Localization, collision avoidance, and centroid navigation of parachutes swarm

Intelligent distributed systems

S. Manfredi N. Andreetta

University of Trento

27/09/2023

Outlines

1. Scope of the project
2. Adopted models
3. Solution
4. Results
5. Conclusions and future work

1 Scope of the project

2 Adopted models

3 Solution

4 Results

5 Conclusions and future work

Scope

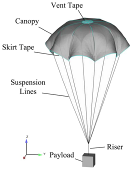
- Drive a swarm of parachute from a releasing point to the ground
- Focus on the centroid of the swarm rather than the single chute
- Each parachute has to:
  - Localize itself (and possibly the others)
  - Estimate the swarm centroid
  - Identify a centroid trajectory and move to fulfil it
  - Avoid collision with others

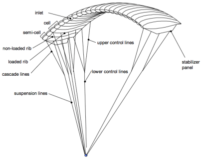
1. Scope of the project
2. Adopted models
3. Solution
4. Results
5. Conclusions and future work

Parachute models

Two models considered for movement on the horizontal plane:

1. Simplest $\Rightarrow$ Linear with actuation in all the directions
2. More realistic $\Rightarrow$ Unicycle like





In vertical direction:

- Gravity acts as non-controllable input
- Chute can not fall at any speed nor can stay still $\Rightarrow$ boundaries in the velocity
- Max falling velocity changes if the sail is open or close

Wind acts as noise in the three directions

Communication and measurement systems

- Communication requires bi-directionality $\Rightarrow$ UWB
- Measurements:
  - Absolute position $\Rightarrow$ GPS (5 m uncertainty)
  - Absolute orientation $\Rightarrow$ Compass (1° uncertainty)
  - Relative position $\Rightarrow$ Stereo camera, Lidar, Camera+UWB (1 m uncertainty)

The probability of having the measurements is taken into account

1 Scope of the project

2 Adopted models

3 Solution

4 Results

5 Conclusions and future work

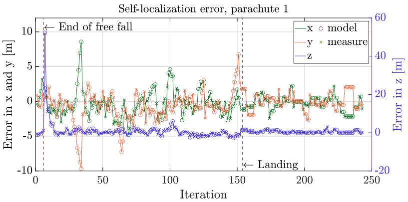
Structure of the solution

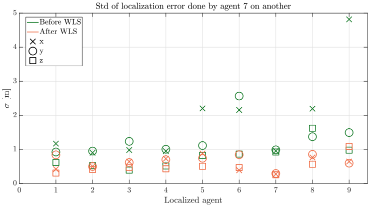
1. Propagation of the dynamic
2. Localization
3. Distribute information
4. Estimation of the global centroid
5. Collision avoidance (Voronoi tessellation)
6. High level motion control
7. Low level motion control

Localization

Each agent localizes itself and the others with:

1. Self-localization via KF/EKF with absolute measure
2. Measure the position of others
3. WLS with others in communication
4. Discharge the consensus reached on themselves



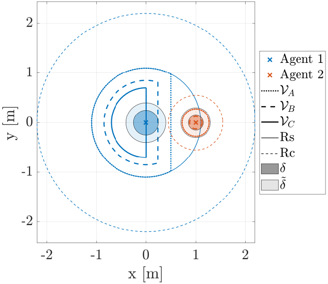


Collision avoidance

Main idea: Voronoi tessellation *2D+z* & remain inside

Each parachute considers:

- Parachutes close in the horizontal plane ⇒ Local information
- Parachutes that are below itself ⇒ Brake in z-direction
- Its encumbrance and finite velocity
- Error in the self localization ⇒ Increase its encumbrance
- Error in the localization of the others ⇒ Move them closer



High-level motion control

- Goal: how and where to move the centroid to make it land
- Centroid modelled as a fictional parachute:
$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} I_2 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \Delta t \begin{bmatrix} I_2 \end{bmatrix} \begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix}$$
  - LQR algorithm: $u_t = K_t x_t$
- Computation of each parachute's desired position:
  - Same control for all the parachutes
  - Inverse Kinematics: postural task concept from robotics's field
- Emergency case: different target point when a parachute pops up from the bottom

Low-level motion control

- Linear model:

$$u = -k_p \left(p_i - C_{\mathcal{V}_i} \right)$$

$$C_{\mathcal{V}_i} = \frac{1}{M_{\mathcal{V}_i}} \int_{\mathcal{V}_i} \varphi\left(q\right) q dq \ , \ M_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \varphi\left(q\right) dq$$

  - Straight motion: fully actuated in xy plane
  - Vertical control:

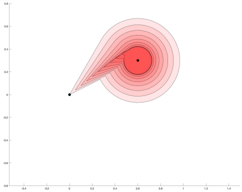
$$k_{pz} = -\bar{v}_z / R_{sv}$$

$$u_{tmp} = -\bar{v}_z - k_{pz}(z_i - z_{min,i})$$

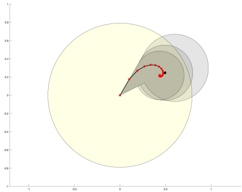
$$v_z = \min(u_{tmp}, v_{z,min} - \bar{v}_z)$$

Low-level motion control

- Non-linear model: unicycle-like control
  - Forward and angular velocity, breaks
  - Suppose zero minimum velocity
  - Non-straight motion: use of Motion Predicted Area



Truncated Ice-Cream Motion Cone updating as the agent moves.



The cone is moved to be fit into the Voronoi cell.

1 Scope of the project

2 Adopted models

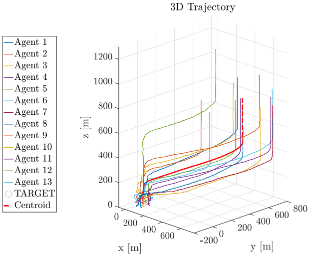
3 Solution

4 Results

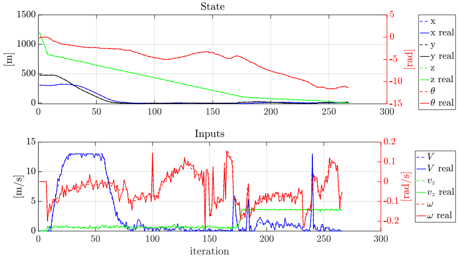
5 Conclusions and future work

Complete simulation

Trajectory and results of a simulation

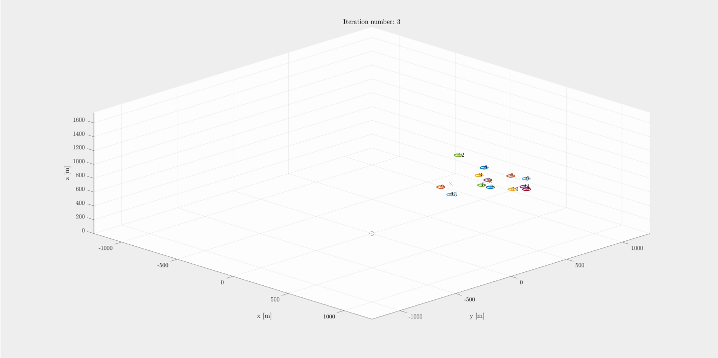


3D trajectory with 13 parachutes and all probabilities set to 1.



States and inputs behaviour of agent 12.

Complete Simulation



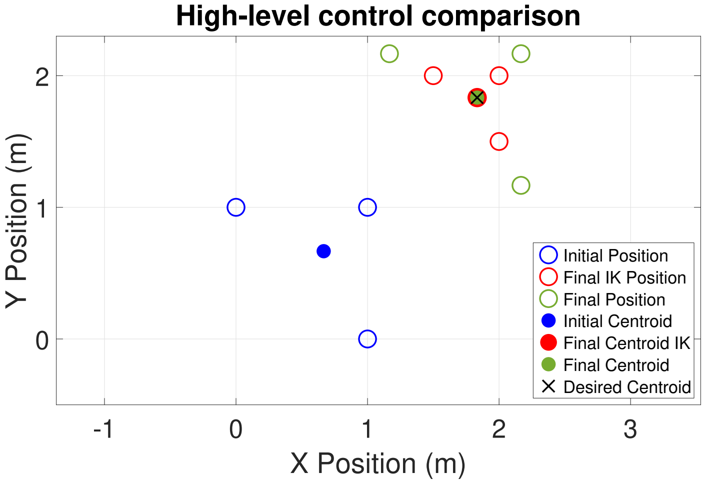
Effect of inverse kinematics

- Comparative test with both models, same initial conditions
- No uncertainties nor noises
- IK reduces the dispersion of the chutes around the target
- Helps to keep network connectivity

Comparison between the use of IK or not

#	Linear				Non-linear			
	Dist. [m]		RMS [m]		Dist. [m]		RMS [m]	
	IK	No IK	IK	No IK	IK	No IK	IK	No IK
1	2.99	0.01	2.99	0.01	3.00	0.02	3.00	0.02
3	6.21	3.05	25.36	33.34	7.72	4.42	25.36	27.74
5	4.32	6.13	29.41	38.76	6.86	5.61	32.84	39.83
7	3.07	4.86	35.20	41.88	6.30	4.57	32.62	47.31
9	5.12	5.47	37.56	49.35	7.21	8.64	41.49	48.43
11	2.31	2.53	43.18	53.43	8.83	5.90	43.44	55.80
13	4.70	3.47	46.37	58.23	3.42	7.52	46.91	57.78

Effect of inverse kinematics



1 Scope of the project

2 Adopted models

3 Solution

4 Results

5 Conclusions and future work

Conclusions

Results:

- Optimal control $\Rightarrow$ Centroid of the swarm moves correctly
- KF/EKF + WLS $\Rightarrow$ Localization
- Voronoi tessellation $\Rightarrow$ Collisions avoided
- Use of postural task $\Rightarrow$ Parachute close to target

Future improvements:

- More advanced parachute dynamics
- Localization with more sophisticated methods
- More advanced controls