2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Efficient Global Trajectory Planning for Multi-robot System with Affinely Deformable Formation

Hao Sha[1], Yuxiang Cui[1], Wangtao Lu[1], Dongkun Zhang[1], Chaoqun Wang[2], Jun Wu[1], Rong Xiong[1], Yue Wang[1†]

***Abstract*— Global trajectory planning is crucial for long-range formation navigation tasks of multi-robot systems in efficiency improvement and energy saving, whose main challenges are the joint space constraints of the whole team and the long-range deployment. To overcome the above difficulties, we reformulate the original problem into an affine formation planning problem in parameter space. Further, we propose a front-end & back-end framework for global trajectory planning of Multi-Robot Systems (MRS) with affinely deformable formation. For the front-end, an RL-steering affine formation RRT* method is designed to search a global formation-level trajectory in affine parameter space, combining the efficient BVP-solving capability of RL and the global guidance and generalizing ability of RRT*. For the back-end, we propose a formation-level affine parameter trajectory optimization method to refine the front-end trajectory, and further transform it into per-agent trajectories for execution. Extensive benchmarks and ablation experiments in simulation show the effectiveness of our framework for the global trajectory generation of a multi-UAV system with affinely deformable formation. The appendix can be seen here[3].**

I. Introduction

Global trajectory planning is crucial for long-range multi-robot navigation tasks in efficiency improvement and energy saving. However, it remains challenging when deployed in multi-robot tasks with formation-level constraints like collaborative object transportation [1], formation containment rescue [2], and mutual localization using line-of-sight sensors [3]. Those tasks commonly require the MRS to navigate in formation and ensure formation-level collision avoidance throughout the whole process, in which the obstacles are not allowed to cross through the formation. Therefore, the challenge for global trajectory planning of such MRS is to deal with the joint space constraints of the whole team and the long-range deployment.

Recent researches [4]–[7] factorize the formation planning problem into multiple individual planning problems that softly regard formation-level constraints, which cannot guarantee formation-level collision avoidance due to pure agent-level collision consideration. To free the planning framework from the above difficulties, [1] formulates the formation-level planning problem into the formation parameter space, further solved by a sampling-based method. Then the main issue is the solving of the boundary value problem (BVP) for node expansion during search. Generally, BVP for kinodynamic planning is formulated as an obstacle-neglected unconstrained optimal control problem followed by extra constraint checks [8]. Due to the time-consuming formation-level collision check and a large number of iterations caused by formation-level constraints only guaranteed in a short node-expansion horizon, this kind of method may be inefficient during searching in formation tasks, as shown in Fig. 1(a). Meanwhile, integrating formation-level constraints into the BVP makes it intractable for both closed-form and numerical solving for the non-convex and nonlinear properties. To improve search efficiency, [1] uses the corridor-based method to simplify the obstacle-aware BVP solving and avoid formation-level collision checks for node expansion. However, the generation of convex corridors still limits its application due to the high time consumption of nonlinear optimization.

With the explosion trend of deep learning in decision-making tasks, there emerges another line of work that

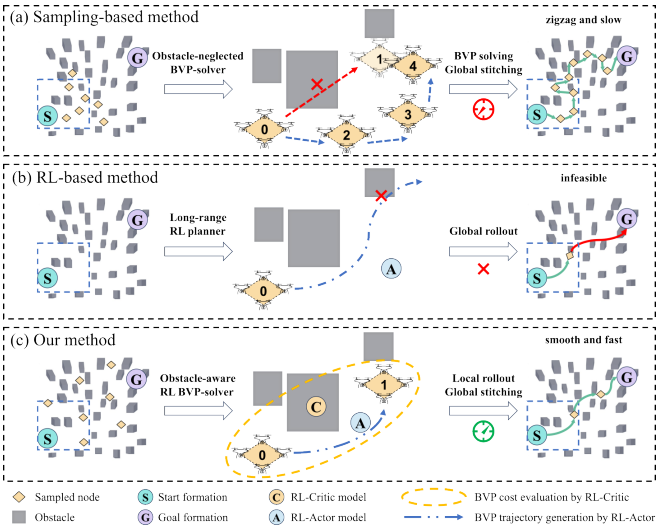

Fig. 1. Comparison between the sampling-based method, RL-based method, and our method. The sampling-based method suffers from inefficient BVP solving with time-consuming collision checks, leading to short node expansion horizon and zigzag planning results. The RL-based method suffers from low sample efficiency and weak generalizability for long-range planning problems. Our method, called RL-steering affine formation RRT*, combines the efficient obstacle-aware BVP-solving capability of RL and the global guidance and generalizing ability of RRT*.

This work was supported by the National Nature Science Foundation of China under Grant 62373322, Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F030001, Innovation and Development Special Fund of the Hangzhou Chengxi Sci-tech Innovation Corridor.

[1] The State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, P.R. China.

[2] The school of control science and engineering, Shandong University, Jinan, P.R.China.

[3]Appendix: https://drive.google.com/drive/folders/1dVA_zH9-q5dvhF2KbLUf5aigYTXE8mfB?usp=sharing.

employs deep reinforcement learning (RL) in multi-robot motion planning [9], [10] for its high efficiency and strong ability to solve non-convex and nonlinear problems. However, extending RL policy to global trajectory planning has two main difficulties as shown in Fig. 1(b): Firstly, RL is inefficient in long-horizon planning [11] and can easily be trapped in complex environments [12] due to its low sample efficiency in training. Second, the complex long-range environment layout cannot be adequately represented and fed to policy, leading to weak generalization.

In this paper, we utilize the complementary strength of RL policy and sampling-based search as shown in Fig. 1(c). Inspired by [13] and [1], we first model the original problem in the affine formation parameter space. Then, we design an RL-steering affine formation RRT* method, in which the affine formation BVPs are solved in parameter space using actor-critic RL with obstacle awareness for tree node expansions. Thanks to the efficient cost evaluation, the horizon of the node expansion can be much longer than obstacle-neglected BVP-based node expansion. Meanwhile, evaluating cost in RRT* becomes simply an RL-Critic network inference and high search efficiency can be expected. Finally, with the guidance of search results, a dense global formation-level trajectory composed of local trajectories generated by the RL-Actor network can be obtained. To further refine the trajectory, we propose the formation-level trajectory back-end optimization in the affine parameter space rather than the agent-wise optimization [14]. In this way, the formation-level constraints can be conveniently modeled and the affine formation constraint can be naturally satisfied. We summarize the contributions as:

1) We propose an RL-steering affine formation RRT* method to search a global formation-level trajectory in the affine parameter space by combining the efficient BVP-solving capability of RL and the global guidance and generalizing ability of RRT*.
2) We propose a formation-level affine parameter trajectory optimization method to refine the front-end trajectory and transform it into per-agent trajectories for execution.
3) We conduct extensive benchmark and ablation experiments on a multi-UAV system in simulation, which shows the efficiency and effectiveness of our method.

II. Related Work

A. Formation Front-end Planning

Båberg et al. [4] use a combination of constraint-based programming (CBP) and RRT to achieve formation obstacle avoidance, where RRT samples in the formation center position parameter space and CBP is used for steering considering formation maintaining and agent-level obstacle avoidance. [7] conducts formation-level global path-finding through a bidirectional RRT approach with agent-level collision check, which augments the formation scale parameter into the sampling space to enhance flexibility. In [4], [5], [7], although traveling in formation, robots avoid collision individually and the formation polygon may be inevitably split by obstacles, which is not up to the setting of this paper.

[1] solve the problem by proposing a sampling-based graph-search algorithm where convex corridors in free space are sampled and connected if their intersections are traversable for robot formation. [14] utilize the A* algorithm to plan the formation center path, and then a series of new formations are generated with scale factors for safety at each turning point along it. However, obstacle consideration becomes the main efficiency limit of the above methods, which leads to the short horizon and large time consumption due to corridor generation or formation-level collision check.

B. Formation Trajectory Optimization

Quan et al. [6] integrate a Laplacian-based formation similarity error metric into the spatial-temporal planning framework, simultaneously handling formation preservation and obstacle avoidance. To further reduce the computation overhead of the Laplacian-based gradient calculation, [7] only uses the above metric to generate optimal position sequences and then replaces it with the distance error between the generated reference and actual positions in iterative trajectory optimization. [14] elegantly introduces the affine formation constraint into the optimization framework of [15] by proving the transmissibility of the affine transformation between trajectory points and control points in the B-spline. Although the above methods consider the formation constraints in some contents, they can not ensure formation-level collision avoidance.

C. Combination of RL and Sampling-based Method

[12], [16] have conducted interesting combinations of RL with sampling-based planning methods to achieve kinodynamic planning for single robot tasks. The sampling-based method acts as a global planner to divide the long-term goal into sub-goal for RL policy, while RL policy acts as the local planner to offer the kinodynamic steering for the sampling-based methods. [16] adopt a TTR(time to return) estimation network to predict the node steering cost, while we use the critic model instead for its better consistency with the policy network and more comprehensive predictions for the optimized cost of BVPs than the TTR metric.

III. Affine Formation BVP

A. Problem Formulation

1) Affine formation: Consider a group of N planar moving mobile agents in $\mathbb{R}^2$ and $N \geq 3$. Let $p_{ag}^i \in \mathbb{R}^2$ be the position of agent i and $p_{ag} = [{p_{ag}^1}^T, \ldots, {p_{ag}^N}^T]^T \in \mathbb{R}^{2N}$ be positions of all the agents. Following the definition in [13], the time-varying formation $p_{ag}(t)$ can be affinely transformed from the nominal formation template as below:

$$p_{ag}(t) = [I_N \otimes A(t)]q + 1_N \otimes b(t) \tag{1}$$

where $\otimes$ denote as the Kronecker product, $q = [q_1^T, \ldots, q_N^T]^T \in \mathbb{R}^{2N}$ is a predefined agent-level time-invariant nominal formation template, $A(t) \in \mathbb{R}^{2\times 2}$ and $b(t) \in \mathbb{R}^2$ are the planar affine formation component:

$$A(t) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix} \begin{bmatrix} 1 & m(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_x(t) & 0 \\ 0 & s_y(t) \end{bmatrix},$$
$$b(t) = \begin{bmatrix} b_x(t) \\ b_y(t) \end{bmatrix}. \tag{2}$$

By defining affine transformation parameters as:

$$\begin{aligned} &p_{af}(t) = [\theta(t), m(t), ls_x(t), ls_y(t), b_x(t), b_y(t)]^T \in \mathbb{R}^6, \\ &ls_x(t) = ln(s_x(t)), ls_y(t) = ln(s_y(t)), \end{aligned} \tag{3}$$

any time-variant formation configuration $p_{ag}(t)$ can be represented by the combination of the time-variant affine transformation parameters $p_{af}(t)$ and the time-invariant nominal template q, namely $p_{ag}(t) = f(p_{af}(t), q)$. Then the multi-agent planning problem becomes an affine formation parameter planning problem, relaxing the dimension of the state space from $2N$ to 6, where $N \geq 3$ for affine formation. The time-varying affine parameter $p_{af}(t)$ is also named as the affine parameter position, with its first-order and second-order derivative w.r.t t as the affine parameter velocity $v_{af}(t)$ the acceleration $a_{af}(t)$ respectively, similar to the agent-level states p_{ag}, v_{ag}, a_{ag}.

2) Affine formation BVP: An affine formation planning problem can be formulated as a BVP in the affine parameter space, given a task-specific predefined formation template q and a pair of initial formation parameter p_{af}^{init} and goal formation parameter p_{af}^{goal} as the boundary values. Then it can be further modeled in optimal control fashion as follows:

$$\begin{aligned} \min_{p_{af}(t), v_{af}(t)} \ & \mathcal{J} = \int_0^\tau \mathbf{C}(p_{af}(t), v_{af}(t), p_{af}^{goal}) dt \\ s.t. \quad & \forall t \in [0, \tau], \dot{p}_{af}(t) = v_{af}(t), \\ & p_{af}(0) = p_{af}^{init}, p_{af}(\tau) = p_{af}^{goal}, \\ & p_{ag}(t) = f(p_{af}(t), q) \in \mathcal{P}_{ag}^{free}, \\ & v_{ag}(t) = h(p_{af}(t), v_{af}(t), q) \in \mathcal{V}_{ag}^{valid} \end{aligned} \tag{4}$$

where τ is the time duration of the trajectory, $\mathbf{C}(p_{af}(t), v_{af}(t), p_{af}^{goal})$ is the objective function, defined as Equ. 7 in Appendix A, including time, goal formation similarity, and control effort cost terms, the $f(\cdot)$ and $h(\cdot)$ represent the mapping between the agent-level state and the formation-level state according to Equ. 8 in Appendix A. An optimal control law is to be found to drive the affine parameter system from the initial formation to the goal formation minimizing the objective functional while considering the formation-level obstacle avoidance constraints and agent-level dynamic constraints.

Difficulties of formation BVP. The formation-level obstacle avoidance constraints are naturally non-convex, while the agent-level dynamic constraints are nonlinear in the affine parameter space according to the nonlinear mapping $f(\cdot)$ and $h(\cdot)$. The non-convex and nonlinear nominal constraints make it intractable for both the analytical and optimization-based methods to efficiently solve the affine formation BVP.

B. Reinforcement Learning BVP Approximation

To deal with the non-convex and nonlinear constraints, we employ goal-conditioned reinforcement learning to solve the formation BVP.

1) Goal-conditioned RL formulation: The observation $o_t = [M_{local}(t), p_{af}(t), q]$ consists of obstacle representation $M_{local}(t)$, current affine parameter $p_{af}(t)$, and the nominal formation template q. For obstacle representation, we use a local occupancy map anchoring on the formation center. The nominal formation template is involved since the same affine parameter can lead to different actual formations conditioned on different templates. The goal g is the terminal boundary value p_{af}^{goal} of the affine formation BVP. The action a_t is the affine parameter velocity $v_{af}(t)$ same as the control input of affine formation BVP.

According to the objective functional and constraints of the affine formation BVP defined in Equ. 4, the reward for RL training is designed as follows:

$$R = \lambda^{\mathrm{T}}[r_{\text{goal}}, r_{\text{time}}, r_{\text{effort}}, r_{\text{formC}}, r_{\text{interC}}, r_{\text{viol}}]^{\mathrm{T}} \tag{5}$$

where r_{goal} consist of the sparse reward for goal formation achievement and the dense punishment related to the negative Euclidean distance to the goal in the affine parameter space, r_{time} is the constant time cost per step, r_{effort} is a dense punishment related to agent-level velocity to minimize total energy cost, r_{formC} consist of sparse punishment for formation shape polygon collision and dense punishment related to the distance between the nearest obstacle and formation polygon calculated by GJK (Gilbert–Johnson–Keerthi) [17] algorithm, r_{interC} represents punishment related to the collision between agents in the formation, r_{viol} is a dense exponentially punishment related to the agent-level violation on the dynamic constraints to build a soft barrier for limits violation, λ is the weight vector to trade-off the above factors.

2) Policy learning: We adopt the TD3 framework [18] in policy training for its continuous action space decision-making ability and actor-critic architecture. Specifically, the RL-Actor model $\pi(o_t, g)$ acts as an efficient formation-level BVP approximation.The RL-Critic $\mathbf{Q}(o_t, a_t, g)$ is further used for cost prediction before actual policy execution, which plays a key role in our global planning algorithm.

Network architecture. We use a convolutional neural network to encode the grid map and multi-layer perceptron networks to encode other components in the observation. During training, the curriculum learning paradigm is applied by gradually increasing the difficulty of scenes e.g. obstacle density and goal formation range. The architecture and training details are described in Appendix A.

IV. Method

In this section, we introduce the proposed front-end & back-end framework as shown in Fig. 2.

A. RL-steering Affine Formation RRT*

Although good at dealing with non-convex and nonlinear constraints, the RL-BVP approximation struggles with the

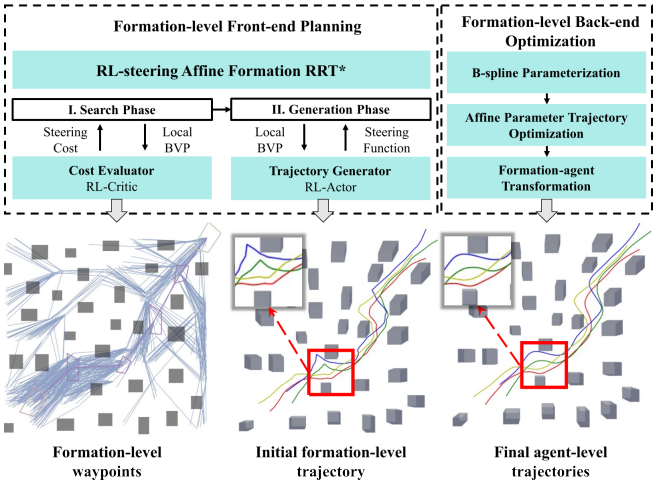

Fig. 2. The total framework of our method.

long-range global BVP. If we extend the local map $M_{local}(t)$ to the global map and give the goal g with the terminal goal formation parameter $p_{af}^{terGoal}$, the low sampling efficiency will cause the failure of learning in such a high-dimensional space. Thus, we propose to regard RL-BVP approximation as a steering function, which is embedded in RRT* for affine formation planning in long range. In this way, the new node sampled in each step of RRT* is assigned as the goal of RL-BVP approximation for cost evaluation and trajectory rollouts. The main workflow of the RL-steering affine formation RRT* method is described as Alg. 1, which consists of the search phase and the generation phase.

1) Search Phase: This phase aims at searching for a sequence of affine parameter waypoints. We mainly replace the Euclidean distance-based cost metric assisted by collision checks in the classical RRT* with the obstacle-aware RL-Critic model, which avoids time-consuming collision checks and extends the node expansion horizon. Since the search process mainly cares about the cost to the given nodes rather than the explicit trajectory, node expansion steering in this phase can be achieved by simple inference of the critic model in 1-3 ms without unnecessary rollouts of RL-Actor model.

Critic-based expansion. During an iteration, a random node p_{af}^{rand} is firstly sampled with formation-level collision check from the affine parameter space $\mathbb{R}^6$ defined in Section III-A. **Nearest**() is used to find the nearest node $p_{af}^{nearest}$ in the search tree with p_{af}^{rand} roughly determined by the Euclidian distance in the affine parameter space for efficiency. Then, **Project**() is used to project p_{af}^{rand} into the range of the local map M_{local} anchored on $p_{af}^{nearest}$ as a new node p_{af}^{new}. To be specific, if the p_{af}^{rand} situated out of the domain of the M_{local}, linear interpolation in the affine parameter space will be adopted to regenerate the p_{af}^{new} near the boundary of the local map with formation-level collision check. Centered on the p_{af}^{new}, **Near**() is used to find those nodes in the domain of local map M_{local} as the candidate parent node set as P_{af}^{near}. To select the best parent node, the costs c_{steer} between each candidate parent node in X_{near} and p_{af}^{new} are evaluated by **RLSteering-C**() as $c_{steer} = U - \mathbf{Q}(o_t, a_t, g)$, where U is the upper bound of Q-value to ensure c_{steer} to be positive. Note that $\mathbf{Q}(o_t, a_t, g)$ reflects the cost of obstacle-aware formation BVP, thus the horizon of the node expansion can be long without explicit collision check. After the parent node selection, the new edge topologically connecting p_{af}^{parent} and p_{af}^{new} is added to the tree. Then **Rewire**() is conducted to update the tree structure for potentially shorter paths, among which the **RLSteering-C**() is also used to evaluate the cost. Finally, when p_{af}^{new} reaches $p_{af}^{terGoal}$ in a predefined threshold, a series of waypoints can be backtracked from $\mathcal{T}$ as $\mathbf{P}_{af}$.

Algorithm 1: RL-steering Affine Formation RRT*

Notations: global grid map M_{global}, affine parameter search tree $\mathcal{T}$, affine parameter path $\mathbf{P}_{af}$, affine parameter trajectory set $\mathbf{T}_{af}$, front_end affine parameter trajectory $\mathbf{T}_{front\_end}$
input : $M_{global}, q, p_{af}^{init}, p_{af}^{terGoal}, \mathcal{T}$
output : $\mathbf{T}_{front\_end}$

I.Search phase:
Initialize: $\mathcal{T} \leftarrow \emptyset \cup \{p_{af}^{init}\}$;
for $i \leftarrow 1$ **to** n **do**
  $p_{af}^{rand} \leftarrow$ **Sample**(M_{global});
  $p_{af}^{nearest} \leftarrow$ **Nearest**$(\mathcal{T}, p_{af}^{rand})$;
  $p_{af}^{new} \leftarrow$ **Project**$(p_{af}^{nearest}, p_{af}^{rand})$;
  $P_{af}^{near} \leftarrow$ **Near**$(\mathcal{T}, p_{af}^{new})$;
  $c_{min} \leftarrow \infty, p_{af}^{parent} \leftarrow NULL$;
  for $p_{af}^{near} \in P_{af}^{near}$ **do**
    $c_{steer} \leftarrow$ **RLSteering-C**$(p_{af}^{near}, p_{af}^{new})$;
    if ***Cost***$(p_{af}^{near}) + c_{steer} < c_{min}$ **then**
      $c_{min} \leftarrow$ **Cost**$(p_{af}^{near}) + c_{steer}$;
      $p_{af}^{parent} \leftarrow p_{af}^{near}$;
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{p_{af}^{parent}, p_{af}^{new}\}$;
  Rewire$(P_{af}^{near}, p_{af}^{new})$;
  if ***IsArrive***$(p_{af}^{new}, p_{af}^{terGoal})$ **then**
    $\mathbf{P}_{af} \leftarrow$ **ExtractPath**$(\mathcal{T})$;
    break;
II.Generation phase:
for $\{p_{af}^{cur}, p_{af}^{next}\} \in \boldsymbol{P}_{af}$ **do**
  $t_{af} \leftarrow$ **RLSteering-A**$(p_{af}^{cur}, p_{af}^{next})$;
  if ***IsConstraintViolated***(t_{af}) **then**
    $t_{af} \leftarrow$ **Recovery**$(p_{af}^{cur}, p_{af}^{next})$;
  $\mathbf{T}_{af} \leftarrow \mathbf{T}_{af} \cup t_{af}$;
return $\mathbf{T}_{front\_end} \leftarrow$ **StitchTraj**$(\mathbf{T}_{af})$;

2) Generation Phase: In this phase, **RLSteering-A**() is employed to connect each consecutive node pair in $\mathbf{P}_{af}$ by iteratively calling the RL-Actor model to generate the dense trajectory $\mathbf{T}_{front\_end}$. Note that the trajectory generation is achieved by model-based forward rollout assuming ideal state transition. As shown in Equ. 4, since the RL-Actor

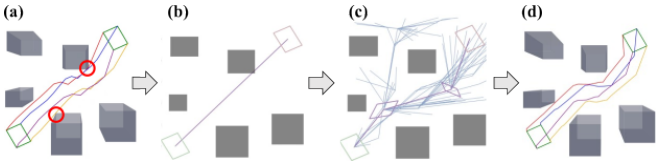

Fig. 3. A case study of the constraint violation check and recovery process for the front-end. (a): To perform constraint violation check along the global trajectory and locate the local segment that violates constraints. (b): To extract the node pair on duty for the constraints violation. (c): To search the intermediate nodes between node pairs in (b) to divide the challenging BVP into simpler ones. (d): Trajectories regeneration through RL-Actor between new sampled nodes to guarantee constraint satisfaction.

model considers the formation-level affine parameter dynamics, time and effort cost, obstacle avoidance, and agent-level velocity limits, the final trajectory is time & energy-efficient and obstacle-aware. Compared with classic RRT* performing trajectory generation in each step, our method only generates the explicit trajectory along the RRT* result, avoiding numerous trajectory rollouts for tentative node connection. Besides, explicit collision checks are also avoided since the RL-Actor ensures the safety of the trajectory. To further accelerate the processing, the piecewise trajectory rollouts are calculated in parallel among all consecutive node pairs in batch on GPU.

Constraint violation check and recovery. Since the RL policy serves as the approximation of the Boundary Value Problem (BVP), occasional constraint violations may occur during **RLSteering-A**(). To ensure the constraints guarantee of T_{front_end}, violation checks are performed using **IsConstraintViolated**() for each rollout segment t_{af}. For the segment violating the constraints, denoted as t_{af}^{vio}, **Recovery**() is employed: i) Locally, an affine RRT* is conducted between the relevant node pair $p_{af}^{cur}, p_{af}^{next}$ of t_{af}^{vio}, using a Euclidean-distance-based cost metric and formation-level collision check to search for intermediate nodes that divide the challenging BVP into simpler sub-BVPs; ii) The trajectory for each simpler sub-BVP is regenerated through **RLSteering-A**(), and the process of constraint violation check and recovery is recursively invoked until the total t_{af}^{vio} is fixed without constraint violation. A case study of the above process is illustrated in Fig. 3.

B. RRT* planner and RL learner co-enhancing

To achieve the RRT* planner and RL learner co-enhancing, we close the planning-learning loop like [19]. During the co-enhancing, there are two processes conducted in synchrony: i) RL learner gathers experiences from the planning process of the RL-steering affine formation RRT* and uses the data to persistently enhance the RL-Critic and RL-Actor models; ii) the enhanced steering cost evaluator and steering function are fed back to the RRT* planner. The RRT* planner provides data augmentation for RL learning to improve the generalizing ability and sample efficiency; at the same time, the RRT* planner benefits from the RL learner for improved planning performance and efficiency.

C. Affine Parameter Trajectory Post-optimization

The generated affine parameter trajectory $\mathbf{T}_{front\_end}$ is further refined by trajectory post-optimization and transformed to agent-level trajectories for execution.

Trajectory parametrization. The affine parameter trajectory $\mathbf{T}_{front\_end}$ from front-end is parameterized as a B-spline curve with a degree of 3 and K control points $\{Q_1, \cdots, Q_K\}$, where $Q_i \in \mathbb{R}^6$ is in affine formation parameter space.

Trajectory optimization. The affine parameter trajectory optimization problem is originally a nonlinear constrained problem according to the affine formation BVP in Equ.2. Note that the trajectory $\mathbf{T}_{front\_end}$ satisfies all the constraints, we can employ it as a feasible initial guess, which enables a simplification of the problem to be unconstrained by turning the hard constraints to soft forms. Thus, we model the affine parameter trajectory refinement as an unconstrained non-linear optimization problem, which is formulated as follows:

$$\arg\min_{Q} J = \lambda_s J_s + \lambda_f J_f + \lambda_r J_r + \lambda_c J_c + \lambda_g J_g \tag{6}$$

where $\lambda_s, \lambda_f, \lambda_r, \lambda_c, \lambda_t$ are the respective weights, the objective function defined above considers factors of both the formation level and agent level. For the former, formation-level collision avoidance J_c and goal formation fitness J_c are considered. For the latter, agent-level smoothness J_s, dynamic feasibility J_f, and reciprocal collision avoidance J_r are considered. The details of the cost terms are presented in Appendix B. Thanks to the unconstrained problem simplification, the problem can be solved efficiently using L-BFGS [20] in about 1-5 ms. We follow [21] to keep no constraint violation by performing agent-level and formation-level constraint checks along the trajectory iteratively before the optimization termination.

Agent-level trajectories transformation. We finally transform the formation-level affine parameter trajectory $[p_{af}(t), v_{af}(t), a_{af}(t)]$ into agent-level trajectories $[p_{ag}(t), v_{ag}(t), a_{ag}(t)]$. following Equ. 8 in Appendix A, arriving at executable trajectories for all robots.

V. Experimental Results

A. Metrics

To conduct fair performance comparisons among different methods, we have designed a series of metrics to evaluate the performance: 1) time metric m_t: the total time consumption of the actual execution of the trajectories; 2) path metric m_p: the path length calculated by the average accumulated velocity of all agents during the trajectory execution; 3) formation metric m_f: the accumulated degree of discrepancy between the actual formation and the terminal goal formation configurations, which is defined in Appendix C. We further define comprehensive metric m_c as the average of the above three metrics to evaluate the performance from different aspects in a unified way.

B. Benchmark Experiments

1) Setup: To demonstrate the efficiency and robustness of our method, benchmark comparisons are conducted with cutting-edge formation planning methods, including Alonso's

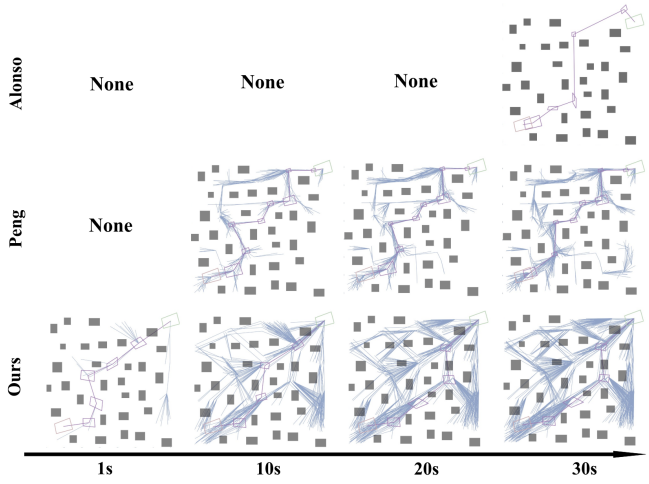

Fig. 4. Front-end search results of different methods at different time steps. Three rows correspond to Alonso(up), Peng(middle), and Ours(bottom) respectively. "None" means no feasible solution.

method [1] and Peng's method [14]. Through the XTDrone [22] simulation environment, we simulate a multi-UAV system flying in formation across a cluttered environment under a velocity limit of 2m/s. Given a randomly generated 2D occupancy grid map, a task-specific formation template, and a pair of the initial and terminal goal affine parameters, the planning framework of each method is conducted to generate trajectories for each robot execution. The experiments are conducted in three kinds of forest-like scenarios with different obstacle densities. For each obstacle density, experiments are conducted in ten random maps to calculate the average performance. For each scenario, we record five trials at different time steps of the front-end search phase to compare the efficiency of methods. More details can be seen in Appendix C.

2) Results: The results are summarized in Table I. It states that our method can find the feasible trajectory with a shorter time and lower cost m_c than both Alonso's and Peng's methods in all three kinds of scenarios. In some cases, the m_c of our solution at the first time step is even lower than the solution of other methods at the fifth time step. The time efficiency mainly owes to the obstacle-aware RL-Critic model that frees our method from both the time-consuming collision check and safe corridor construction, while the lower cost mainly gives credit to our back-end optimizer. Table II shows the statistical results of benchmark experiments on 100 maps in three kinds of forest-like scenarios, which shows the performance advantage of our method on each separate metric over other methods. Fig. 4 shows the front-end search results, where our method has an obviously larger node expansion horizon than the other methods. Fig. 5 shows the final trajectories planned by different methods in forest-like and corridor-like scenarios, the trajectories from our method have a faster finish, shorter path length, and better formation maintenance for the goal formation.

TABLE I

MAIN RESULTS OF BENCHMARK EXPERIMENTS.

Scenario	Method\Time	0.8s	1s	1.5s	2s	5s
Sparse1	Alonso	-	-	-	-	-
	Peng	-	34.01	34.01	34.43	32.34
	Ours	**33.68**	**33.68**	**27.55**	**27.55**	**26.89**
Sparse2	Alonso	-	-	36.93	36.93	35.02
	Peng	33.60	32.91	33.92	31.17	31.50
	Ours	**24.30**	**24.30**	**24.30**	**23.78**	**23.78**
Sparse3	Alonso	-	-	-	-	39.13
	Peng	-	-	-	32.01	29.90
	Ours	**28.37**	**28.37**	**27.60**	**27.27**	**25.29**
	Time	1s	2s	4s	7s	10s
Medium1	Alonso	-	-	-	-	-
	Peng	-	-	42.17	38.22	38.17
	Ours	**30.31**	**30.31**	**28.01**	**28.77**	**21.08**
Medium2	Alonso	-	-	-	-	51.94
	Peng	-	-	-	-	-
	Ours	**-**	**38.57**	**38.23**	**36.28**	**28.73**
Medium3	Alonso	-	-	-	-	42.92
	Peng	-	40.02	34.61	34.61	34.61
	Ours	**36.17**	**36.17**	**33.70**	**33.70**	**33.70**
	Time	1s	5s	10s	20s	30s
Dense1	Alonso	-	-	-	-	48.88
	Peng	-	-	-	50.25	48.64
	Ours	**-**	**47.35**	**28.47**	**28.47**	**28.47**
Dense2	Alonso	-	-	-	-	-
	Peng	-	35.08	35.08	35.12	35.12
	Ours	**29.46**	**27.65**	**27.81**	**27.81**	**27.81**
Dense3	Alonso	-	-	-	-	50.91
	Peng	-	39.33	38.42	39.12	38.20
	Ours	**-**	**32.21**	**32.21**	**28.46**	**28.46**

TABLE II

BENCHMARK EXPERIMENTS RESULTS ON SEPARATE METRICS IN SPARSE, MEDIUM, AND DENSE FOREST SCENARIOS.

Scenario	Method	m_t	m_p	m_f	m_c
Sparse	Alonso	35.29	48.37	27.57	37.07
	Peng	29.55	45.04	17.82	30.80
	Ours	**24.79**	**42.01**	**17.74**	**28.18**
Medium	Alonso	34.84	53.03	39.81	42.56
	Peng	33.64	52.53	40.41	42.19
	Ours	**22.74**	**43.00**	**32.26**	**32.66**
Dense	Alonso	30.39	48.37	38.52	39.09
	Peng	28.94	45.04	34.94	36.31
	Ours	**23.74**	**42.01**	**32.61**	**32.78**

C. Ablation Study

1) Setup: We have designed 7 ablation methods to prove the necessity and evaluate the impact of each component for our total framework. The notations and the ablation components of them are summarized in Table III. Note that the ablation experiments share the same setup as the benchmark experiments.

TABLE III

NOTATIONS AND SETTINGS OF THE ABLATION METHODS

Method	**Ablation component**	**Alternative**
Ours w/o CE	Critic-based cost evaluator	Euclidian distance cost with formation-level collision-check
Ours w/o AS	affine parameter search space	translation parameter search space
Ours w/o AG	Actor-based trajectory generator	minimum acceleration trajectory generation [23]
Ours w/o RL	all of the RL elements in front-end	combination of Ours w/o AS and Ours w/o AG
Ours-PF	total front-end	Peng's front-end
Ours-AF	total front-end	Alonso's front-end
Ours-PB	total back-end	Peng's back-end

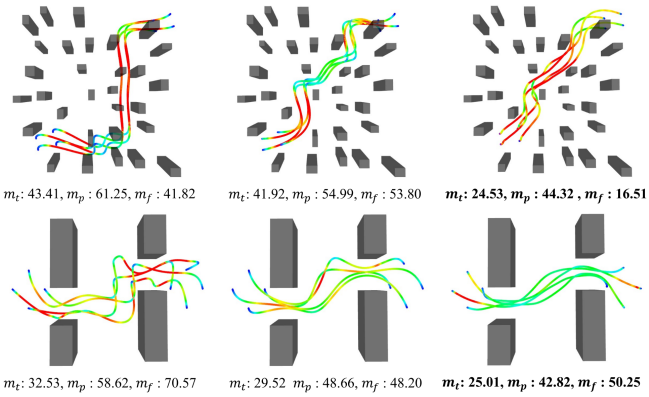

Fig. 5. Final trajectories planned by different methods in forest-like and corridor-like scenarios. Three columns correspond to Alonso(left), Peng(middle), and Ours(right) respectively. The more the color of the trajectory shifts towards warmer tones, the higher the speed it represents.

2) Results: The results in three kinds of forest-like scenarios are summarized in Fig. 6. The two columns represent the comprehensive metric cost m_c and success rate of each method respectively, while the two color bars record the average performance at 1s and 30s for front-end search respectively. It states that our method has lower comprehensive costs and higher success rates than most of the ablation methods in all three kinds of scenarios at each time step. **Ours-PB** and **Ours** have similarly high success rates but **Ours** has the lower cost. This means that the front-end efficiently ensures the existence of solutions while the back-end mainly contributes to the solution refinement. The conclusion can also be supported by the fact that **Ours-PF** has similarly low costs with **Ours** while its success rate sharply drops down. The performance of **Ours-AF** is weak due to the over-conservation of its corridor-based front-end that can not be refined by our back-end optimization. **Ours w/o AG** has more serious performance degradation than **Ours w/o CE** from **Ours**, since the alternative minimum acceleration trajectory generation is not obstacle-aware like the RL-Actor model. This undoubtedly leads to a serious mismatch with the RL-Critic cost evaluator and initial guesses for the back-end optimization. Instead, **Ours w/o CE** uses a more conservative cost evaluator with collision check which actually eases the burden of the consecutive RL-Actor model but at the price of search efficiency. Since **Ours w/o RL** abandons all of the RL elements for both cost evaluator and trajectory generator, it nearly suffers the overlayed performance degradation of both **Ours w/o CE** and **Ours w/o AG**. The performance degradation of **Ours w/o AS** from the **Ours** is slighter than both **Ours w/o CE** and **Ours w/o AG** since it retains full RL elements. It only suffers from the loss of search space from affine transformation to pure translational transformation.

D. Real World Experiment

Real-world experiments on quadrotor formation are conducted in forest-like and corridor-like scene respectively.

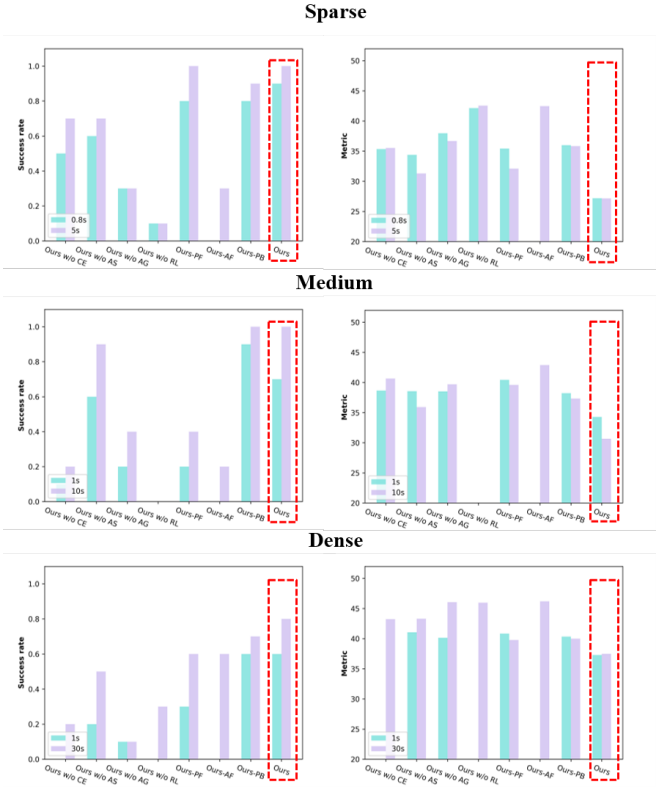

Fig. 6. The ablation experiment results. The three rows of the sub-figures correspond to the sparse, medium, and dense forest scenarios respectively. The two columns of the sub-figures record success rate (left ↑) and comprehensive metric m_c (right ↓) respectively. The different color bars of each sub-figure represent different front-end search time settings.

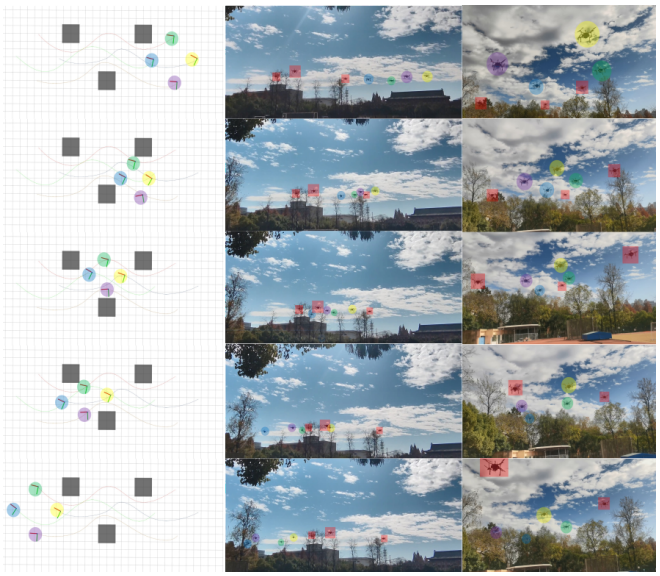

Fig. 7. Visualization of real experiment in the forest-like scenario. The vertical axis represents the timeline, while each row corresponds to the snapshot of a time step. Three columns correspond to top-down view (left), side view (middle), and front view (right) respectively. The red square annotations indicate obstacles, while the circular annotations in other colors (yellow, purple, blue, and green) denote the quadrotors in the formation.

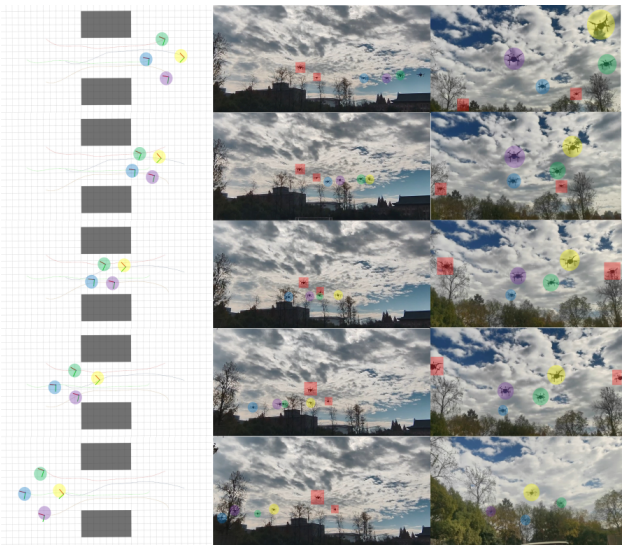

Fig. 8. Visualization of real experiment in the corridor-like scenario.

The formation consists of four quadrotors with the square nominal formation template. The initial formation is set as a diamond shape, while the target formation is transformed from The initial formation through a translation. Some other quadrotors serve as obstacles hovering at the same altitude. The experiment visualizations shown in Fig. 7, 8 demonstrate the effectiveness of our method in the real world.

VI. Conclusion

In this paper, we propose a front-end & back-end framework for global trajectory planning of MRS with affinely deformable formation. For the front-end, an RL-steering affine formation RRT* method is designed to search a global formation-level trajectory in the affine parameter space combining the efficiency of RL and the generalization of RRT*. For the back-end, a formation-level affine parameter trajectory optimization method is proposed to refine the initial trajectory. Extensive benchmark and ablation experiments on a multi-UAV formation system prove the efficiency and effectiveness of our method.

References

[1] 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.

[2] J. Hu, P. Bhowmick, I. Jang, F. Arvin, and A. Lanzon, "A decentralized cluster formation containment framework for multirobot systems," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1936–1955, 2021.

[3] L. Yin, F. Zhu, Y. Ren, F. Kong, and F. Zhang, "Decentralized swarm trajectory generation for lidar-based aerial tracking in cluttered environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9285–9292, IEEE, 2023.

[4] F. Båberg and P. Ögren, "Formation obstacle avoidance using rrt and constraint based programming," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 1–6, IEEE, 2017.

[5] W. Liu, J. Hu, H. Zhang, M. Y. Wang, and Z. Xiong, "A novel graph-based motion planner of multi-mobile robot systems with formation and obstacle constraints," *arXiv preprint arXiv:2210.03340*, 2022.

[6] L. Quan, L. Yin, C. Xu, and F. Gao, "Distributed swarm trajectory optimization for formation flight in dense environments," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4979–4985, IEEE, 2022.

[7] L. Quan, L. Yin, T. Zhang, M. Wang, R. Wang, S. Zhong, Y. Cao, C. Xu, and F. Gao, "Formation flight in dense environments," *arXiv preprint arXiv:2210.04048*, 2022.

[8] H. Ye, N. Pan, Q. Wang, C. Xu, and F. Gao, "Efficient sampling-based multirotors kinodynamic planning with fast regional optimization and post refining," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3356–3363, IEEE, 2022.

[9] Y. Cui, H. Zhang, Y. Wang, and R. Xiong, "Learning world transition model for socially aware robot navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9262–9268, 2021.

[10] Y. Cui, L. Lin, X. Huang, D. Zhang, Y. Wang, W. Jing, J. Chen, R. Xiong, and Y. Wang, "Learning observation-based certifiable safe policy for decentralized multi-robot navigation," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5518–5524, 2022.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[12] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 5113–5120, IEEE, 2018.

[13] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, 2018.

[14] P. Peng, W. Dong, G. Chen, and X. Zhu, "Obstacle avoidance of resilient uav swarm formation with active sensing system in the dense environment," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10529–10535, IEEE, 2022.

[15] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 4101–4107, IEEE, 2021.

[16] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "Rl-rrt: Kinodynamic motion planning via learning reachability estimators from rl policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.

[17] M. Montanari, N. Petrinic, and E. Barbieri, "Improving the gjk algorithm for faster and more reliable distance queries between convex objects," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 3, pp. 1–17, 2017.

[18] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[19] P. Cai and D. Hsu, "Closing the planning–learning loop with application to autonomous driving," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 998–1011, 2022.

[20] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[21] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.

[22] K. Xiao, L. Ma, S. Tan, Y. Cong, and X. Wang, "Implementation of uav coordination based on a hierarchical multi-uav simulation platform," in *Advances in Guidance, Navigation and Control: Proceedings of 2020 International Conference on Guidance, Navigation and Control, ICGNC 2020, Tianjin, China, October 23–25, 2020*, pp. 5131–5143, Springer, 2022.

[23] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: The 16th International Symposium ISRR*, pp. 649–666, Springer, 2016.

[24] K. Qin, "General matrix representations for b-splines," in *Proceedings Pacific Graphics' 98. Sixth Pacific Conference on Computer Graphics and Applications (Cat. No. 98EX208)*, pp. 37–43, IEEE, 1998.

[25] Q. Wang, Z. Wang, L. Pei, C. Xu, and F. Gao, "A linear and exact algorithm for whole-body collision evaluation via scale optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3621–3627, IEEE, 2023.