Multi-Formation Planning and Coordination for Object Transportation

Weijian Zhang, Charlie Street, Masoumeh Mansouri

***Abstract*— Multi-robot formations have numerous applications, such as cooperative object transportation in smart warehouses. Here, robots must deliver objects in formation while avoiding intra- and inter-formation collisions. This requires solutions to multi-robot task assignment, formation generation, rigid formation maintenance, and route planning. In this paper, we present a cooperative multi-formation object transportation system which explicitly handles inter-formation collisions. For formation generation, we propose a distributed motion planning approach which combines artificial potential field methods and leader-follower based control. For formation planning, we present a heuristic search-based algorithm which uses convex segmentation techniques, and extend the minimum snap method to synthesise smooth trajectories while maintaining the formation. We also propose a variant of the dynamic window approach to avoid collisions between formations. We demonstrate the efficacy of our approach in simulation.**

I. Introduction

A common approach for large object transportation in robotics is to use multiple robots to collaboratively push, cage, or grasp the objects [1]. This paper addresses cooperative transportation problems where multiple formations carry objects on top of them (see Fig. 1), which we refer to as *multi-formation planning and coordination (MFPC)*. Here, the formations are *rigid*, i.e. the formation remains unchanged during transportation, which is necessary to balance heavy loads. To solve MFPC, we must generate and maintain effective formations until a goal location is reached while avoiding collisions with other formations and obstacles.

In this paper, we decompose MFPC into several subproblems (see Fig. 2). For formation generation, we assign robots to formations using the Hungarian algorithm [2]. We then use an artificial potential field (APF) method for navigation towards each robot's formation location. APF methods are a common distributed approach for formation control [3], but can become trapped in local minima, which prevents robots from reaching their target positions. To address this, we combine a wall-following strategy with consensus-based APF for robust formation generation.

After formation generation, each formation should move toward its destination while maintaining form and avoiding collisions with other formations and obstacles; this is the planning component of MFPC. For this, we employ a consensus-based method [4], where robots follow a virtual leader in the formation centre, and reach a consensus on velocity to maintain form [5]. The virtual leader should maintain the formation by guaranteeing a kinematically feasible path for each follower. To achieve this, we present a heuristic-based global path planner which computes optimal plans under the formation and kinematic constraints. We then synthesise smooth trajectories from the global path using an extended minimum snap approach [6]. To avoid inter-formation collisions, we propose a variant of the dynamic window approach (DWA) [7]; this is the coordination component of MFPC. Finally, we combine first-order consensus control [8] and pure pursuit control [9] to track robot trajectories and complete object transportation.

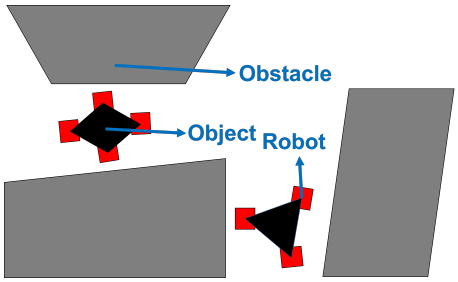


Fig. 1. Object transportation for multiple rigid formations.

The main contribution of this work is a comprehensive MFPC framework that integrates formation generation, planning, and coordination techniques. We demonstrate the performance of our MFPC system and the individual components empirically in simulation.

II. Related Work

Formation generation requires a team of robots to organise into a predefined shape [10]. Centralised approaches to formation generation can attain high performance, but scale exponentially in the number of robots, making them intractable for realistic problems [11].

In [12], an APF-based consensus control method is presented to formulate coordination and control strategies between robots without considering avoiding obstacles. An improved APF is proposed in [13] for path planning of a multi-robot formation which efficiently avoids getting trapped in local minima caused by obstacles but fails to address deadlocks among robots reaching their goal positions. An alternative decentralised approach is proposed in [14], which requires only the relative positions of robots and obstacles from each robot. This approach, however, does not guarantee

Weijian Zhang, Charlie Street and Masoumeh Mansouri are with the School of Computer Science at the University of Birmingham, wxz163@student.bham.ac.uk and {c.l.street, m.mansouri}@bham.ac.uk

Charlie Street and Masoumeh Mansouri are UK participants in Horizon Europe Project CONVINCE, and supported by UKRI grant number 10042096.

979-8-3503-0704-7/23/$31.00 ©2023 IEEE

2023 European Conference on Mobile Robots (ECMR) | 979-8-3503-0704-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ECMR59166.2023.10256314

solution convergence. A template-based technique is used in [15] where no experiments are provided regarding the scalability issue. In our work, we ensure achieving formation convergence, a deadlock-free and scalable solution by using the Hungarian algorithm [2] for optimal task assignment and the first-order consensus method for driving robots to their goal positions [8].

Formation planning has leveraged the wealth of classical AI search methods. For instance, in [16], a relaxed A^* planner is employed to generate an optimal collision-free global path over a map where obstacles are inflated by a circle that covers the entire footprint of the formation. However, such conservative approximations can result in a loss of precision and, at times, failure to find a solution. In this paper, we define the obstacle-free space without making any approximations of the obstacles. Another example includes constrained optimisation for non-rigid formation planning [17] in a dynamic environment, an approach which cannot be applied directly to the type of rigid MFPC considered in this paper. An alternative approach is to use sampling-based methods such as rapidly-exploring random trees [18] or probabilistic roadmaps [19] that consider the geometric constraints of the formation. However, these approaches rely on sampling a large number of configurations to find a path which often suffers from abrupt variations in direction. To alleviate this problem, we use heuristic search to reduce the number of sampling configurations and abrupt changes in direction. Further, we apply a modified minimum snap method [6] to produce a feasible smooth trajectory.

For rigid object transportation, formations should be maintained during locomotion. For instance, a leader-follower strategy combined with APF is used in [20]. The leader robot determines its navigation path through APF, and the other robots in the group follow the leader to maintain the formation using distance-angle $(l-\phi)$ control. However, dynamic obstacles are not considered. In [21], the authors demonstrate a hierarchical quadratic programming approach, such that the *a priori* unknown obstacles can be detected and avoided at runtime. However, due to the lack of a global planner to generate a set of waypoints, the control may run the risk of falling into local minima. In [22], an improved A* algorithm is adopted to generate optimal global paths and a multiple sub-target APF is proposed for local path planning of the formation. In this approach, when the formation avoids dynamic obstacles, the generated trajectory toward the local subgoal may not be executable for the robot. In our work, we benefit from the smooth and executable global trajectories generated by the formation planner. Further, we extend the dynamic window approach (DWA) [7] by introducing formation prioritization to achieve collision avoidance between formations while ensuring that the robot motion constraints are not violated.

III. Preliminaries

We begin by defining our assumptions over the workspace, robots, and formations.

Workspace. Let $W \subset \mathbb{R}^2$ be a 2D workspace which contains a set of static obstacles $O \subset W$, and let $F = W \setminus O$ denote the obstacle-free workspace.

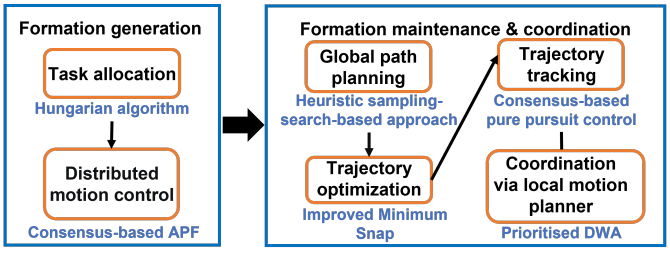


Fig. 2. The proposed problem decomposition for MFPC.

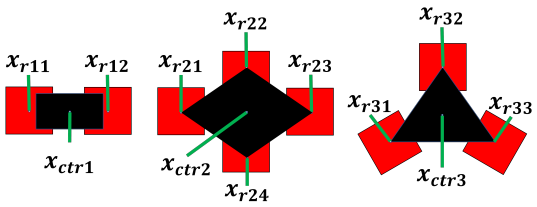


Fig. 3. Linear, rectangular, and triangular formations consisting of several robots (in red) and an object (in black). Each robot i is placed with its state $\boldsymbol{x}_{ril}$ relative to the jth formation's centroid state $\boldsymbol{x}_{ctrj}$.

Robots. Consider a team of homogeneous rectangular robots $R = \{1, ..., |R|\}$, where $|R|$ is the cardinality of R (see Fig. 3). Let $\boldsymbol{x_i}(t) = \langle x_i(t), y_i(t), \theta_i(t) \rangle$ be the state of robot $i \in R$ at time t, where $x_i(t), y_i(t) \in \mathbb{R}$ denote robot i's position at time t, and $\theta_i(t) \in [-\pi, \pi)$ denotes its orientation. We assume all robots are holonomic, and use a unicycle kinematic model:

$$\dot{\boldsymbol{x_i}}(t) = \frac{d}{dt}\begin{bmatrix} x_i(t) \\ y_i(t) \\ \theta_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \cdot cos\theta_i(t) \\ v_i(t) \cdot sin\theta_i(t) \\ \omega_i(t) \end{bmatrix} \quad i \in R, t \in [0, t_{f_i}], \tag{1}$$

where $v_i(t)$ and $\omega_i(t)$ denote the velocity and angular velocity applied at time t respectively, and t_{f_i} is the finite horizon for robot i. At each time step, we assume robots can observe the state of all other robots and obstacles within sensing range $d_0 \in \mathbb{R}_{\geq 0}$.

Formations. In this paper, we consider formations in straight lines, rectangles, and triangles (see Fig. 3). We write $z_j = < \boldsymbol{x}_{ctrj}, \boldsymbol{x}_{rj1}, ... \boldsymbol{x}_{rjk} >$ to denote the configuration of the jth formation, formed of k robots, where $\boldsymbol{x}_{ctrj}$ denotes the formation's centroid state, and $\boldsymbol{x}_{rjl}$ is the relative state of the lth robot with respect to $\boldsymbol{x}_{ctrj}$.

IV. Formation Generation

In this section, we present a distributed approach for formation generation which combines APFs with consensus-based leader-follower control and a wall-following strategy to avoid local minima and unreachable targets. The first sub-problem for formation generation is to allocate each robot i to a formation location $g_i \in G$, where g_i corresponds to a corner of a formation (see Fig. 3). For this, we use the optimal Hungarian algorithm [2] with a Euclidian distance cost function.

To drive robot i towards g_i while avoiding obstacles, we employ a modified APF approach [13]. Let $F_g(l_i(t))$ be the gravitational force towards the goal, where $l_i(t) =< x_i(t), y_i(t) >$ denotes the position of robot i at time t, and let $F_r(l_i(t))$ be the resultant repulsive force from the obstacles and other robots within robot i's sensing range. Under the APF, robot i's control input is given by $u_i(t) = F_g(l_i(t)) + F_r(l_i(t))$.

In cluttered environments, robots may become trapped in local minima and oscillate around their current location indefinitely. Local minima may be detected by observing a robot's state or the forces applied to a robot. Formally, a local minima occurs if either of the following hold at time t:

$$|F_g(l_i(t)) + F_r(l_i(t))| < \varepsilon \text{ or } |l_i(t-\tau) - l_i(t)| < \rho s, \quad (2)$$

where $\varepsilon \in \mathbb{R}_{>0}$ and $\rho \in [0,1]$ are small thresholds, and s is the distance travelled by robot i in the past τ timesteps. The first inequality holds if the gravitational and repulsive forces are approximately zero. The second inequality holds if the robot's position has changed less than the distance it has travelled. When a local minima is detected, we remove the gravitational and repuslive forces and drive the robot along the wall of the nearest obstacle towards the formation location. The gravitational and repulsive forces are reapplied once the inequalities in (2) are violated.

Local minima may also occur due to robot deadlocks. For example, a robot may be blocked from its formation location by the repulsive forces of robots who have already reached theirs. For this, we combine first-order consensus [8] and leader-follower control [23]. Here, the blocked robot becomes the leader, and the other robots become followers. First-order consensus control ensures the robots satisfy the relative position relationships in the formation and converge to velocity consistency, thus driving them toward the target formation. Let $a_{ij} = 1$ if robot i can communicate with robot j, and 0 otherwise. Then, let $N_i = \{j \mid a_{ij} = 1\}$, i.e. the robots that robot i can directly communicate with. Under first-order consensus control [8], the control input of the leader robot i is set to be:

$$u_i(t) = F_g(l_i(t)) + F_r(l_i(t)) + \sum_{f \in N_i} w_{if}(t)(\boldsymbol{x}_{if}(t) - r_{if}), \quad (3)$$

where r_{if} is the required relative distance between the leader i and follower f, and $\boldsymbol{x}_{if}$ is the state of the follower f with respect to the leader i. The weight $w_{if}(t)$ is given by $w_{if}(t) = 2 - e^{-(x_{if}(t) - r_{if})^2}$. Similarly, the control input for follower f is as follows, where $u_i(t)$ is the control input of the leader:

$$u_f(t) = u_i(t) + F_r(l_f(t)) + \sum_{f' \in N_j} w_{ff'}(x_{ff'}(t) - r_{ff'}). \quad (4)$$

V. Formation Planning

In this section, we synthesise a discrete global path T for the centre of each formation which minimises the distance from its initial configuration z_s to the goal configuration z_g. For this, we incrementally construct a graph $G = (V, E)$ over the workspace, where each node $v \in V$ lies inside a convex polygon P, and edges $(v, v') \in E$ connect nodes. We begin with a heuristic approach for partitioning the workspace into convex polygons (Subsection V-A), which we then use for global path planning (Subsection V-B).

A. Heuristic Workspace Decomposition

The obstacle-free workspace F can be partitioned into a set of convex polygons P which maintain the formation's geometric constraints, defined as:

$$A_n v_m \leq b_n, \forall n = 1, ..., N, m = 1, ..., M, \quad (5)$$

where A_n and b_m are the parameters of the separated hyperplanes, N is the number of sides of the polygon, and v_m represents the outer vertices of the formation's convex hull. In Alg. 1, we show how to generate the next convex polygon P in a partition, given the existing polygons $\mathbb{P} = \{P_1, ..., P_K\}$. For this, we use IRIS [24], which given an initial seed point alternately solves two convex optimisations: (1) finding a set of hyperplanes that separate an ellipse e from the obstacles O via quadratic optimization, and (2) finding the largest ellipse within the polygon P via a semi-definite program. Each convex polygon P has a corresponding ellipse e; we denote the set of existing ellipses as $\mathbb{E} = \{e_1, ..., e_K\}$.

For efficient global planning (see Subsection V-B), polygon construction should be guided towards the goal. Therefore, in Alg. 1, we introduce a heuristic strategy for sampling new IRIS seed points. First, we discretise the workspace into cells, and assign a cost to each cell in a matrix B (lines 2-6). Cells in obstacles or existing ellipses have cost $-\infty$, and all other cells have cost:

$$Cost(c) = \frac{1}{K} \sum_{k=1}^{K} ||c - e_k||_2 - \gamma^{(1-\zeta)} ||c - g||_2. \quad (6)$$

Here, $\gamma \in \mathbb{R}_{>0}$ is a weight parameter, $\zeta \in [0,1]$ is the proportion of the map covered by obstacles and ellipses, and g is the cell containing goal configuration z_g. Intuitively, (6) assigns high cost to cells which are further from existing polygons and closer to the goal. However, the second term in (6) decays as the explored area increases, which admits backtracking to explore regions further from the goal. We select the cell c^* with maximum cost in B as the new seed point for IRIS (lines 9-10), pushing the polygons towards the goal. After running IRIS, we test whether the new polygon P_{new} and ellipse e_{new} should be accepted into the partition (lines 11-16). Polygon P_{new} is rejected if the seed point c^* is less than distance α from a previous seed point, or if e_{new} is less than distance β from an existing ellipse $e_k \in \mathbb{E}$. If P_{new} is rejected, Alg. 1 must be re-run. Thresholds α and β decay as the workspace is explored to relax the conditions for polygon acceptance, as the distance to the nearest cell or ellipse will decrease as more polygons are added.

B. Global Path Planning

Using Alg. 1, we incrementally construct a graph $G = (V, E)$ by adding convex polygons to the workspace until a path exists between z_s and z_g, where nodes are added for

Algorithm 1: Heuristic IRIS

Input: Obstacle set O, existing polygons $\mathbb{P}$ and ellipses $\mathbb{E}$, previously selected cells $\mathbb{C}$
Output: P_{new}

$P_{new} \leftarrow \varnothing$
foreach $c \in W$ **do**
  if $c \in O$ *or* $c \in \mathbb{E}$ **then**
    $B\{c\} \leftarrow -\infty$
  else
    $B\{c\} \leftarrow \text{Cost}(c)$

while $P_{new} = \varnothing$ **do**
  $c^* \leftarrow \arg\max_{c \in W} B$
  $\{P_{new}, e_{new}\} \leftarrow \text{IRIS}(c^*)$
  foreach $c \in \mathbb{C}$, $e_k \in \mathbb{E}$ **do**
    if $\|c^* - c\| < \alpha$ *or* $\|e_{new} - e_k\| < \beta$ **then**
      foreach $c \in e_{new}$ **do**
        $B\{c\} \leftarrow -\infty$
      $\{\alpha, \beta\} \leftarrow \text{UpdateThreshold}(\alpha, \beta, B)$
      $P_{new} \leftarrow \varnothing$

$\mathbb{P} \leftarrow \mathbb{P} \cup P_{new}$, $\mathbb{E} \leftarrow \mathbb{E} \cup e_{new}$, $\mathbb{C} \leftarrow \mathbb{C} \cup c^*$
return P_{new}

each polygon and polygon intersection. We then synthesise a path T over graph G. In detail, we do the following:

1) **Initialise Graph:** Graph G is initialised with nodes at z_s and z_g. Convex polygons are generated from each of these points using IRIS [24]. If these polygons intersect, z_s, z_g, and an intersection point are connected as in step 4.
2) **Test for Path:** If a path exists from z_s to z_g in G go to step 5, else go to step 3. If the total area covered by polygons exceeds a threshold, terminate without a plan.
3) **Add New Polygon:** Generate a new polygon P_{new} using Alg. 1. Then, add a node z_{new} to G at the location within P_{new} closest to the goal z_g. Formally:
$$z_{new} = \arg\min_{z \in P_{new}} (z - z_g)^2. \tag{7}$$
4) **Add Intersection Nodes and Edges:** Find all existing polygons $P_k \in \mathbb{P}$ who intersect with P_{new}. Given the nodes z_{new} and z_k computed from P_{new} and P_k respectively using (7), find the configuration z_{inter} which minimises the perpendicular distance from the straight line between z_{new} and z_k. Formally:
$$z_{inter} = \arg\min_{z \in P_{new} \cap P_k} dis(z, line(z_{new}, z_k))^2. \tag{8}$$
Next, add the node z_{inter} and edges (z_{new}, z_{inter}) and (z_{inter}, z_k) to G. By optimising z_{inter} using (8), we minimise edge length. Following this, return to step 2.
5) **Find Shortest Path:** Synthesise the shortest path from z_s to z_g on graph G using A* search [25].

We demonstrate our global planner in Fig. 4. In Fig. 4(a), a triangular formation must reach the top right of the map. Fig. 4(b) shows the convex polygons generated from the initial and goal configurations. These regions do not intersect, and so we use Alg. 1 to add a new polygon (see Fig. 4(c)). The intersections between these three polygons admit a graph which connects the initial configuration to the goal configuration, and so a global path can be found (see Fig. 4(d)).

VI. Trajectory Generation and Formation Control

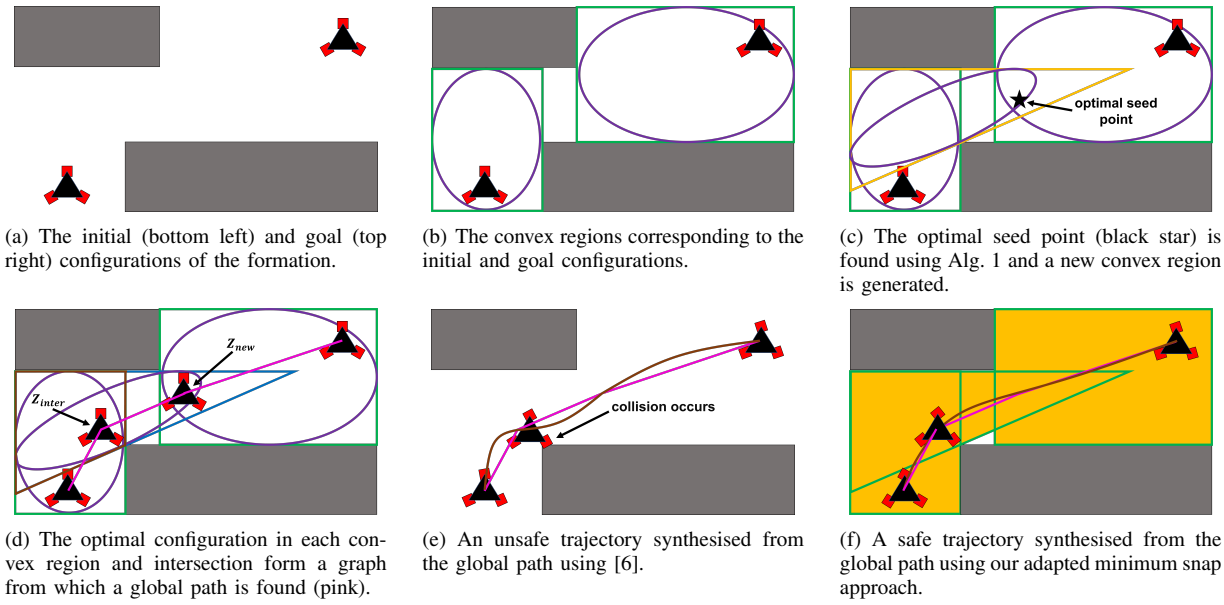
The global path T computed in Subsection V-B contains sharp turns which cannot be executed smoothly by a robot. To mitigate this, we now present techniques for trajectory optimisation, trajectory tracking, and local motion planning.

A. Trajectory Optimization

To synthesise continuous collision-free trajectories from the global path T which respect robot kinematic constraints, we consider an extended minimum snap approach [6]. Minimum snap cannot be applied directly to formations, as the geometric constraints of the formation are ignored, which may cause collisions (see Fig. 4(e)). To avoid collisions, we apply the inequality constraints in (5), and minimise the deviation from global path T.

In minimum snap, trajectories are represented in S segments. The rth trajectory segment is represented as an order q polynomial over the current time t:
$$p_r(t) = [1, t, t^2, ..., t^q] \cdot \boldsymbol{p},\ t \in [0, t_f], \tag{9}$$
where $\boldsymbol{p}$ is the coefficient matrix of the trajectory polynomial. Here, $p_r(t)$ and $\dot{p}_r(t)$ represent the position and orientation of the formation's centroid, respectively. The time information for each segment is computed assuming a trapezoidal velocity-time profile.

To adapt minimum snap to formations, we first include the inequality constraints in (5) during optimisation. This ensures the formation remains inside a convex polygon generated by IRIS [24], which guarantees collision avoidance. Second, we introduce an error term between the optimised trajectory and the trajectory corresponding to global path T, where T can be segmented based on the intermediate nodes in the path. The rth segment of the global path trajectory p_{o_r}, composed of endpoints $p_{t_{r-1}}$ and p_{t_r} can be represented as:
$$p_{o_r} = [p_{t_r} - \frac{p_{t_r} - p_{t_{r-1}}}{t_r - t_{r-1}} t_r, \frac{p_{t_r} - p_{t_{r-1}}}{t_r - t_{r-1}}, 0, 0, ..., 0]. \tag{10}$$
With this, the quadratic program for our extended minimum snap approach can be written as:
$$\min_p \sum_{r=1}^{S} \int_{t_{r-1}}^{t_r} (p_r^{(4)}(t))^2 + \lambda (p_r(t) - p_{o_r}(t))^2 dt \ \ s.t\ (5), \tag{11}$$
where λ is a weighting term. An example trajectory optimised with our approach is shown in Fig. 4(f).



(a) The initial (bottom left) and goal (top right) configurations of the formation.

(b) The convex regions corresponding to the initial and goal configurations.

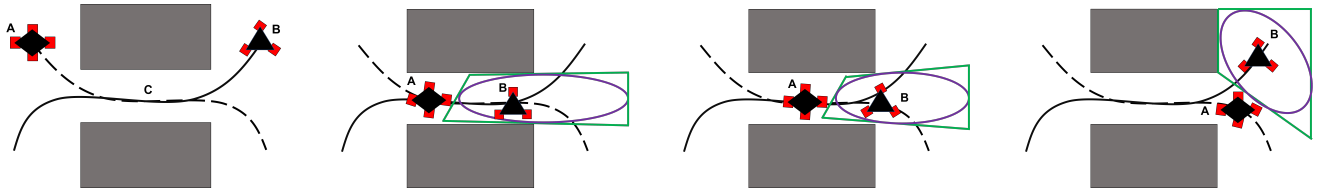
(c) The optimal seed point (black star) is found using Alg. 1 and a new convex region is generated.

(d) The optimal configuration in each convex region and intersection form a graph from which a global path is found (pink).

(e) An unsafe trajectory synthesised from the global path using [6].

(f) A safe trajectory synthesised from the global path using our adapted minimum snap approach.

Fig. 4. An illustrative example of global path planning and trajectory optimization for a triangular formation.



(a) Formations A and B are on track to collide at point C.

(b) Formation B treats formation A as an obstacle and computes its convex region.

(c) Formation B backs off to prevent a collision and computes the updated convex region.

(d) Formation A is eventually clear of B, and B returns to trajectory tracking.

Fig. 5. An illustrative example of our prioritised DWA for local motion planning and coordination. Solid and dashed lines represent robot trajectories.

B. Consensus-based Trajectory Tracking

To track the global trajectory, we combine pure pursuit control [9] with a consensus-based approach [4], where the formation centre is the virtual leader. For each robot, we use first-order consensus control to maintain the formation. Similar to (4), the control input for the ith follower of the jth formation at time t is given by:

$$u_i(t) = u(x_{ctrj}) + F_r(l_i(t)) + \sum_{f\in\{1,\ldots,k\}} w_{if}(x_{if}(t) - r_{if}), \quad (12)$$

where x_{ctrj} is the state of the virtual leader, $u(x_{ctrj})$ is the virtual leader's control input, and there are k robots in the formation.

C. Local Motion Planning Using DWA

When one formation senses another, it should deviate from its trajectory to prevent collisions. For this, we employ a prioritised DWA, where formations with longer trajectories have higher priority [26]. This is an ad-hoc solution for the coordination component of MFPC (see Fig. 2). Formation j begins collision avoidance behaviours upon sensing a higher priority formation. First, formation j computes the convex region around its virtual leader given the other formations and surrounding obstacles. We then sample a set of motion primitives m using DWA [7], i.e. velocities and angular velocities, which respect kinematic constraints and keep the formation within its convex region. The highest value motion primitive according to a function E is then executed, where E is defined as in [7]:

$$E(m) = w_1 \cdot Dir + w_2 \cdot Dis + w_3 \cdot Vel. \quad (13)$$

Here, Dir is the absolute directional change between the current velocity and the velocity in m, Dis is the average distance between formation j and any higher-priority formations after executing m, and Vel is the magnitude of the velocity in m. Formations execute collision avoidance primitives until higher priority formations are out of range, and then switch back to trajectory tracking. This ad-hoc coordination approach does not guarantee collision avoidance, as formations may be occluded, but collisions are reduced, as shown in Sec VII. We demonstrate our prioritised DWA in Fig. 5. In Fig. 5(a), formations A and B are on track to collide, where A has higher priority. Formation B computes its convex region (see Fig. 5(b)), and backs off to prevent a collision (see Fig. 5(c)). This is repeated until formation B is clear of A, and switches back to trajectory tracking (see Fig. 5(d)).

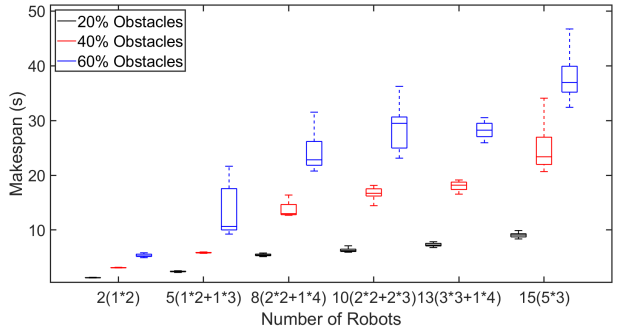


Fig. 6. Formation generation performance as the number of robots increase.

VII. Experiments

In this section, we demonstrate the efficacy of our approach in simulation. All experiments are implemented in MATLAB on an Intel i5 processor at 2.3GHz with 16GB of RAM. We consider 3 10×10m environments with randomly generated quadrilateral obstacles which cover 20%, 40%, and 60% of the environment respectively. All robots are 0.3m squares, where the velocity, angular velocity, acceleration, and angular acceleration are constrained within $\pm 0.3\,\mathrm{m/s}$, $\pm 0.35\,\mathrm{rad/s}$, $\pm 0.2\,\mathrm{m/s^2}$, and $\pm 0.8\,\mathrm{rad/s^2}$ respectively. Formations may be linear, triangular, or rectangular (see Fig. 3).

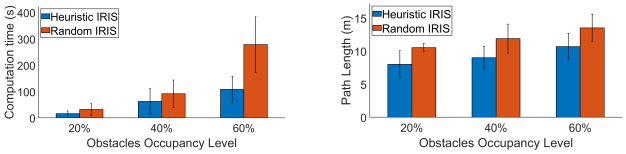
A. Formation Generation Performance

First, we evaluate formation generation performance as the number of robots increase. For each environment and number of robots, we generate 30 random problem instances, i.e. start and goal locations. We evaluate the makespan, i.e. the time for the last robot to reach its formation location. We present our results in Fig. 6, which also shows the formation configuration for each number of robots, e.g. $8(2*2+1*4)$ represents 8 robots forming 2 straight lines and 1 rectangle.

In Fig. 6, the makespan increases as the environment becomes more occluded, as robots must take longer routes, and because robot coordination is harder in tighter spaces. However, the makespan increases slowly with the number of robots, demonstrating how our approach effectively coordinates robots towards their formation locations. In all experimental runs, no robots became trapped in local minima.

B. Formation Planning Performance

Next, we evaluate our global formation planner against a variant which uses IRIS [24] with random seed points. In each environment, we randomly generate 30 formation start and goal locations for each of the three formation shapes, where start and goal locations are at least 5m apart. We combine the results for all formation shapes and present them in Fig. 7, where the area threshold in step 2 in Subsection V-B is set to 80%. Our planner consistently synthesises paths quicker than the random IRIS planner, where the gap increases with the number of obstacles. Our approach also synthesises shorter paths using Alg. 1 and the optimal configurations within each polygon and intersection.



(a) Computation time results. (b) Path length results.

Fig. 7. Global path planning performance as occupancy increases.

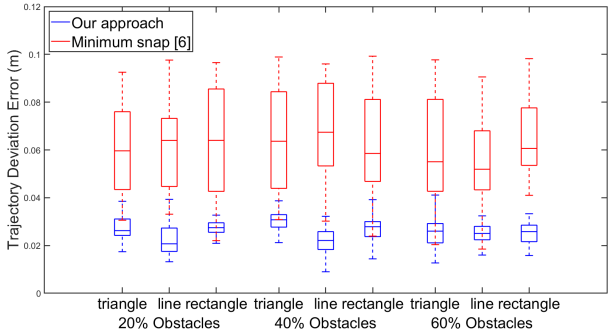


Fig. 8. The average trajectory deviation error of our approach against minimum snap [6].

We also demonstrate the efficacy of our extended minimum snap approach in Section VI-A. Using the global paths synthesised in the previous experiment, we run our trajectory optimisation method and the original minimum snap algorithm in [6]. For each run, we record the average trajectory deviation error, and present the results in Fig. 8. Our approach synthesises trajectories which are consistently easier for robots to track. We also recorded the total collisions in each trajectory: our approach produced collision-free trajectories across all runs, whereas minimum snap trajectories intersected with obstacles at least 4.6 times on average. This is because our approach is guaranteed to satisfy the formation's geometric constraints.

C. Formation Coordination Performance

We now evaluate our prioritised DWA for local motion planning and coordination. We generate 30 random 2-5 formation problems in the 60% occluded environment, and evaluate the number of inter-formation collisions with and without our approach. For each problem instance, the initial and goal formation positions are on either side of a circle of radius 5m centred on the map. We present our results in Table I. Our prioritised DWA resolves all conflicts for 2-3 formations, but not for 4-5 formations. This is because there may not exist feasible collision-free controls for lower priority robots in narrow regions of the environment. Despite this, our approach still reduces the total number of collisions.

VIII. Conclusion

In this paper, we proposed a framework for MFPC that outperforms state-of-the-art techniques. For formation generation, we extended APF approaches to avoid local minima. For formation planning, we applied a heuristic sampling strategy to improve path quality and computation

TABLE I

NUMBER OF INTER-FORMATION COLLISIONS WITH AND WITHOUT OUR PRIORITISED DWA.

Number of Formations	No Prioritised DWA	With Prioritised DWA
2 $(1*3+1*4)$	0.62 ± 0.48	$\mathbf{0}$
3 $(2*3+1*4)$	1.56 ± 0.93	$\mathbf{0}$
4 $(2*3+2*4)$	1.94 ± 0.89	$\mathbf{0.61 \pm 0.44}$
5 $(1*2+2*3+2*4)$	2.47 ± 1.16	$\mathbf{0.92 \pm 0.69}$

time. Finally, for formation coordination, we introduced a prioritised DWA to reduce inter-formation collisions. In future work, we will demonstrate the efficacy of our framework on real robots, explore coordination methods that guarantee inter-formation collision avoidance, and capture the effects of uncertainty on robot execution. Further, we will consider formations of real nonholonomic robots in transport environments shared with humans.

REFERENCES

[1] E. Tuci, M. H. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers in Robotics and AI*, vol. 5, p. 59, 2018.

[2] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistic Quaterly*, vol. 2, pp. 83–97, 1955.

[3] J. Wang, X. Wu, and Z. Xu, "Potential-based obstacle avoidance in formation control," *Journal of Control Theory and Applications*, vol. 6, pp. 311–316, 2008.

[4] Z. Peng, G. Wen, A. Rahmani, and Y. Yu, "Distributed consensus-based formation control for multiple nonholonomic mobile robots with a specified reference trajectory," *International Journal of Systems Science*, vol. 46, no. 8, pp. 1447–1457, 2015.

[5] R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi, and C. Melchiorri, "Edge-weighted consensus-based formation control strategy with collision avoidance," *Robotica*, vol. 33, no. 2, pp. 332–347, 2015.

[6] Z. Ma, H. Qiu, H. Wang, L. Yang, L. Huang, and R. Qiu, "A* algorithm path planning and minimum snap trajectory generation for mobile robot," in *Proceedings of the International Conference on Robotics, Control and Automation Engineering (RCAE)*, pp. 284–288, IEEE, 2021.

[7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[8] K. Moore and D. Lucarelli, "Forced and constrained consensus among cooperating agents," in *Proceedings of IEEE Networking, Sensing and Control*, pp. 449–454, 2005.

[9] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[10] Y. Mohan and S. Ponnambalam, "An extensive review of research in swarm robotics," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NABIC)*, pp. 140–145, IEEE, 2009.

[11] E. Bahceci, O. Soysal, and E. Sahin, "A review: Pattern formation and adaptation in multi-robot systems," tech. rep., Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43, 2003.

[12] M. Machida and M. Ichien, "Consensus-based artificial potential field approach for swarm," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pp. 6671–6677, IEEE, 2021.

[13] Z. Wu, W. Su, and J. Li, "Multi-robot path planning based on improved artificial potential field and b-spline curve optimization," in *Proceedings of the Chinese Control Conference (CCC)*, pp. 4691–4696, IEEE, 2019.

[14] G. Lee and D. Chwa, "Decentralized behavior-based formation control of multiple robots considering obstacle avoidance," *Intelligent Service Robotics*, vol. 11, pp. 127–138, 2018.

[15] J. C. Barca and A. Sekercioglu, "Generating formations with a template based multi-robot system," in *Proceedings of the Australasian Conference on Robotics and Automation*, 2011.

[16] T. Recker, H. Lurz, and A. Raatz, "Smooth spline-based trajectory planning for semi-rigid multi-robot formations," in *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1417–1422, IEEE, 2022.

[17] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[18] S. K. A. De Sousa, R. C. S. Freire, E. Á. N. Carvalho, L. Molina, P. C. Santos, and E. O. Freire, "Two-layers workspace: A new approach to cooperative object transportation with obstacle avoidance for multi-robot system," *IEEE Access*, vol. 10, pp. 6929–6939, 2022.

[19] M. U. Farooq, Z. Ziyang, and M. Ejaz, "Quadrotor uavs flying formation reconfiguration with collision avoidance using probabilistic roadmap algorithm," in *Proceedings of the International Conference on Computer Systems, Electronics and Control (ICCSEC)*, pp. 866–870, IEEE, 2017.

[20] K. H. Kowdiki, R. K. Barai, and S. Bhattacharya, "Leader-follower formation control using artificial potential functions: A kinematic approach," in *Proceedings of the IEEE International Conference On Advances In Engineering, Science And Management (ICAESM)*, pp. 500–505, IEEE, 2012.

[21] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6466–6472, 2021.

[22] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, "The hybrid path planning algorithm based on improved a* and artificial potential field for unmanned surface vehicle formations," *Ocean Engineering*, vol. 223, p. 108709, 2021.

[23] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[24] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Proceedings of Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*, pp. 109–124, Springer, 2015.

[25] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[26] J. P. Van Den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 430–435, IEEE, 2005.