2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Give me scissors: Collision-Free Dual-Arm Surgical Assistive Robot for Instrument Delivery

Xuejin Luo, Shiquan Sun, Runshi Zhang, Ruizhi Zhang, Junchen Wang*

***Abstract*— During surgery, scrub nurses are required to frequently deliver surgical instruments to surgeons, which can lead to physical fatigue and decreased focus. Robotic scrub nurses provide a promising solution that can replace repetitive tasks and enhance efficiency. Existing research on robotic scrub nurses relies on predefined paths for instrument delivery, which limits their generalizability and poses safety risks in dynamic environments. To address these challenges, we present a collision-free dual-arm surgical assistive robot capable of performing instrument delivery. A vision-language model is utilized to automatically generate the robot's grasping and delivery trajectories in a zero-shot manner based on surgeons' instructions. A real-time obstacle minimum distance perception method is proposed and integrated into a unified quadratic programming framework. This framework ensures reactive obstacle avoidance and self-collision prevention during the dual-arm robot's autonomous movement in dynamic environments. Extensive experimental validations demonstrate that the proposed robotic system achieves an 83.33% success rate in surgical instrument delivery while maintaining smooth, collision-free movement throughout all trials. The project page and source code are available at https://give-me-scissors.github.io/.**

I. INTRODUCTION

The scrub nurse is a vital member of the surgical team, primarily responsible for assisting the surgeon with tasks such as delivering surgical instruments and managing retractors. However, this work is highly mechanized and repetitive, which can lead to physical fatigue and potential errors among scrub nurses [1]. Additionally, in units understaffed with nurses, the lack of a scrub nurse leads to a significant decrease in the efficiency of the surgical team, potentially resulting in more serious safety issues [2]. New technological methods, such as robotic-assisted surgery and automation tools, present a promising solution to these challenges [3]. These advanced technologies not only alleviate the burden on scrub nurses but also enhance efficiency of surgical team [4].

Researchers have developed various surgical assistive robots to function as scrub nurses and carry out the highly mechanized task of surgical instrument delivery. A robotic scrub nurse was designed for safe human-robot collaboration in the operating room [5]. Surgeons can request surgical instruments through speech commands and hand gestures. The dual-arm robotic system was proposed for surgical instruments transferring tasks in [6]. A deep learning-based multi-modal robotic framework was presented in [1], which can work with surgeons via speech and image inputs. However, the categories of surgical instruments and the delivery paths must be predefined in these approaches, which limits their autonomy and generalization. Furthermore, the absence of real-time collision avoidance capabilities poses safety challenges in dynamic and complex surgical environment [7].

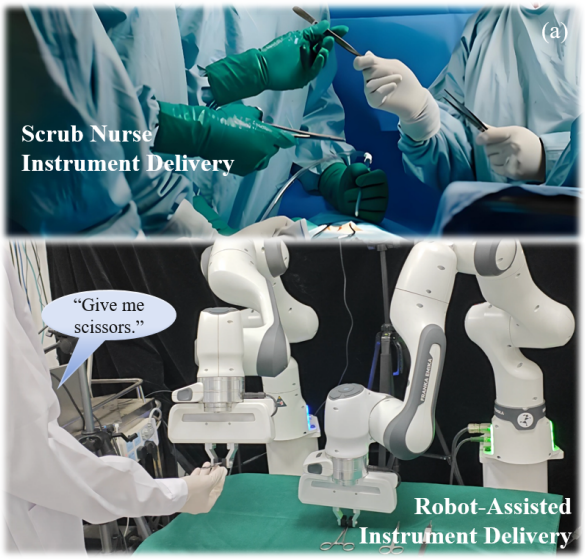

Fig. 1: Surgical instruments transfer process. (a) Scrub nurse instrument delivery. (b) Robot-assisted instrument delivery.

In recent years, Vision Language Models (VLMs) have been widely used in robotic autonomy tasks [8]. They have demonstrated remarkable performance in scene perception and motion planning. VLM is integrated into the robotic planning system in [9], enabling the combination of language instructions and image inputs to generate high-level task plans. VoxPoser [10] applies VLM to obtain the 3D value map used within a planning framework to synthesize closed-loop robot trajectories. ReKep [11] utilizes VLM to associate objects in the environment with keypoints, generating constraints relevant to navigation tasks. T-Rex [12] proposes a VLM-based task-adaptive spatial representation extraction framework for robotic manipulation. Benefiting from VLMs, the robots in these studies demonstrate capabilities in scene semantic understanding and action grounding. In zero-shot tasks, these robots achieved autonomous planning and movement, indicating the excellent generalization abilities. However, VLM-based autonomous planning is an emerging field in robotic-assisted surgery. In the surgical assistive

This work was supported in part by the Natural Science Foundation of China under Grant 62573022, Grant U22A2051; and in part by the Natural Science Foundation of Beijing Municipality under Grant L232037. (*Corresponding author: Junchen Wang*)

X. Luo, S. Sun, R. Zhang, R. Zhang, and J. Wang are with the School of Mechanical Engineering and Automation, Beihang University, Beijing, China. wangjunchen@buaa.edu.cn

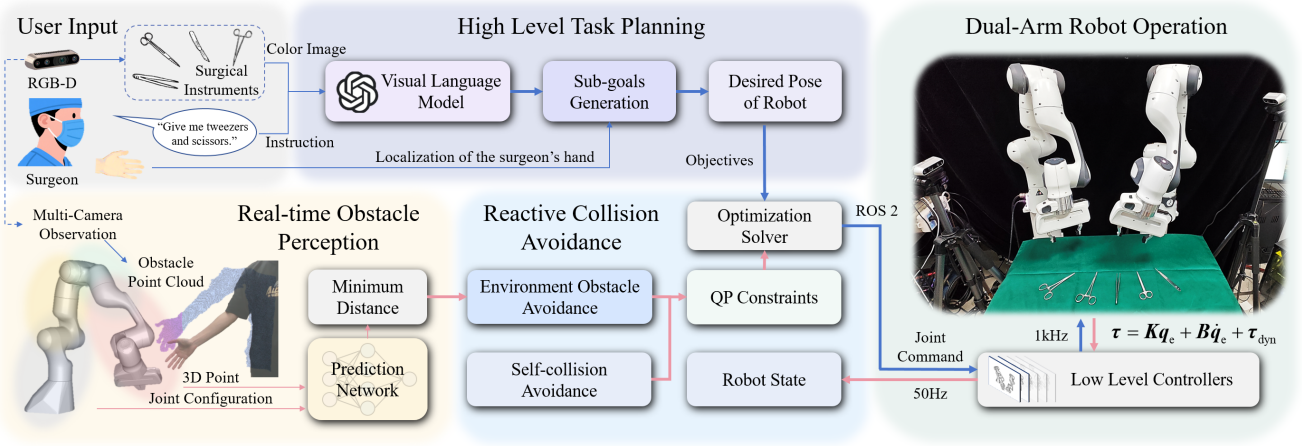

Fig. 2: The pipeline of the collision-free dual-arm surgical assistive robot for instrument delivery. The robot system receives multi-modal inputs from the physical world (i.e., surgeon instruction, color image, depth data). The real-time obstacle perception module computes the minimum distance between robot and environmental obstacles. The QP framework is built upon the minimum distance, ensuring the dual-arm robot's collision-free operation. The high-level task planning utilizes VLM to generate the desired motion objectives for the QP framework.

robotic system, leveraging VLMs for motion planning offers distinctive advantages. On one hand, their multi-modal capabilities enable robots to intuitively comprehend the surgeon's instructions. On the other hand, their autonomous planning ability allows surgical assistive robots to adapt to the dynamic intraoperative environment.

Autonomous motion of robots imposes high demands on system safety [13], [14]. Ensuring collision-free operation during robotic-assisted surgery is particularly challenging [15]. Traditional sampling-based global planners can generate collision-free paths offline [16]. However, high computational cost makes them unsuitable for dynamic and unstructured environments [17]. Several methods for real-time obstacle avoidance have been reported. A reactive collision avoidance method was presented in [18], which generated collision-free motion in joint space. A high-order control barrier functions (CBFs) framework for collision avoidance among convex primitives was proposed in [19]. ToMPC [20] introduced a task-oriented Model Predictive Control (MPC) framework for safe and efficient robotic manipulation in open workspaces. Nevertheless, these works rely on visual markers and sensors, which limits their application in unstructured environments. In addition to obstacle avoidance, dual-arm systems must also consider self-collision avoidance due to the overlap of their workspaces [21]. A real-time dual-arm self-collision avoidance method was proposed in [13]. Building on these previous efforts regarding obstacle and self-collision avoidance, the dual-arm surgical assistive robot faces unique requirements. It must detect obstacles in an unstructured dynamic environment without the use of markers, and perform reactive obstacle avoidance and self-collision avoidance simultaneously.

In this work, we present a collision-free dual-arm surgical assistive robot for instrument delivery. It utilizes VLM to achieve scene semantic understanding, autonomously planning the robot's grasping and delivery paths based on multi-modal inputs, including surgeon's instructions and visual features of surgical instruments. The entire ***zero-shot*** process does not need fine-tuning or predefined operations. During autonomous motion, the robot maintains high real-time perception of the nearest obstacles, without the need for visual markers or prior environmental modeling. A unified quadratic programming (QP) framework is utilized to achieve reactive obstacle avoidance and self-collision avoidance based on minimum distance perception. Extensive real-world experimental validation in collision avoidance and surgical instrument delivery has confirmed the system's robustness and safety. The main contributions are as follows:

- A dual-arm surgical assistive robot for instrument delivery is developed. It utilizes VLM to automatically generate the robot's grasping and delivery motion based on surgeon's instructions.
- A unified real-time QP framework is proposed to achieve dual-arm robot reactive obstacle avoidance and self-collision avoidance simultaneously during the autonomous movement.
- The proposed robotic system achieved a success rate of 83.33% in real-world instrument delivery experiments, with no collisions occurring, thereby demonstrating its effectiveness and safety.

II. Method

The overview of the dual-arm surgical assistive robot is illustrated in Fig. 2. A real-time obstacle perception method is proposed, predicting the distance from the robotic arm links to obstacles. The minimum distance is used to construct the nonlinear constraint in the QP framework to ensure real-time collision avoidance. The proposed QP framework functions

as a safety filter, achieving motion objectives while ensuring that the robot satisfies constraints for obstacle avoidance, self-collision avoidance, and joint limits. The VLM generates task level sub-goals by interpreting multi-modal inputs from surgeon's commands and camera observations, providing motion objectives for the QP framework. The methodology is elaborated as follows: Section II-A introduces the process of real-time obstacle perception. Section II-B describes the architecture of the reactive collision avoidance QP framework. Section II-C presents the task-planning mechanism enabled by the VLM.

A. Real-time Obstacle Perception

The real-time perception of environmental obstacles is essential for achieving reactive collision avoidance. The primary goal is to identify the closest point to the robot and the direction in which the robot should move away. Let $\mathbb{P}$ denote the set of all point clouds in the environment. To reduce computational cost, $\mathbb{P}$ is filtered as follows to isolate the point cloud in the robot's vicinity. For each joint configuration $\boldsymbol{q}$, the occupancy volumes of n links $\mathbb{S}_{\mathrm{r}}(\boldsymbol{q}) = \{s_1, s_2, \ldots, s_{\mathrm{n}}\} \subset \mathbb{R}^3$ are determined through forward kinematics (FK), where $s_{\mathrm{i}} \subset \mathbb{R}^3$ represents the three-dimensional volume occupied by the i-th link. To accelerate collision detection, the manipulator's complex geometry is approximated by a capsule model, denoted as $\mathbb{S}_{\mathrm{cap}}(\boldsymbol{q})$:

$$\mathbb{S}_{\mathrm{cap}}(\boldsymbol{q}) = \bigcup_{i=1}^{n} \left\{ \mathbb{P} \in \mathbb{R}^3 \mid d(\mathbb{P}, l_i) \leqslant r_{\mathrm{cap}} \right\} \tag{1}$$

where l_i is the line segment connecting the adjacent joint centers p_{i1} and p_{i2}. r_{cap} is the radius of the capsule. $d(\mathbb{P}, l_i)$ denotes the minimum distance from a point to the line segment. Eq. 1 enables the rapid identification of points that lie inside safety capsules. It avoids computing the distance from $\mathbb{P}$ to every point on the complex mesh $\mathbb{S}_{\mathrm{r}}(\boldsymbol{q})$, which have a time complexity of $O(n^2)$.

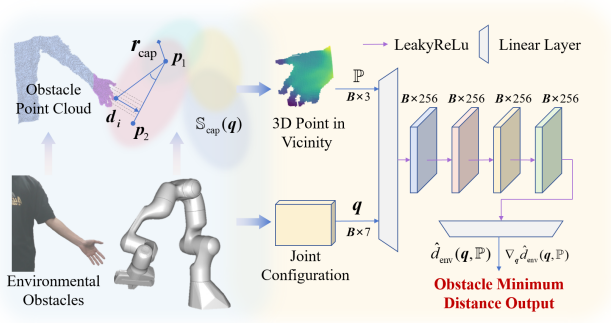

Fig. 3: Real-time obstacle perception process. The robot joint configuration and point cloud of obstacles are taken as input. Output is the minimum distance between robot and obstacle.

The set $\mathbb{P}$ within $\mathbb{S}_{\mathrm{cap}}(\boldsymbol{q})$ contains a substantial number of points belonging to the robot. To prevent self-collision interference, these points must be filtered. The filtering process is performed by first generating the 2D robot mask via image segmentation [22]. Then the mask is mapped to the synchronized depth map to extract the robot's point cloud.

Then the robot vicinity $\mathbb{S}_{\mathrm{v}} \subset \mathbb{R}^3$ is calculated as follows

$$\mathbb{S}_{\mathrm{v}}(\boldsymbol{q}) = \left\{ \mathbb{S}_{\mathrm{cap}}(\boldsymbol{q}) \cap \bar{\mathbb{S}}_{\mathrm{r}}(\boldsymbol{q}) \right\} \tag{2}$$

Let $\mathbb{P}_{\mathrm{v}} \subset \mathbb{R}^3$ be the set of environmental point cloud in $\mathbb{S}_{\mathrm{v}}(\boldsymbol{q})$. $\bar{\mathbb{S}}_{\mathrm{r}}(\boldsymbol{q})$ denotes the complement of $\mathbb{S}_{\mathrm{r}}(\boldsymbol{q})$, representing the points that do not belong to the robot. The closest point $\mathbb{P}_{\min} \in \mathbb{P}_{\mathrm{v}}$ to $\mathbb{S}_{\mathrm{r}}(\boldsymbol{q})$ and the corresponding minimum distance $d_{\min}$ are calculated as

$$\begin{aligned} \mathbb{P}_{\min} &= \underset{\mathbb{P}_i \in \mathbb{P}_{\mathrm{v}}}{\operatorname{argmin}} \left\| \mathbb{P}_i - \mathbb{S}_{\mathrm{r}}(\boldsymbol{q}) \right\|^2 \\ d_{\min} &= \left\| \mathbb{P}_{\min} - \mathbb{S}_{\mathrm{r}}(\boldsymbol{q}) \right\| \end{aligned} \tag{3}$$

where $\mathbb{P}_i$ is the i-th point in $\mathbb{P}_{\mathrm{v}}$. To avoid high computational costs, a distance prediction neural network $\hat{d}_{\mathrm{env}}(\boldsymbol{q}, \mathbb{P})$ is proposed to approximate the $d_{\min}$. The network architecture is illustrated in Fig. 3. The inputs include $\mathbb{P}_{\mathrm{v}}$ and the current robot joint configuration $\boldsymbol{q}$. Linear layers with high-dimensional features are employed to capture the implicit relationships between the obstacle points and the robot. Non-linear activation functions are utilized to model the complex interactions. The output consists of the minimum distance $\hat{d}_{\mathrm{env}}(\boldsymbol{q}, \mathbb{P})$ and gradient obtained through backpropagation.

B. Reactive Collision Avoidance Framework

A QP framework is utilized to achieve reactive collision avoidance. The QP framework is formulated as follows:

$$\begin{aligned} \Delta \boldsymbol{q}_{\mathrm{de}} = \underset{\Delta \boldsymbol{q}}{\operatorname{argmin}} & \frac{1}{2} \alpha \underbrace{\left\| \Delta \boldsymbol{q} - \boldsymbol{J}^{\dagger}(\boldsymbol{q}_{\mathrm{c}}) \boldsymbol{v}_{\mathrm{de}} \right\|^2}_{\mathrm{A}} \\ & + \frac{1}{2} \beta \underbrace{\left\| \Delta \boldsymbol{q} + \boldsymbol{q}_{\mathrm{c}} - \boldsymbol{q}_{\mathrm{de}} \right\|^2}_{\mathrm{B}} + \underbrace{\Delta \boldsymbol{q}^T \boldsymbol{Q} \Delta \boldsymbol{q}}_{\mathrm{C}} \end{aligned} \tag{4}$$

$$\text{s.t.} \quad \begin{cases} \boldsymbol{q} = \boldsymbol{q}_{\mathrm{c}} + \Delta \boldsymbol{q} \\ \ln\left(\frac{\hat{d}_{\mathrm{env}}(\boldsymbol{q}, \mathbb{P})}{\lambda} \right) + \Delta \boldsymbol{q} \cdot \nabla_{\boldsymbol{q}} \hat{d}_{\mathrm{env}}(\boldsymbol{q}, \mathbb{P}) \geq 0 \\ \ln\left(\frac{\hat{d}_{\mathrm{self}}(\boldsymbol{q})}{\mu} \right) + \Delta \boldsymbol{q} \cdot \nabla_{\boldsymbol{q}} \hat{d}_{\mathrm{self}}(\boldsymbol{q}) \geq 0 \\ \boldsymbol{q}^{\min} < \boldsymbol{q} < \boldsymbol{q}^{\max} \\ \zeta^{\min} < \Delta \boldsymbol{q} < \zeta^{\max} \end{cases} \tag{5}$$

where $\Delta \boldsymbol{q} = \left[\Delta \boldsymbol{q}^{\mathrm{L}}, \Delta \boldsymbol{q}^{\mathrm{R}} \right]^T$ represents the joint-space increments of left and right arm, respectively. In Eq. (4), the goal is to optimize the desired joint-space $\Delta \boldsymbol{q}_{\mathrm{de}}$ to satisfy dual-arm robot motion tasks while maintaining safety.

1) Cartesian Velocity Objective: Term A in Eq. (4) denotes the optimization objective for dual-arm cartesian velocity. $\boldsymbol{J}^{\dagger}(\boldsymbol{q}_{\mathrm{c}})$ is the pseudo-inverse of the Jacobian matrix at current joint configurations $\boldsymbol{q}_{\mathrm{c}}$. $\boldsymbol{v}_{\mathrm{de}}$ is desired Cartesian velocity. It is transformed into joint-space increments via $\boldsymbol{J}^{\dagger}(\boldsymbol{q}_{\mathrm{c}})$, ensuring $\Delta \boldsymbol{q}$ is optimized to approach dual-arm desired Cartesian velocity. α is a positive weight factor.

2) Reference Joint Objective: Term B in Eq. (4) represents the optimization objective for dual-arm robot desired joint configurations. $\Delta \boldsymbol{q}$ is optimized to narrow the gap between the current joint configurations $\boldsymbol{q}_{\mathrm{c}}$ and the desired joint configurations $\boldsymbol{q}_{\mathrm{de}}$. β is the positive weight factor of term B.

The optimization objectives of Term A and Term B ensure that the motion of the dual-arm robot aligns with the task goals. Term C is utilized to keep the variation range of $\Delta \boldsymbol{q}$ as small as possible. $\boldsymbol{Q}$ is the weight matrix.

3) Obstacle Collision Constraints: The distance prediction neural network $\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})$ proposed in II-A is utilized to construct the obstacle collision constraints. λ is the safety distance threshold between robot and environment obstacles. The gradient of $\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})$ with respect to $\boldsymbol{q}$ is given by

$$\nabla_{\boldsymbol{q}} \hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P}) = \frac{\partial \hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})}{\partial \boldsymbol{q}} = \mathcal{B}\left(\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})\right) \quad (6)$$

where $\mathcal{B}$ denotes the backpropagation operation. When $\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})$ is less than λ, $\ln\left(\frac{\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})}{\lambda}\right)$ will be negative. Then it forces $\Delta \boldsymbol{q}$ to align with the direction of the gradient $\nabla_{\boldsymbol{q}} \hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})$ to satisfy the constraint. Due to the nonlinearity of the logarithm function, the influence of the term $\ln\left(\frac{\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})}{\lambda}\right)$ on $\Delta \boldsymbol{q}$ increases as the robot approaches the environment obstacles. Conversely, when $\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})$ is greater than λ, the constraint relaxes, allowing $\Delta \boldsymbol{q}$ to be optimized freely.

4) Self-collision Constraints: In third term of Eq. (5), the self-collision constraint is established using the distance prediction neural network $\hat{d}_{\text{self}}(\boldsymbol{q})$. Inspired by [13], $\hat{d}_{\text{self}}(\boldsymbol{q})$ is trained to predict the minimum distance between dual arms. μ is the corresponding safety distance threshold. $\nabla_{\boldsymbol{q}} \hat{d}_{\text{self}}(\boldsymbol{q})$ is the gradient of $\hat{d}_{\text{self}}(\boldsymbol{q})$ with respect to $\boldsymbol{q}$, follows a pattern analogous to that in Eq (6). The mechanism of the self-collision avoidance constraint is similar to II-B.3.

The last two lines of Eq. (5) represent the constraints of joint limits and joint velocity respectively. $\boldsymbol{q}^{\min}$ and $\boldsymbol{q}^{\max}$ denote the lower and upper bounds of the joints, while $\zeta^{\min}$ and $\zeta^{\max}$ represent the lower and upper limits of joint velocity. The QP framework (4)(5) serves as a real-time safety filter, ensuring collision-free motions of the dual-arm robot in dynamic environment.

C. Task Planning

High-level task planning provides the desired motion objectives for the QP framework based on the surgeon's commands. The specific process of high-level task planning is illustrated in Fig. 2. The robot system interprets semantic information from the surgeon and effectively processes multi-modal inputs. During surgery, the instructions from the surgeon are recorded and converted into text format, denoted as $\boldsymbol{T}$. Multiple RGB-D cameras are employed to capture color images and depth information of the surgical instruments. Let the images be denoted as $\boldsymbol{I}$, and the point clouds transformed from depth be represented as $\mathbb{P}_{\text{obj}}$. We calculate the pixel features $\boldsymbol{X}$ utilizing visual model DINOv2 [23]. Then segmentation model SAM [24] is applied to get the masks of n objects, denoted as $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n\}$. For each mask $\boldsymbol{m}_i$, the 3D keypoint $\boldsymbol{p}_i$ is generated as follows

$$\boldsymbol{p}_i = \underset{o_i \subset \mathbb{R}^3}{\operatorname{argmin}} \sum_{j=1}^{k} \left\{ \left\| \mathbb{P}_j^{m_i} - o_i \right\|^2 + \left\| \boldsymbol{X}_i^j - \boldsymbol{X}_i^{o_i} \right\|^2 \right\} \quad (7)$$

where o_i is the center of the i-th object point cloud. $\mathbb{P}_j^{m_i}$ represents the j-th point within one of the k points that belongs to the mask $\boldsymbol{m}_i$. $\boldsymbol{X}_i^j$ and $\boldsymbol{X}_i^{o_i}$ denote the features of the j-th point and the center point of the i-th object, respectively. The keypoints $\boldsymbol{p}_i$ are projected onto $\boldsymbol{I}$ as numbered visual markers, represented as $\boldsymbol{I_p}$. This visual prompting approach allows the VLM to bridge the gap between geometric segments and semantic understanding without explicit category labels. Inspired by [11], we derive the task-level objectives by taking $\boldsymbol{T}$ and $\boldsymbol{I_p}$ as inputs:

$$\mathcal{G}_{\text{task}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_l\} = \text{VLM}(\boldsymbol{T}, \boldsymbol{I_p}, \mathcal{P}) \quad (8)$$

where $\mathcal{P}$ is the prompt template [1]. $\mathcal{G}_i$ is the i-th stage sub-goal, taking the form of L_2 norm objective function associated with the keypoints. Meanwhile, VLM determines the grasp and release action stages. To identify the surgeon's interaction intentions, a hand landmarks detector Mediapipe [25] is applied in task planning to locate the keypoint of the surgeon's hand near the robot's space. After minimizing the loss function and performing interpolation, a series of desired Cartesian positions are obtained for the dual-arm robot. Then the desired joint configurations $\boldsymbol{q}_{\text{de}}$ are calculated via inverse kinematics (IK) for the QP framework.

III. Experiments and Results

A. Experimental Setup

The experimental platform consisted of both simulation and real-world components. The dual-arm robotic system comprised two Franka Research 3 robotic arms. Two industrial computers (MIC-770-V2, Advantech, China) with Intel Core i7-10700 CPUs and 8 GB memory, running Ubuntu 22.04 LTS with the PREEMPT_RT kernel at 1 kHz, served as the low-level controllers. For high-level operations, two separate workstations were employed: the first, equipped with an Intel Core i9-14900KF CPU and an NVIDIA RTX 4090 GPU, was dedicated to high-level task planning and QP optimization; the second workstation was responsible for RGB-D camera data acquisition and real-time perception processing. Three Intel RealSense D435i RGB-D cameras provided overlapping views for environmental perception. ROS2 served as the middleware to connect all system devices. The Flexible Collision Library (FCL) [26] was utilized to generate the ground truth for minimum distances. Numerical computations of QP optimization were performed using the SLSQP solver within the SciPy library.

B. Collision Avoidance Simulation Experiment

1) Implementation Details: To evaluate the performance of proposed collision avoidance QP framework, simulation experiments of dual-arm robot collision avoidance were conducted. The robot model was constructed within the PyBullet corresponding to the real robot. Each robotic arm's end-effector followed an elliptical trajectory (see Fig. 4(a)), which led to potential self-collision between arms. Two spheres

[1]Please refer to the source code available at `https://give-me-scissors.github.io/`

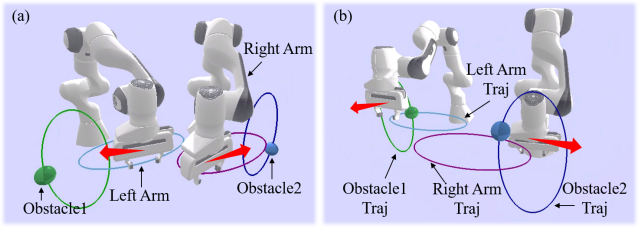

Fig. 4: Obstacle and self-collision avoidance process of the dual-arm robot in simulation. The red arrows indicate the avoidance directions for dual-arm robot. (a) Self-collision avoidance. (b) Obstacle avoidance.

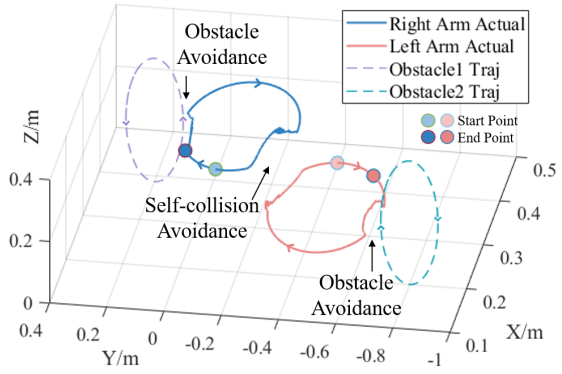

Fig. 5: The trajectories of dual-arm robot's end-effector and obstacles during simulation. The avoidance motion is achieved by our method.

with a radius of 3 cm (see Fig. 4(b)) were generated to follow elliptical trajectories near the dual arms, simulating the dynamic obstacles in the environment. $\hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P})$ was a MLP-based network consisting of 4 hidden layers with 256 neurons, and the training process was consistent with [27]. The parameter in $\hat{d}_{\text{self}}(\boldsymbol{q})$ and the training process were the same as in [13]. In the experiment, the safety distance threshold λ and μ in Eq. (5) were adopted as 0.1 m. α was 5, β was 1, $\boldsymbol{Q}$ was an identity matrix. The joint configurations, end-effector positions and obstacles positions were recorded during the experiment. The actual values of the minimum self-distance between dual arms and the minimum distance between robot and obstacles were calculated using FCL.

2) Comparison Methods: Three state-of-the-art (SOTA) reactive collision avoidance methods [28]–[30] were compared. All these methods were extended to encompass both obstacle avoidance and self-collision avoidance. (1) DawnIK [28] incorporates collision avoidance as an optimization objective. $\epsilon_1, \epsilon_2 = 0.1$ (represented in meters). λ and μ are consistent with Eq. (5), and similarly for the following. (2) CollisionIK [29] encodes distance to collision state using the following cost function. The scalar values n, s, c, r are the same as in [29]. $\epsilon_3 = 0.1$. (3) CBF-QP [30] constructs the collision constraint in QP framework, formulated as:

$$\dot{b}(q, \Delta q) + \gamma(b(q)) \geq 0 \tag{9}$$

TABLE I: Comparison results between the competing methods and our method

Method	Obs	Self	Opt Cost Time (s)	Mean Pos Error (m)	Max Accel (m/s^2)
DawnIK [28]	✘	✔	0.034	0.116	31.94
CollisionIK [29]	✔	✔	0.038	0.136	32.10
CBF-QP [30]	✔	✔	0.035	0.055	39.47
Ours	✔	✔	**0.022**	**0.054**	**17.77**

'Obs' represents the obstacle avoidance. 'Self' indicates the self-collision avoidance.

where $\gamma(r) = \mathcal{K}r$ is the linear class-$\mathcal{K}$ function. According to the obstacles collision and self-collision avoidance in our experiment, $b(q)$ can be divided into $b_{\text{env}}(q)$ and $b_{\text{self}}(q)$. where $b_{\text{env}}(q) = \hat{d}_{\text{env}}(\boldsymbol{q}, \mathbb{P}) - \lambda$ and $b_{\text{self}}(q) = \hat{d}_{\text{self}}(\boldsymbol{q}) - \mu$ substitute the first two terms of Eq. 5. $\mathcal{K}$ = 0.9.

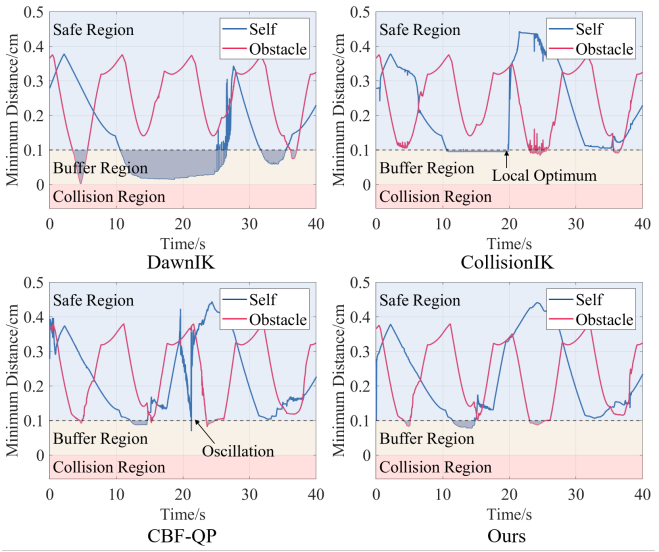

Fig. 6: Minimum self-distance between dual arms and minimum distance between right arm and obstacle. The shadow region represents the portion where the minimum distance falls below the safety distance threshold.

3) Results: The simulation trajectories of the dual-arm robot's end-effectors and the obstacles are illustrated in Fig. 5. The dual-arm robot effectively achieved self-collision avoidance and obstacle avoidance. Meanwhile, it was able to return to the desired trajectories smoothly after completing the avoidance maneuvers. The comparative experimental results are shown in Table I. The CollisionIK, CBF-QP and our proposed method successfully accomplished obstacle avoidance, while DawnIK failed (see Fig. 6). All four methods were capable of achieving self-collision avoidance. Our method has the shortest optimization time, demonstrating better real-time performance. This is attributed to its rapid convergence during the optimization process. Additionally, our method exhibits minimum mean position error, indicating that it can closely align with the desired trajectory while ensuring safety. It also recorded the smallest maximum accel-

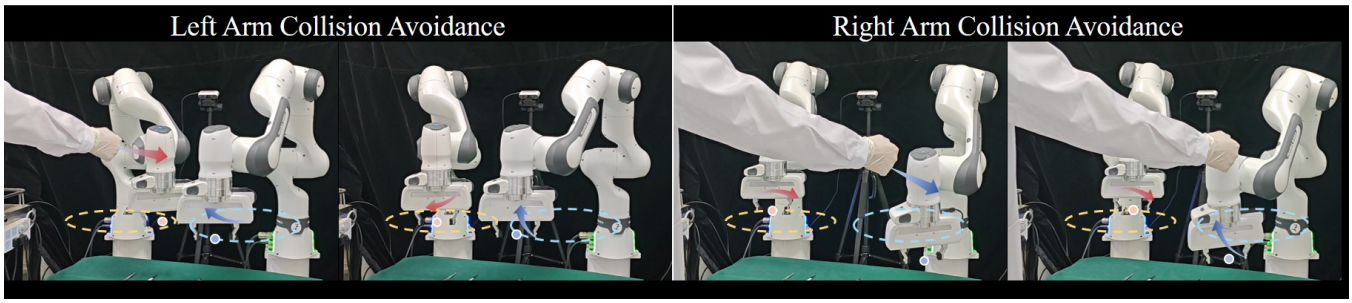

Fig. 7: Dual-arm robot avoids obstacle and self-collision during motion. Yellow and blue ellipses represent the desired trajectories of dual-arm robot end-effector. Red/blue arrow denotes the motion direction of the left/right arm.

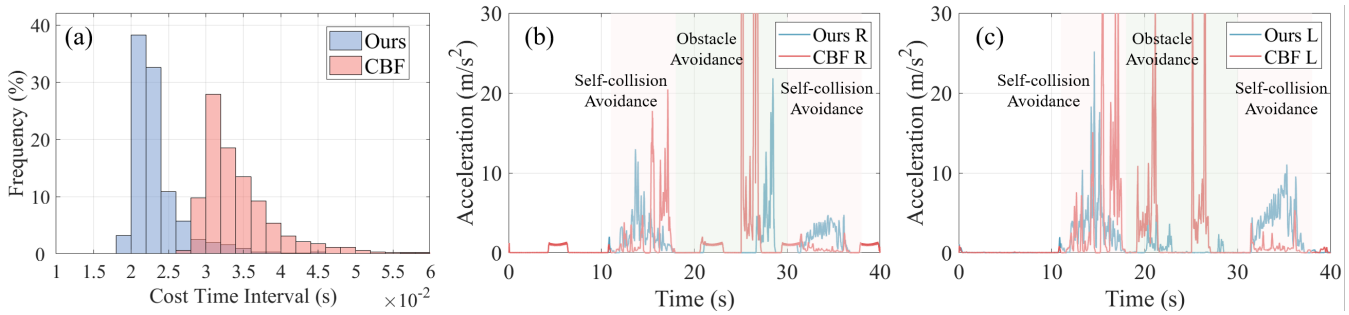

Fig. 8: The results of the collision avoidance real-world experiment. (a) Histogram of optimization cost time. (b) Right arm end-effector's acceleration. (c) Left arm end-effector's acceleration.

eration among all methods, reflecting smoothness during the avoidance process. Fig. 6 illustrates the minimum distance during the simulation among all four methods. DawnIK encountered an obstacle collision at 4.7s. CollisionIK got trapped in a local optimum during self-collision avoidance. CBF-QP exhibited unstable oscillations. In contrast, our method consistently maintained smooth avoidance motion.

C. Collision Avoidance Real-world Experiment

1) Implementation Details: To verify the safety of the robotics system in complex surgical scenarios, we conducted collision avoidance experiment in real-world settings. Dual-arm robot's end-effectors followed the same elliptical trajectory as in simulation experiment to verify the self-collision avoidance. The surgeon approached the dual arms to evaluate the effectiveness of dynamic obstacle avoidance. Multiple RGB-D cameras were employed to collaboratively capture the point cloud of obstacles near the robot. The minimum distance between the obstacle and the robot was then calculated using the method proposed in II-A. The safety distance thresholds λ and μ were both adopted as 0.1 m. Obstacle collision constraint and self-collision constraint were activated when the corresponding minimum distance fell below λ or μ. The radius of the capsules used to approximate the vicinity of the robot was set at 0.25 m. The CBF-QP mentioned in Eq. 9 was employed as the competing method. Both the proposed QP framework and CBF-QP operated at a frequency of 40 Hz. During the experiment, we recorded the optimization cost time and dual-arm end-effector's Cartesian acceleration. We conducted 10 trials for each method, maintaining the same trajectory. In each trial, the surgeon approached the left and right arms separately between 18 s and 30 s. No visual markers were used during the collision avoidance experiments.

2) Results: The experimental process is illustrated in Fig. 7. When the surgeon approached the dual-arm robot, both arms executed collision avoidance maneuvers while maintaining self-collision avoidance between arms. The experimental results are presented in Fig. 8. Compared to the CBF-QP method, the proposed method required less time for optimization (see Fig. 8(a)), demonstrating superior real-time performance. Fig. 8(b)(c) shows that our method exhibited less jittering during the collision avoidance motion. This indicates that the nonlinear constraints of proposed QP framework effectively ensure the smoothness of the collision avoidance process. Both the proposed method and the CBF-QP approach successfully avoided collisions in all trials. The experimental results demonstrate that the proposed method possesses real-time reactive obstacle avoidance and self-collision avoidance capabilities in dynamic unstructured environments.

D. Robot-Assisted Instrument Delivery Experiment

1) Implementation Details: To evaluate the effectiveness of the dual-arm robot in delivering surgical instruments, we designed experiments based on the actual tasks performed by scrub nurses. Four commonly used surgical instruments (scalpel, tweezer, scissors, and hemostat) were placed on a green sterile drape. The dual-arm robot was required to autonomously grasp and transfer instruments according to

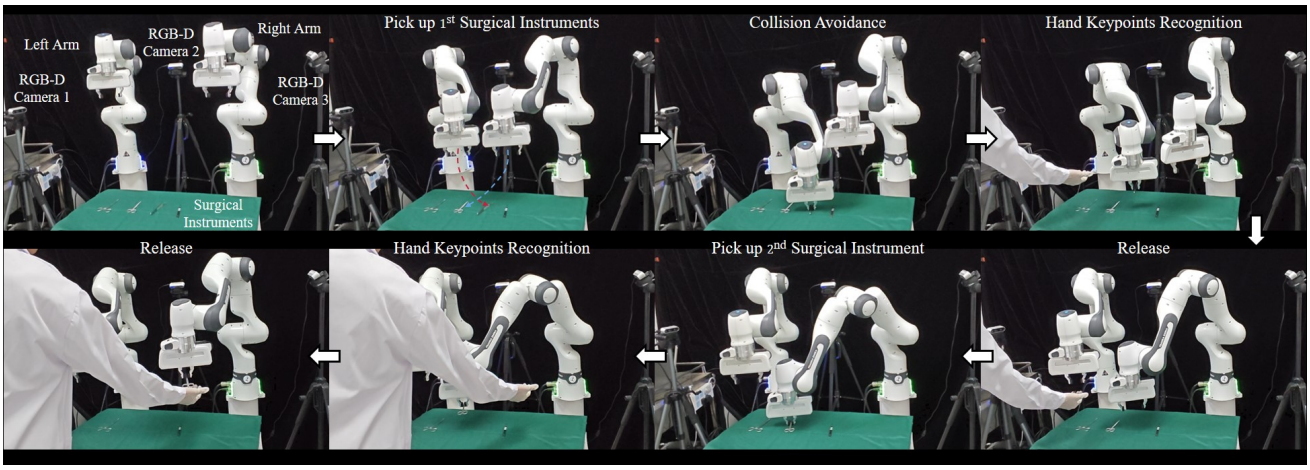

Fig. 9: The experimental process of robot-assisted surgical instrument delivery. The dual-arm robot grasped different surgical instruments in accordance with the surgeon's instructions. The surgical instruments were then transferred to the surgeon's hand in sequence. The entire process was conducted within the proposed QP framework to prevent collisions.

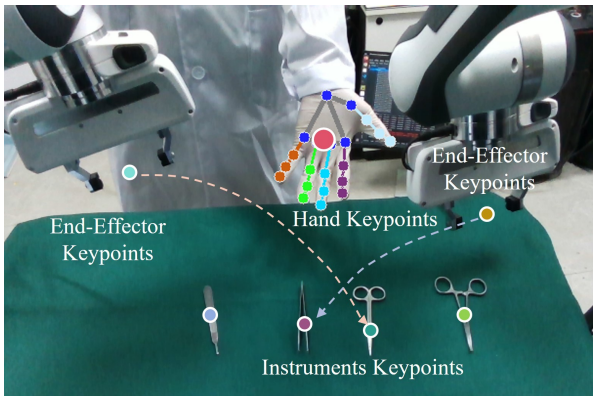

Fig. 10: The keypoints in the experimental process. They represent the 3D positions of the corresponding objects. The arrows indicate the sub-goals of the robot.

the surgeon's instructions, without relying on predefined pathways. By implementing the method proposed in II-C, the sub-goals of the robot were automatically generated by VLM (GPT-4o was applied). These sub-goals were associated to the keypoints in the environment, illustrated in Fig. 10. The keypoints of the surgical instruments were calculated using the approach in II-C, while the keypoints of the dual-arm robot end-effectors were obtained through FK. The average of these landmarks was regarded as the keypoint of the surgeon's hand. The dual-arm robot was tasked with delivering two surgical instruments simultaneously. By testing all 6 possible combinations ($C_4^2 = 6$) of the four instruments 5 times each, a total of 30 trials were performed. This setup ensured that each instrument was tested 15 times in total. The positions of the instruments were randomly arranged in each trial. During the experiments, we recorded the number of collisions to evaluate the safety of the robotic system. We also recorded the number of successful detections, grasps, and deliveries. Then the overall success rate was calculated.

TABLE II: Results of Instrument Delivery Experiment

Type	Collision	Detect	Grasp	Delivery	Success Rate
Scalpel	0	15/15	13/15	13/13	86.67%
Tweezer	0	15/15	15/15	15/15	100.0%
Scissors	0	14/15	12/14	12/12	80.00%
Hemostat	0	14/15	11/14	10/11	66.67%

2) Results: The experimental process for one trial is illustrated in Fig. 9. The dual-arm robot successfully transferred two surgical instruments to the surgeon while avoiding any collisions. For more details of the experiment, please refer to the supplementary video. The experimental results are presented in Table II. No collisions occurred in any of the trials, validating the reliability and safety of the proposed method. In terms of detection, no errors were observed with the scalpel and tweezers, while one error occurred with the scissors and hemostat. This was because the similar shapes of the scissors and hemostat led to misjudgment by the VLM. The grasping of tweezers was completely successful, while the success rates of the other three instruments were 86.67%, 85.71% and 78.57% respectively. The reason for the failed grasps was that these instruments were thin and smooth, which made them difficult to grasp on a flat desktop. The delivery process was entirely successful, except for one instance where the surgeon failed to catch the hemostat. The average success rate across all trials is 83.33%, demonstrating the stability and effectiveness of the robotic system.

IV. Conclusions and Discussions

This paper presents a collision-free dual-arm surgical assistive robot for instrument delivery. The key contribution is the establishment of a robotic system that leverages VLM to intuitively understand surgeon's instructions and autonomously plan movements for grasping and delivering surgical instruments. The dual-arm robotic system operates

in real-time within a QP framework, enabling reactive avoidance of obstacles and self-collision during autonomous motion in the dynamic environment. Simulations and real-world experiments demonstrate that the proposed robotic system is capable of achieving stable and smooth collision-free motion. It can deliver the surgical instruments required by surgeons with an average success rate of 83.33%, indicating the effectiveness.

Compared with robotic scrub nurses in previous studies, the proposed surgical assistive robot eliminates reliance on predefined pathways, significantly enhancing the generalization. The reactive collision avoidance capabilities of the proposed robotic system make it more suitable for dynamic and unstructured surgical environment. Nonetheless, several limitations persist. There is a lack of effective grasping strategies for thin and smooth surgical instruments placed on flat surface. The motion planning relies on the accuracy of keypoints generation and the object recognition by the VLM, as misjudgments can lead to task failure. In future work, we will explore utilizing the VLM as a monitor for evaluating sub-goals objectives, leveraging its latent world knowledge to achieve closed-loop correction of task planning.

REFERENCES

[1] W. Y. Ng, Y. Huang, K. Xie, P. W. Y. Chiu, and Z. Li, "Multimodal robotic surgical instrument transfer and sorting platform: Scrub nurse robot," in *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2024, pp. 1783–1788.

[2] M. G. Jacob, Y.-T. Li, and J. P. Wachs, "A gesture driven robotic scrub nurse," in *2011 IEEE international conference on systems, man, and cybernetics*. IEEE, 2011, pp. 2039–2044.

[3] Y. Xu, Y. Mao, X. Tong, H. Tan, W. B. Griffin, B. Kannan, and L. A. DeRose, "Robotic handling of surgical instruments in a cluttered tray," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 775–780, 2015.

[4] W. Y. Ng, W. Ma, P. A. Heng, P. W. Y. Chiu, and Z. Li, "Large language model-embedded intelligent robotic scrub nurse with multimodal input for enhancing surgeon–robot interaction," *Advanced Intelligent Systems*, p. 2500483, 2025.

[5] M. G. Jacob, Y.-T. Li, and J. P. Wachs, "Surgical instrument handling and retrieval in the operating room with a multimodal robotic assistant," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 2140–2145.

[6] Q. Wu, M. Li, X. Qi, Y. Hu, B. Li, and J. Zhang, "Coordinated control of a dual-arm robot for surgical instrument sorting tasks," *Robotics and Autonomous Systems*, vol. 112, pp. 1–12, 2019.

[7] Y. Xian, X. Zhang, X. Luo, J. Li, L. Zou, K. Xie, J. Li, Y. Li, Y. Huang, D. T. M. Chan, *et al.*, "A semi-autonomous stereotactic brain biopsy robotic system with enhanced surgical safety and surgeon-robot collaboration," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 12, pp. 3288–3299, 2023.

[8] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, H.-S. Fang, *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023.

[9] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 462–12 469.

[10] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[11] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *CoRL*, 2024, pp. 4573–4602.

[12] Y. Chen, W. Li, S. Wang, H. Zhuang, and Q. Wu, "T-rex: Task-adaptive spatial representation extraction for robotic manipulation with vision-language models," *arXiv preprint arXiv:2506.19498*, 2025.

[13] X. Luo, R. Zhang, S. Yang, Z. Sun, R. Zhang, and J. Wang, "Reactive self-collision avoidance for dual-arm robots using a temporal feature modeling and fusion network," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 22 625–22 637, 2025.

[14] X. Luo, S. Yang, H. Xu, C. Lu, and J. Wang, "Cooperative manipulator control based on igh ethercat master," in *2023 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. IEEE, 2023, pp. 109–114.

[15] Y. Xian, Y. Sun, X. Luo, Y. Hu, L. Zou, D. T. M. Chan, D. Y. C. Chan, and Z. Li, "Task automated stereotactic brain biopsy robotic system with clf-cbf-based safety-critical neuronavigation," *IEEE/ASME Transactions on Mechatronics*, 2025.

[16] R. Laha, M. Becker, J. Vorndamme, J. Vrabel, L. F. Figueredo, M. A. Müller, and S. Haddadin, "Predictive multi-agent-based planning and landing controller for reactive dual-arm manipulation," *IEEE Transactions on Robotics*, vol. 40, pp. 864–885, 2024.

[17] M. Koptev, N. Figueroa, and A. Billard, "Real-time self-collision avoidance in joint space for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1240–1247, 2021.

[18] N. F. Mikhail Koptev and A. Billard, "Reactive collision-free motion generation in joint space via dynamical systems and sampling-based mpc," *The International Journal of Robotics Research*, vol. 43, no. 13, pp. 2049–2069, 2024.

[19] S. Wei, R. Khorrambakht, P. Krishnamurthy, V. Mariano Gonçalves, and F. Khorrami, "Collision avoidance for convex primitives via differentiable optimization-based high-order control barrier functions," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2025.

[20] X. Jia, W. Wang, J. Yang, Y. Pan, and H. Yu, "Tompc: Task-oriented model predictive control via admm for safe robotic manipulation," *IEEE Robotics and Automation Letters*, 2025.

[21] S. Zhang and F. Pecora, "Online and scalable motion coordination for multiple robot manipulators in shared workspaces," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 2657–2676, 2024.

[22] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, *et al.*, "Curobo: Parallelized collision-free robot motion generation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8112–8119.

[23] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 4015–4026.

[25] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[26] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3859–3866.

[27] M. Koptev, N. Figueroa, and A. Billard, "Neural joint space implicit signed distance functions for reactive robot manipulator control," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 480–487, 2022.

[28] S. Marangoz, R. Menon, N. Dengler, and M. Bennewitz, "Dawnik: Decentralized collision-aware inverse kinematics solver for heterogeneous multi-arm systems," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023.

[29] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9995–10 001.

[30] X. Ding, H. Wang, Y. Ren, Y. Zheng, C. Chen, and J. He, "Online control barrier function construction for safety-critical motion control of manipulators," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 8, pp. 4761–4771, 2024.