

Formation Control and Path Planning of Multi-robot Systems via Large Language Models

Dong XUE^{1*}, Xuanjie ZHOU¹, Ming WANG¹ & Fangzhou Liu^{2*}

¹*Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education,
East China University of Science and Technology,
Shanghai 200237, China;*

²*Research Institute of Intelligent Control and Systems, School of Astronautics, Harbin Institute of Technology,
Harbin 150001, China*

Appendix A Prompt

This part contains the prompt involved in this experiment, which will be used for the input to the model and is the key content to realize the decision of LLM.

System Prompt

Role

You are the decision maker of a multi-robot system. You have all the information so far and will provide decision-making instructions for the next cooperative motion to track the target point.

Context

In this space, multiple small quadrotor UAVs are deployed, each with an arm length of about 0.12 units. To ensure safety, a safety zone with a radius of 0.5 units needs to be maintained around the center of the UAV fuselage. At the same time, to avoid collisions, these UAVs should keep at least 1.5 units away from obstacles floating in space and a safe distance from each other while moving. In this way, the UAVs can ensure a safe distance between each other while performing their missions, and also avoid collisions with obstacles.

Motion setting

1. It is ensured that each UAV can be as close as possible to its intended formation node to keep the formation compact and stable.
2. The distance cost during the whole system movement is considered, which is the total energy or time cost required for the UAV to move and maintain the formation.
3. A one-to-one correspondence between UAVs and formation nodes is established to ensure that each UAV has a clear position and responsibility.
4. Special attention is paid to the inside of the formation, where no collisions between UAVs occur, ensuring safety and reliability during flight.
5. It is ensured that the generated motion path does not collide with obstacles and maintains a certain safe distance.

Workflow

1. Make decisions for the whole team based on <Current Information> and <Goals>.
2. Refer to the thinking process of <Task Examples>, select the appropriate coordinate points and formation structure to meet the requirements of the target.
3. The final generated content should meet the requirements of <Specifications>, if not, modify to meet the requirements.

* Corresponding author (email: dong.xue@ecust.edu.cn, fangzhou.liu@hit.edu.cn)

Output Prompt

(Output Prompt 1)

Specifications

Follow the JSON format below to provide your answer. (Note that the JSON output does not contain any comments of any kind !)

```
'''json{
  "Motion_control": [{
    "Robot_name": "Alice",
    "Current_point": (2.0, 1.0, 3.0),
    "Next_point": (3.0, 2.0, 3.0)
  }],
  {
    "Robot_name": "Bob",
    "Current_point": (1.0, 2.0, 1.0),
    "Next_point": (2.0, 2.0, 2.0)
  }],
  {
    "Robot_name": "Charlie",
    "Current_point": (3.0, 0.0, 2.0),
    "Next_point": (3.0, 1.0, 2.0)
  }
}],
  "Reason": "xxx"
}
json'''
```

(Output Prompt 2)

Specifications

Follow the JSON format below to provide your answer. (Note that the json output does not contain any comments of any kind !)

```
'''json{
  "Motion_control": {
    "Current_point": (2.0, 1.0, 3.0),
    "Next_point": (3.0, 2.0, 3.0)
  },
  "Reason": "xxx"
}
json'''
```

Initialize Formation Prompt

Current Information

- Current robot information(name, coordinates, velocity):{}
- Formation target :{}

Goal

Specify {} as the leader and all other members as followers. Generate the next position of the leader and followers, and gradually move towards the unitormation target.

Constraints

- Select the point on the line from the leader to the Formation target, 1 units Euclidean distance units away from the leader_current, as the leader_next.
- leader_next is no more than 1 unit away from leader_current, and there is a forward displacement in all three directions.
- Disallow the selection of coordinates associated with the Formation target to be introduced into leader_next.
- All robots together form a <{}> formation structure with the same z-height.
- Firstly, the next position of the leader is determined, and then the next coordinate is generated for all robots according to the <Formation Rule> .
- The output node coordinates should be a certain value, and no formula should be used.

Formation Shape Prompt

Formation Rule

Follower1: The first follower will be positioned at a new location obtained by subtracting 1 unit from the x-coordinate and subtracting 0.5 unit from the y-coordinate of leader_next. The z coordinate remains the same.

Follower2: The second follower will be positioned at a new location obtained by adding 1 unit to the x-coordinate and subtracting 0.5 unit from the y-coordinate of leader_next. The z coordinate remains the same.

Follower3: The third follower will be positioned at a new location obtained by subtracting 1 unit from the y-coordinate of leader_next. The x and z coordinates remain the same.

Task Examples

If leader_current is (0, 0, 0):

leader_next is (3, 3, 3).

follower1 will be (2, 2.5, 3).

follower2 will be (4, 2.5, 3).

follower3 will be (3, 2, 3).

Next step coordinates Prompt

Current Information

- Uavs Center :{}

- Formation target :{}

- Obstacles(Position and number):{}

Goal

Choose the next movement position for the Uavs Center, and approach the Formation target as fast as possible.

Constraints

- The midpoint on the line between the current coordinate and the formation target is selected as the new coordinate for the next step.

Task Examples

The coordinate of the Leader is (0, 1, 0) and the Formation target is (0, 8, 0). Directly select (0, 4, 0) as the next motion target.

Go around Prompt

Current Information

- Uavs Center :{}

- Formation Target :{}

- Team Speed{}

- Obstacles(Position and number):{}

Goal

Select the next position on the way to the Formation Target to ensure that the movement path always keeps a safe distance of 3 from the obstacle.

Constraints

- For the Uavs center's next point, select a value that is 3 units higher than the average z-axis height of the obstacles as the height for the Uavs center's z-axis. The x and y coordinates remain the same.

- If the z-axis height of the formation center has already surpassed the average z-axis height of the obstacles by 2-3 units, then the formation target should be directly chosen as the next point for the center point.

Task Examples

Uavs Center (0,1,0), Formation Target (0,5,0), obstacle (0,2,0), (0,3,0). Select the next step (0,2,3) and move from above the obstacle.

Delay Predict Prompt

Current Information

- Robot Information (name, coordinates, velocity): {}

Goal

Estimate the delay between the leader and followers based on their current positions and velocities.

Constraints

- Choose {} as the team leader.
- Predict the current information of the follower based on the previous information of the follower provided.
- Use the predicted current leader and follower information to predict leader-follower latency according to the <Delay rule> .
- Provide a single, deterministic delay value without multiple possibilities.

Delay rule

lf1 = Euclidean distance between the leader and follower1

lf2 = Euclidean distance between the leader and follower2

lf3 = Euclidean distance between the leader and follower3

total_distance = lf1 + lf2 + lf3

$\lambda = \text{total_distance} * 1$

C_delay = 0.08

Delay = C_delay * Poisson_distribution(λ , 1)

Task Examples

leader info: ['Alice', [1.88, 2.87, 0.87], [0.1, 0.16, 0.02]]

follower1 info: ['Bob', [0.82, 2.78, 0.91], [0.13, 0.18, -0.04]]

follower2 info: ['Charlie', [1.91, 1.86, 0.8], [0.07, 0.11, 0.01]]

follower3 info: ['David', [0.85, 1.8, 0.78], [0.11, 0.17, 0.02]]

Plug in the <Delay Rule> formula to calculate the Poisson delay is 0.32

Specifications

Follow the JSON format below to provide your answer. (Note that the JSON output does not contain any comments of any kind !)

```
'''json
```

```
{
  "Delay": "xxx"
}
```

```
json'''
```

Workflow

1. Make decisions for the whole team based on <Current Information> and <Goals> .
2. Use <Constraints> to guide calculations and ensure all conditions are met.
3. The final generated content should meet the requirements of <Specifications>. If not, modify it to ensure all requirements are satisfied.
4. The detailed prediction calculation process is given to reduce the text explanation.

Approach Target Prompt

Current Information

- Uavs Center :{}

- Formation target: {}

Goal

The target point is directly selected as the Formation target to move quickly.

Task Examples

The coordinate of the Uavs Center is (1, 1, 1) and the Formation target is (7, 6, 8). Directly select (7, 6, 8) as the next motion target.

Adjust formation Prompt**# Current Information #**

- Robot Information (name, coordinates, velocity): {}
- Formation Target :{}
- Detected Obstacles list :{}

Goal

- Choosing the next step position for each UAV individually, both choose to move towards the formation target, and adjust the Angle to avoid collision if an obstacle is found.

Constraints

- Select the next point that is 3-4 units away from the current point.
- The relative structure between the strata is not considered and there is no need to maintain the stability of the strata.

Task Examples

The target center coordinates is (3, 3, 3). The current coordinates of the leader are (1, 1, 0), and there is an obstacle at (2, 1, 2). The followers, currently at (0, 0, 1) and (0, 1, 0), can temporarily move to (1, 2, 0) and (0, 0, 0) to avoid the obstacle. After bypassing the obstacle, they will reform the triangular structure near the target center.

Appendix B Algorithm**Algorithm B1** Prompt Selection

Input: UAV information, obstacle information, target point information

```

if UAV motion state == Initialization then
  | return Initialize Formation Prompt
else if UAV motion state == Keep moving forward then
  | if Current formation is close to formation target then
  |   | return Approach Target Prompt
  | else
  |   | return Next Step Coordinates Prompt
else if UAV motion state == Avoiding obstacles then
  | if Obstacle avoidance strategy == Circumnavigation then
  |   | return Go Around Prompt
  | else if Obstacle avoidance strategy == Adjusting formation then
  |   | return Adjust Formation Prompt
else if UAV motion state == Direct to Target Point then
  | return Approach Target Prompt

```

Algorithm B2 UAV Movement Algorithm**Input:** Update UAVs TargetsInitialize *Motion_result* $\leftarrow \emptyset$ Initialize *next_move_mode* $\leftarrow \text{None}$ Initialize *Obstacles_list* $\leftarrow \emptyset$ Initialize *one_round_time* $\leftarrow \Delta t$ **while** *one_round_time* > 0 **do** *Obstacles_list* $\leftarrow \{\text{Obstacles in the Perception Range of the UAV}\}$ Initialize *avoid_flag* $\leftarrow \text{True}$ **if** (*Obstacles in the obstacle list*) and (*not exist in the leader's record*) **then** *avoid_flag* $\leftarrow \text{False}$ **if** *avoid_flag* == *True* **then**

Control UAVs to Move Towards Their Respective Targets for 0.01s

if *one_round_time* ≤ 0 **then** *origin_result* $\leftarrow$ current UAV information *next_move_mode* $\leftarrow \text{SteadyMovement}$ **else** *origin_result* $\leftarrow$ current UAV information *next_move_mode* $\leftarrow \text{ObstacleAvoidance}$ *one_round_time* $\leftarrow \text{one_round_time} - 0.01s$ Update global information (*origin_result*, *next_move_mode*, *Obstacles_list*)**Algorithm B3** Movement Strategy**while** *True* **do**

LLM makes decisions and UAVs update the next step goal

while *True* **do** **if** *motion_state* == *steady_movement* **then** **if** *formation_stable* **then** **if** *distance(current_position, target_position)* > *target_distance_threshold* **then**

Execute Movement According to Algorithm C2

 Update *motion_state* **else** **Break** **else**

Execute Movement According to Algorithm C2

 Update *motion_state* **else if** *motion_state* == *obstacle_avoidance* **then** **Break**

Update current UAVs and environment information

if *reached_target_point* **then** **Break**

C Sample dialogue

System:

"# Role # xxx. # CONTEXT # xxx. # Motion setting # xxx. # Workflow # xxx."
(More details in the **System Prompt**)

User:

"# Current Information # - Current robot information(name, coordinates, velocity):[['Alice', (1.0, 2.0, 0.9), array([0, 0, 0])], ['Bob', (0.0, 1.2, 1.0), array([0, 0, 0])], ['Charlie', (1.3, 1.0, 0.5), array([0, 0, 0])], ['David', (0.2, 0.3, 0.3), array([0, 0, 0])]] - Formation target :[8.8, 9.4, 9.5]. # Goal # xxx. # Constraints # xxx. # Formation Rule # xxx. # Task Examples # xxx. # Specifications # xxx."
(More details in the **Initialize Formation Prompt**, **Formation Shape Prompt** and **Output Prompt 1**)

Assistant:

```
xxx"""json
{ "Motion_control": [ { "Robot_name": "Alice", "Current_point": (1.0, 2.0, 0.9), "Next_point": (2.0, 3.0, 0.9) },
{ "Robot_name": "Bob", "Current_point": (0.0, 1.2, 1.0), "Next_point": (1.5, 3.0, 0.9) }, { "Robot_name":
"Charlie", "Current_point": (1.3, 1.0, 0.5), "Next_point": (2.0, 2.5, 0.9) }, { "Robot_name": "David",
"Current_point": (0.2, 0.3, 0.3), "Next_point": (1.5, 2.5, 0.9) } ], "Reason": "The next points are calculated to
maintain a compact formation structure with a safety margin while moving towards the target formation point." }
xxx"""
```

User:

"# Current Information # - Uavs Center :[1.71, 2.7, 0.74] - Formation target :[8.8, 9.4, 9.5] - Obstacles(Position and number):[]. # Goals # xxx. # Constraints # -xxx. # Task Examples # xxx. # Specifications # xxx"
(More details in the **Next step coordinates Prompt** and **Output Prompt 2**)

Assistant:

```
xxx"""json { "Motion_control": { "Current_point": (1.71, 2.7, 0.74), "Next_point": (5.26, 6.05, 5.12) },
"Reason": "The next point is the midpoint between the current coordinate and the formation target, allowing for
efficient progression towards the target." }
xxx"""
```

• • •

User:

"# Current Information # - Uavs Center :[7.32, 8.1, 7.53] - Formation target:[8.8, 9.4, 9.5]. # Goals # xxx. # Task Examples # xxx. # Specifications # xxx."
(More details in the **Approach Target Prompt** and **Output Prompt 2**)

Assistant:

```
xxx"""json
{ "Motion_control": { "Current_point": (7.32, 8.1, 7.53), "Next_point": (8.8, 9.4, 9.5) }, "Reason": "The
formation target is directly selected as the next motion target to ensure rapid movement towards the intended
destination." }
xxx"""
```