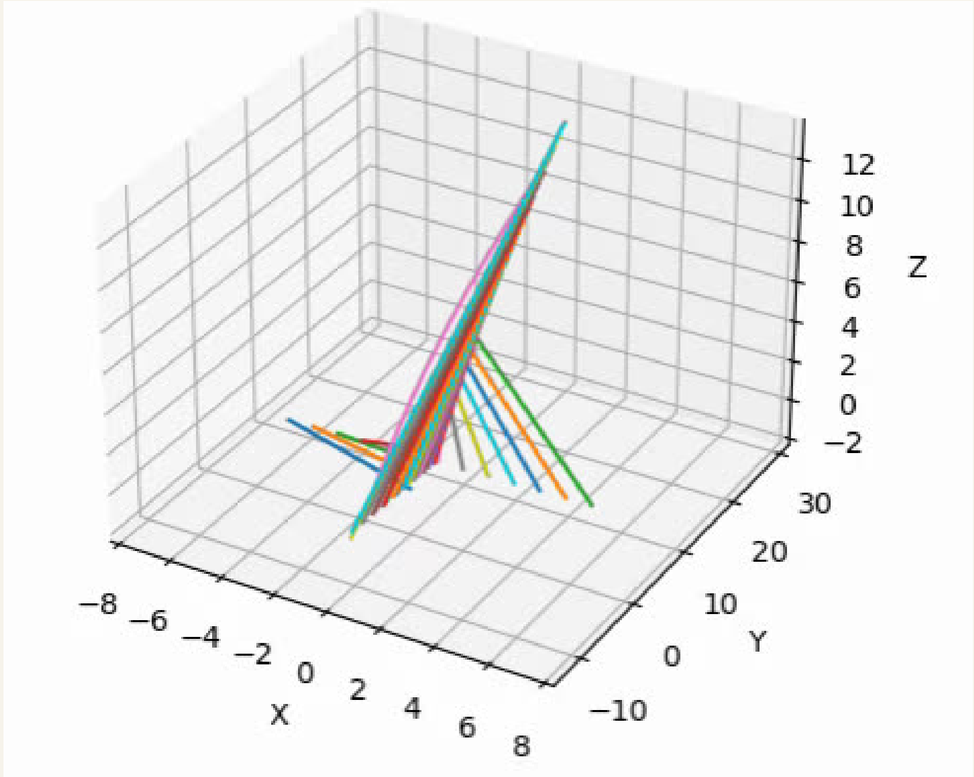
FORMATION OF LETTER ‘Z’

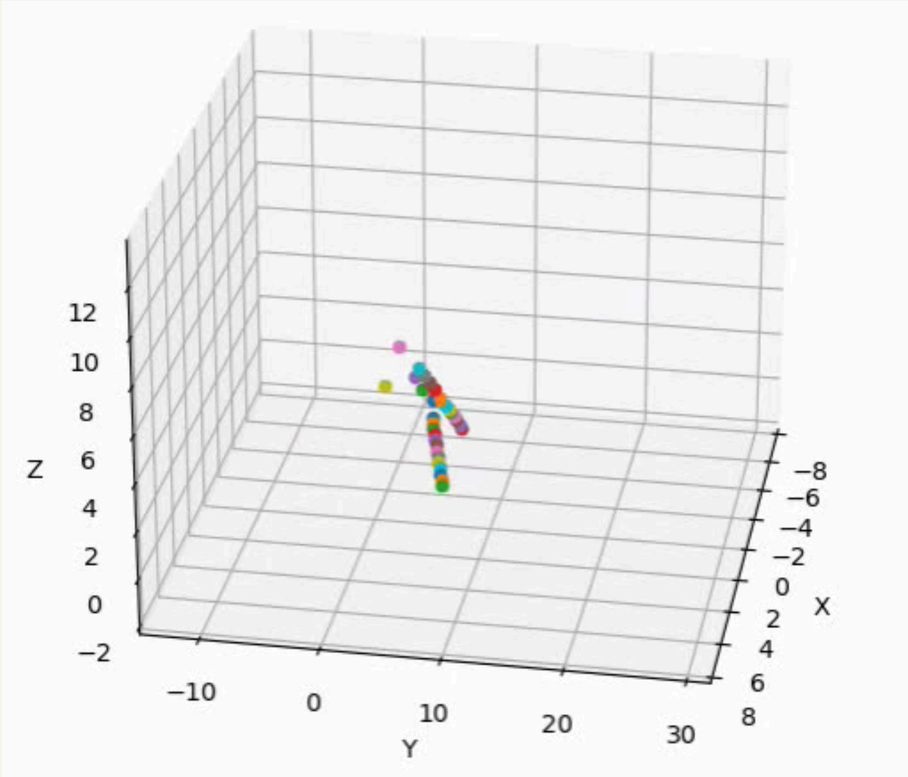




SCALING OF DRONES

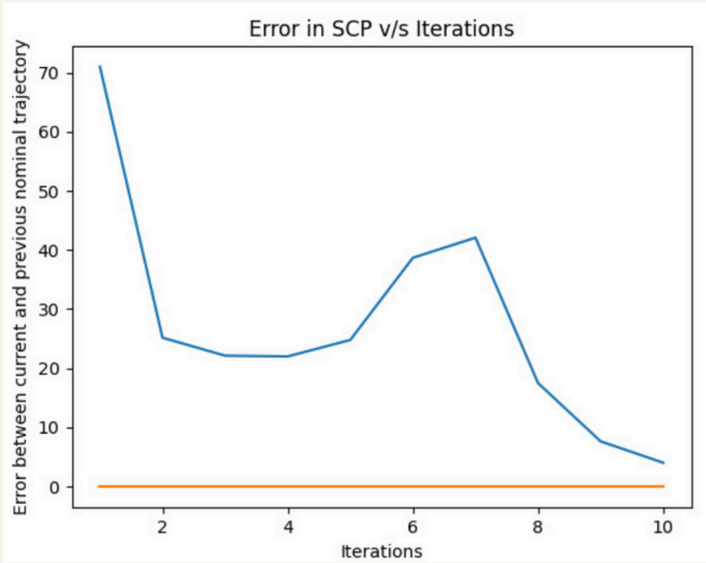
- **We also proceeded further with more number of drones and took 30 drones and target coordinates:**





Error vs Iterations in SCP

30 drones Letter T



15 drones Letter T

15 drones Letter Z

FURTHER DISCUSSION

Adding obstacle avoidance constraints in the swarm trajectory generation and ensuring safety for non-collision.

Priority of drones affects the generated trajectory , for strictly monotonically decreasing SCP error we need unbiased algorithms to reach most optimal assignment

IIT Bombay | 2024

THANK YOU

Presented By : Group 43