Predictive Multi-Agent based Planning and Landing Controller for Reactive Dual-Arm Manipulation

Riddhiman Laha*[1], Marvin Becker*[2], Jonathan Vorndamme*[1], Juraj Vrabel[1], Luis F.C. Figueredo[1], Matthias A. Müller[2], and Sami Haddadin[1]

***Abstract*—Future robots operating in fast-changing anthropomorphic environments need to be reactive, safe, flexible, and intuitively use both arms (comparable to humans) to handle task-space constrained manipulation scenarios. Furthermore, dynamic environments pose additional challenges for motion planning due to a continual requirement for validation and refinement of plans. This work addresses the issues with vector-field-based motion generation strategies, which are often prone to local-minima problems. We aim to bridge the gap between reactive solutions, global planning, and constrained cooperative (two-arm) manipulation in partially known surroundings. To this end, we introduce novel planning and real-time control strategies leveraging the geometry of the task-space that are inherently coupled for seamless operation in dynamic scenarios. Our integrated multi-agent global planning and control scheme explores controllable sets in the previously introduced *Cooperative Dual Task-Space* and flexibly controls them by exploiting the redundancy of the high Degree-of-Freedom (DoF) system. The planning and control framework is extensively validated in complex, cluttered, non-stationary simulation scenarios where our framework is able to complete constrained tasks in a reliable manner while existing solutions fail. We also perform additional real-world experiments with a two-armed 14 DoF torque-controlled KoBo robot. Our rigorous simulation studies and real-world experiments reinforce the claim that the framework is able to run robustly within the inner loop of modern collaborative robots with vision feedback.**

***Index Terms*—Reactive and sensor-based planning, motion and path planning, dual arm manipulation, collision avoidance.**

I. Introduction

DUAL-ARM robotic manipulation is an underdeveloped research field despite holding significant untapped potential [1], [2]. Cooperative manipulation can overcome the limitations of a single arm when it comes to, for instance, handling heavy, large or bulky payloads. Nevertheless, the advantages of multi-robot collaboration come at the cost of increased complexity. Bimanual manipulation requires more than the sum of unimanual tasks [2]–[4]. Indeed, dual-arm manipulation tasks, as shown in Fig. 1, require intrinsic spatial and temporal coordination while adhering to closed-chain geometric constraints [5]. These additional complexities are reflected in the modeling, planning, and control of the coordinated multi-contact tasks. As a result, dual-arm manipulation in the real world is often prone to failure, particularly in human-centric activities. These often take place in unstructured and constrained environments and are subjected to desired and undesired contact forces.

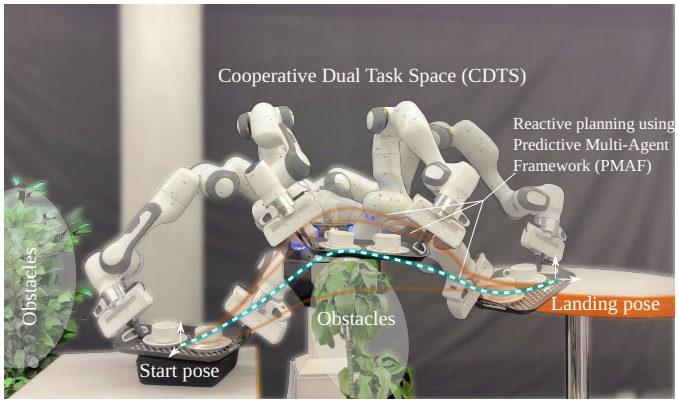


Fig. 1: Our globally informed multi-agent planner guides the constrained dual-arm cooperative system through the absolute frame (located between the two end-effectors) towards the landing (goal) location on the table whilst avoiding obstacles in the scene. Throughout motion execution, tasks like joint limits avoidance, end-effector tilt control, and safe interaction control are utilized flexibly in order to successfully complete the complex manipulation task.

In real-world scenarios, achieving autonomy and safe dual-arm manipulation requires the ability to plan reactive collision-free maneuvers and trajectories in real-time. Notwithstanding, classical sense-plan-act architectures which form the current state-of-the art in motion-planning – mostly due to the success of sampling-based global planners in the exploration of collision-free space – do not meet the demand for reactiveness, especially for higher DoFs systems [6]–[8]. On the other end, all existing scarce results driven by reactive planners for dual-arm manipulation, more suitable for real-time, fully rely on a single vector-field solution, which considerably reduces their stability in addressing different scenarios. Single vector-field dynamics cannot account for the often competing multi-objectives in cluttered dynamic scenes. These might be distance to obstacles, path-length, and remaining distance to goal among others.

In this study, instead, we propose a multi-agent exploration approach that plans virtual trajectories using various maneuvering strategies. This intelligent look-ahead solution facilitates robust and adaptive performance at all levels and enhances planner reliability in finding feasible paths in different scenarios. This approach is closely connected to a hybrid Cooperative Set-based Task Priority control (CoSTP) framework, which better integrates reactive motion with multiple constraints of varying degrees. Our strategy relaxes and prioritizes constraints, which is critical for effective dual-arm manipulation, and therefore improves responsiveness and efficiency of the resulting planner.

Take, for example, the cooperative task in Fig. 1. The Predictive Multi-Agent Framework (PMAF) generates a globally sub-optimal trajectory, which is further refined and relaxed by the CoSTP. This combination leads to, for instance, reduced

*Equal contribution.

This work was partially funded by the State of Bavaria for the Geriatronics project, the Region Hannover in the project *roboterfabrik*, the Alfried Krupp von Bohlen und Halbach Foundation, and KoBo34 by the BMBF (grant 16SV7985). Please note that S. Haddadin has a potential conflict of interest as shareholder of Franka Emika GmbH. *Corresponding author: Marvin Becker.*

[1]The authors are with Munich Institute of Robotics and Machine Intelligence, Technische Universität München (TUM), 80992 Munich, Germany (e-mail: {riddhiman.laha,jonathan.vorndamme,luis.figueredo,sami.haddadin}@tum.de, vrabel.juraj@icloud.com).

[2]The authors are with the Institute of Automatic Control, Leibniz University Hannover, Germany (e-mail: {becker,mueller}@irt.uni-hannover.de).

DoFs collision-free funnel constraints – around the desired trajectory rather than the entire trajectory – and tray-cup tilt safety margin constraints – around the z-axis. Conversely, the CoSTP also provides cooperative manipulability and joint-space information to the PMAF, which reshapes and refines the vector fields with informed cooperative manipulation capabilities of the robot. Our proposed integrated planning and control solution also tackles geometric uncertainties and unwanted or desired contacts on the load as in the final placement in Fig. 1. For successful placement of objects, consistency in the contact surface is critical to avoid failure and build-up of unexpected stresses. Our hybrid approach incorporates stabilizing features of the desired contact within CoSTP enhancing admittance in load orientation – similar to landing-gear shock-absorbing systems in airplanes that improves robustness in planning.

This work is the first to systematically integrate multi- agent planning for dual-arm reactive motion generation and control. In our initial investigations in this topic [8]–[10], we showed that vector-field based strategies were able to guide both single and dual-arm systems to the goal while staying within safe operation bounds. In this article, we present extensions and more in-depth insights from practical experiments. More specifically, we comprehensively combine and reshape results from [10] and [9], and extend them in a complementary fashion to improve decision-making of multi-agent planners grounded on its cooperative capabilities with prioritized dual-arm control, including contact-force-based skills such as safe landing in partly known environments. The proposed real-time solution raises the potential for realistic use cases as shown later. Our contributions extend dual-arm manipulation literature through:

- Predictive virtual multi-agent exploration with various maneuvering strategies grounded with informed cooperative manipulation capabilities leading to suboptimal real-time obstacle avoidance;
- Flexible prioritized task switching strategy that adapts to different scenarios in the reduced workspace leading to a refined and relaxed solution;
- Integrated sensed contact wrenches into planning and control framework to both evade undesired contacts as well as to explore stabilizing features of desired environment contact;
- Elaborate simulation and real-world studies validating the effectiveness and reliability of our approach and providing insights of general reactive planning for dual-arm setups;
- Our ROS based software framework is available for the community and can be accessed through [11].

II. Related Work

Dual-arm manipulation has a prolific research history. Due to the recent efforts of including anthropomorphic benefits [12]–[16], dual-arm manipulation has gained more traction in the research community [17]. Based on the literature, bimanual manipulation can be broadly divided into un-coordinated and coordinated systems [18]–[23]. Furthermore, coordinated dual-arm manipulation can be categorized into (a) goal-coordinated and (b) bimanual/cooperative systems. In this work, we will focus on the more challenging and constrained problem of coordinated systems, i.e., both manipulators tightly coupled are working towards one common task. More specifically, we are interested in coordinated cooperative manipulation where both the arms cooperate with each other in lifting or constrained tasks like moving the cup-tray system as depicted in Fig. 1. In this regard, the authors in [24] and [25] consider the two robots as a cooperating system and formulate the absolute and relative Jacobians to map task space velocities to the joint space for trajectory generation. This was further developed in [26] where task Jacobians where presented in terms of Jacobians of the single manipulators. In this work, we take advantage of the cooperative scheme in the dual quaternion domain, first introduced in [27]. This work is further extended to address switchings during tracking of existing trajectories in [28]. However, most of these works do not consider local or global planning in the presence of obstacles in the scene. To the best of the authors' knowledge, this is the first work that connects cooperative manipulation to reactive planning strategies and safe interaction control.

Sampling based offline planners [29], [30] like Rapidly exploring Random Trees (RRTs) [31], [32] have been quite popular in the motion planning community in the last decades. This can be attributed to their ability to handle high degrees of freedom and trivial implementation. However, the classic sense-plan-act strategy is not suitable for effective collaborative manipulation [6], [7] and generating robot motions in fast changing environments. In spite of efficient newer implementations [33]–[36], solutions still tend to have significant computational costs [37], [38], need non-trivial sampling in constrained manifolds as in the case of cooperative manipulation, and suffer from narrow passage problems [39], [40].

As reactive motion generation methods are intrinsically connected to control, it provides faster interfaces to integrate with controllers operating in rapidly changing environments [41]. The seminal work in the area took inspiration from electrostatic phenomena to conceive artificial potential field (APF) that steered the robot (usually modelled as a point mass at a specific point of interest) to the desired goal while avoiding obstacles through repulsive fields [42]. The concept was further extended to multi-DoF robot manipulators and mobile robots. Enhanced local-minima-free strategies, e.g., Harmonic and navigation functions were introduced to generate more traction [43]. Similarly, circular fields (CFs) was first proposed in [44], [45] with the aim to allow obstacle avoidance with a defined rotation (local-minima free) model, inspired by Lorentz forces and electro-magnetic fields, instead of a repulsive force as with the classical APF approach. In order to address the problem of consistent artificial current flow for each obstacle and make further improvements for robust operation in partially known environments, the design of the artificial obstacle currents was changed in [46]. Recently, a meticulous mathematical analysis of CF motion planning was conducted in [47], where we were able to derive guarantees for collision avoidance and goal convergence under clearly defined conditions. Yet, different approaches were also proposed in the recent past like modeling the magnetic field using Maxwell's equations [48] and Gyroscopic Forces Method (GFM) for pushing the limits of vector field-based motion generation strategies.

Another issue that needs to be treated in this regard is scaling to robots with many DoFs, e.g., humanoids and cooperative robots, which are manipulating along increasingly complex

environments and constraints. Among the few existing works, most rely on a hybrid method of offline computation with real-time reactiveness [49], [50]. Nevertheless, research efforts in the recent past have been targeted towards collision avoidance in human-centric environments by combining vector fields and real-time optimization [51]. Danger fields have also been proposed for safe human-robot coexistence which take into account the Cartesian velocity for a specific point of interest on the robot and change the influence of the vector-field based on the heading direction [52], [53]. It is further combined with a feedback based low level joint-space controller including impedance regulation which is rendered passive by design [54]. This ensures collision avoidance and safety in stationary as well as non-stationary human robot interaction (HRI) scenarios.

However, most of the frameworks are still susceptible to local minima [55], and not connected to global path planning which we consider. Furthermore, handling task constraints and scaling up to bimanual tasks is something that was out of the scope of most of the work in the literature. To leverage the benefits of fast reactive vector-field like approaches and sampling based planning, researchers proposed computationally efficient guided vector field methods with reachability sets to induce global information [56]. To further enhance the idea of injecting global information in the planning framework, in line with our ongoing quest, we generate globally sub-optimal paths with locally reactive CF forces to avoid obstacles in multiple directions [10]. Moreover, we were able to extend the results to the full structure of single arm manipulators in [57]. Regardless, dual-arm manipulation and interaction control was not analyzed.

TABLE I: DESCRIPTION OF MAIN NOTATIONS AND ABBREVIATIONS.

Notation	Corresponding Description
$\underline{\boldsymbol{x}}$	$\triangleq$ Unit dual quaternion
$\underline{\boldsymbol{x}}^*$	$\triangleq$ Dual quaternion conjugate
$\mathcal{P}(\underline{\boldsymbol{x}})$	$\triangleq$ Primary part of $\underline{\boldsymbol{x}}$
$\mathcal{D}(\underline{\boldsymbol{x}})$	$\triangleq$ Dual part of $\underline{\boldsymbol{x}}$
$\underline{\boldsymbol{x}}_a$	$\triangleq$ Absolute pose
$\underline{\boldsymbol{x}}_r$	$\triangleq$ Relative pose
$\underline{\boldsymbol{S}}$	$\triangleq$ Set of all dual quaternions
$\overset{+}{\boldsymbol{H}}, \overset{-}{\boldsymbol{H}}$	$\triangleq$ Hamilton operators
vec_8	$\triangleq$ Mapping from $\underline{\boldsymbol{S}} \to \mathbb{R}^8$
$\boldsymbol{q}$	$\triangleq$ Stacked joint positions $\in \mathbb{R}^{14}$
$\boldsymbol{J}_{\underline{\boldsymbol{x}}_a}$	$\triangleq$ Absolute pose jacobian
$\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$	$\triangleq$ Relative pose jacobian
$\boldsymbol{J}_{\phi_\chi}$	$\triangleq$ EE tilt constraint jacobian
$\boldsymbol{J}_{q_i}$	$\triangleq$ Joint limits avoid. jacobian
$\boldsymbol{J}_{\mathrm{g,rot}}$	$\triangleq$ Safe landing jacobian
$\boldsymbol{P}$	$\triangleq$ Nullspace projector
$\mathbf{F}_c$	$\triangleq$ Circular Field (CF) force
$\mathbf{F}_g$	$\triangleq$ Attractive goal force
$\mathbf{F}_d$	$\triangleq$ Damping force
$\mathbf{F}_\sigma$	$\triangleq$ Manipulability guidance force
$\boldsymbol{c}$	$\triangleq$ Artificial current vector
$\boldsymbol{b}$	$\triangleq$ Magnetic field vector
$\boldsymbol{n}$	$\triangleq$ Surface normal vector
$\boldsymbol{v}$	$\triangleq$ Normalized velocity
$\boldsymbol{p}_a$	$\triangleq$ Agent pose (translation)
$\hat{\boldsymbol{p}}_a$	$\triangleq$ Desired agent pose
r_r	$\triangleq$ Radius control point approximation
r_d	$\triangleq$ Radius detection shell
$\mathbb{O}$	$\triangleq$ Set containing obstacle info
$\wedge$	$\triangleq$ Logical And operator
$\tilde{\mathfrak{F}}$	$\triangleq$ Evaluation function for agents

III. PROBLEM FORMULATION

In this work, we are interested in finding a valid path $\mathbb{P}$ in the free task space[1] $\mathcal{T}^c \subset \mathrm{Spin}(3) \ltimes \mathbb{R}^3$ from an initial pose $\underline{\boldsymbol{x}}_0$ to a goal pose $\underline{\boldsymbol{x}}_g$ such that the goal pose matches the pose $\underline{\boldsymbol{x}}_l$ of the landing surface. Let $\mathcal{T}_{\mathrm{obs}}$ denote the infeasible set of task space regions occupied by obstacles. Thus, $\mathcal{T}^c$ can be formally represented as $\mathcal{T}^c = \mathcal{T} \setminus \mathcal{T}_{\mathrm{obs}}$, where $\mathcal{T}$ stands for the task space of the robotic system. Table I presents a list of the standard notations used throughout the manuscript.

Furthermore, during task execution, we aim to satisfy multiple geometric cooperative constraints, e.g., keeping a tight grasp on the object while moving the cup-tray system, as shown in Fig. 1, within a safe and acceptable set of orientation deviations. The two end-effectors with a specific relative orientation between them along with the shared load form the Cooperative Dual Task Space (CDTS) (see Sec. V).

Mathematically, this can be encoded by using different task primitives. By primitives we mean useful task definitions (fundamental geometric relationships/building blocks) as illustrated in Fig. 1. These primitives could be defined in the task space as well as in the joint space of the robotic system. As an example, consider the task of holding the relative position and orientation between the two end-effectors. Another primitive would be the task of controlling the tilt of the tray during the motion. Lastly, the task of keeping all the joints within the hardware limits is an example of a primitive defined in the joint space. We denote the general set of possible task primitives in the free workspace of the manipulator by $\mathfrak{S} \subset \mathcal{T}^c$. The controlled set of geometric structures is designated by $\mathfrak{R}$, which is essentially compositions of subsets of $\mathfrak{S}$ (c.f. Definition 2, in Section VI), corresponding to a different cooperative primitive constraint that must be satisfied during the cooperative task. For concrete formalization of primitives we direct readers to Sec. V.

Now, let us suppose there are n_o obstacles in the scene and n_a virtual explorer agents that traverse in $\mathcal{T}^c$. Herein, we assume that the virtual multi-agents have single-integrator dynamics (velocity controlled) with a maximum acceleration[2]. We do not assume that the explorer agents exhibit pure *flocking dynamics*. That is, they move towards a common target but do not necessarily stay close enough to each other. Indeed, due to their intrinsic modeling, each exhibits a different exploration strategy resulting in n_a distinct agent paths $\mathbb{P}_{a,j} \subset \mathcal{T}^c$ with $j = 1, \ldots, n_a$. All agents are evaluated using a cost function $\tilde{\mathfrak{F}} : \mathbb{R}^{3\times n} \to \mathbb{R}^+$. The goal is to exploit the different agent behaviors to select the sub-optimal path to the target (goal) $\underline{\boldsymbol{x}}_g$ with an intuitive evasion strategy such that the dual-arm

[1]Herein, all the concepts utilize dual-quaternion algebra and unit dual quaternion representation $\mathrm{Spin}(3) \ltimes \mathbb{R}^3$ due to its intrinsic advantages (e.g., translation and attitude coupling, singularity-free representation and significant computational efficiency [58]–[60]) over other representations. Moreover, (unit) dual quaternions can be used to describe rigid body motions, twists, wrenches, and multiple geometric primitives, e.g., Plücker lines, planes, cylinders, cones, in a straightforward manner [61], [62]. They can also be used to describe and easily extract geometric primitives even in complex and constrained cooperative spaces [61], [63]. For more details, please see Appendix A.

[2]This is in accordance with the maximum feasible acceleration of the system.

system is able to place the object in an area around the landing location $\underline{\boldsymbol{x}}_l$ whilst still respecting the task-space constraints within $\mathfrak{R}$.

Assumption 1. *The explorer agents have information on the obstacle's dynamics, that is, we can obtain the obstacle's position and velocity up to a certain accuracy.*

This assumption is reasonable because current state-of-the-art computer vision algorithms can perform pose estimation and real-time object tracking, as shown in [64]. The obstacles herein are modeled as spheroids, as any object (both convex and non-convex cross sections) can be approximated by ellipsoids [65]. The spheres can be considered as trivial cases for many objects, however, problems might arise when an object is more elongated along one dimension than the other. In such scenarios, a spheroid or an ellipsoid is certainly a better approximation choice [66]. Formally, we seek to solve the following:

Problem Statement 1. *Given a desired control point of the robot with the initial position $\underline{\boldsymbol{x}}_0$, the goal position $\underline{\boldsymbol{x}}_g \approx \underline{\boldsymbol{x}}_l$ and a set of obstacles $\mathcal{T}_{\text{obs}}$, find a continuously differentiable path $\mathbb{P} \in C^1([0,T], \mathcal{T}^c)$ such that $\mathbb{P}(0) = \underline{\boldsymbol{x}}_0$, $\mathbb{P}(T) = \underline{\boldsymbol{x}}_g \approx \underline{\boldsymbol{x}}_l$, and $B_r\big(\mathbb{P}(t)\big) \cap \mathbb{O}(t) = \emptyset, \forall \mathbb{O}(t) \subseteq \mathcal{T}_{\text{obs}}, \forall t \in [0,T]$, while still satisfying the hard task space constraints $\mathfrak{R}$, joint constraints, and ensuring $\mathfrak{F}^*(\mathbb{P}) = \min_{j \in [1,n_a]} \tilde{\mathfrak{F}}(\mathbb{P}_{a,j})$.*

Here, $C^1([0,T], \mathcal{T}^c)$ denotes all differentiable functions from $[0,T]$ to $\mathcal{T}^c$ whose derivative is continuous, and $B_r\big(\mathbb{P}(t)\big)$ is the ball neighborhood of radius r_r, that is, a spherical approximation of the space occupied by the control point. Thus, $B_r\big(\mathbb{P}(t)\big) \cap \mathbb{O}(t)$ denotes the intersection of the inflated robot path with an obstacle at time t. More specifically, in this work, we aim to encapsulate global planning capabilities along with a local collision avoidance strategy to execute constrained manipulation planning in clutter. Since the gap between planning and control is greatly reduced, the planner is further fed back with information that enables safe HRI.

IV. Reactive Multi-Agent based Motion Generator

This section introduces the reactive motion generation scheme for the reference trajectory of cooperative manipulators. First, we describe the vector field shaped by the different forces, including Circular Field (CF) forces for obstacle avoidance and attractive forces for goal convergence. Second, we illustrate our predictive multi-agent framework that we employ with different force guidance strategies to improve the overall motion generation performance. For the generation of the reference trajectory, we use the following point mass dynamics,

$$m\ddot{\hat{\boldsymbol{p}}}_a = \mathbf{F}_{\text{g}} + \mathbf{F}_{\text{cf}}, \tag{1}$$

where $\ddot{\hat{\boldsymbol{p}}}_a \in \mathbb{R}^3$ is the desired acceleration of the robot control point, m denotes its virtual mass, and $\mathbf{F}_{\text{cf}}$ and $\mathbf{F}_{\text{g}}$ are the CF and the attractive goal force described in the following.

A. Attractive Force

We use the velocity limiting control force from [42] as the virtual attractor dynamics in the form

$$\mathbf{F}_{\text{g}} = -k_d \left(\dot{\boldsymbol{p}}_a - \nu \dot{\boldsymbol{p}}_g \right), \tag{2}$$

with the artificial desired goal velocity

$$\dot{\boldsymbol{p}}_g = \frac{k_a}{k_d} \left(\boldsymbol{p}_g - \boldsymbol{p}_a \right). \tag{3}$$

Here, $\boldsymbol{p}_a \in \mathbb{R}^3$ is the robot position, $\boldsymbol{p}_g \in \mathbb{R}^3$ is the goal position, $k_d > 0$ and $k_a > 0$ are scaling factors and $\nu = \min\left(1, \frac{v_{\max}}{\|\dot{\boldsymbol{p}}_g\|}\right)$ is used to limit the force when the robot velocity reaches a defined maximum magnitude $v_{\max}$ in the direction of the goal. This definition of the attractive force leads to improved trajectories compared to the standard definition of a potential field as it results in the robot being attracted in a straight line towards the goal position while being constrained in its velocity magnitude.

B. Circular Field Force for Collision Avoidance

In order to reactively avoid obstacles, we use CFs which are inspired by the laws of electromagnetism [67] and were first presented in [44]. In contrast to classical artificial potential fields, CF forces do not have local minima [44], do not change the velocity magnitude of the robot, and do not apply a force when the robot is moving parallel to an obstacle surface. Furthermore, CF motion planning guarantees goal convergence and obstacle avoidance under the assumptions defined in [47].

In the CF approach, the robot is modeled as a charged particle moving in artificial electromagnetic fields and is subject to the following modified version of the Lorentz force

$$\mathbf{F}_{\text{cf}} = \frac{k_{\text{cf}}}{\|\boldsymbol{d}\| - r_r} \frac{\dot{\boldsymbol{d}}}{\left\|\dot{\boldsymbol{d}}\right\|} \times \mathbf{B}, \tag{4}$$

where $k_{\text{cf}} > 0$ is a constant gain, and $\boldsymbol{d} = \boldsymbol{p}_o - \boldsymbol{p}_a$ denotes the distance vector from the robot control point to the closest point on the obstacle surface $\boldsymbol{p}_o$. Every obstacle creates an artificial magnetic field

$$\mathbf{B} = \boldsymbol{c} \times \dot{\boldsymbol{d}} \tag{5}$$

with artificial current $\boldsymbol{c}$. The artificial current defines the direction of the magnetic field, and thus, the direction of the CF force, that is, it defines the direction in which an obstacle is avoided. Therefore, it is a crucial component in the motion planning approach, and its choice is discussed in detail in subsection IV-F. Additionally, we define a *detection shell* with radius r_d around each obstacle, so that the robot reacts to an obstacle only when it enters its detection shell. In this way, obstacles further away are neglected, which reduces the resulting computational load. To further improve the computational efficiency, we ignore obstacles that move away from the robot, i.e., $\boldsymbol{d} \cdot \dot{\boldsymbol{d}} > 0$ and are simultaneously not between the robot and the goal, i.e., $\boldsymbol{d} \cdot \left(\boldsymbol{p}_g - \boldsymbol{p}_a\right) < 0$. Moreover, obstacle avoidance should have priority over goal convergence. Therefore, we ensure that the attractive force does not interfere with the CF forces by introducing the following scaling factors, which were partly inspired from [68]. The first term reduces the attractive force when the robot comes close to an obstacle

$$k_{g_{\text{cl}}} = 1 - \exp\left(-\frac{\|\boldsymbol{d}\|}{r_d}\right). \tag{6}$$

The second term reduces the attractive force when an obstacle is between the robot and the goal position and increases the force otherwise

$$k_{g_{\text{oc}}} = \left(1 - \left(\frac{\left(\boldsymbol{p}_g - \boldsymbol{p}_a\right) \cdot \boldsymbol{d}}{\left\|\boldsymbol{p}_g - \boldsymbol{p}_a\right\| \|\boldsymbol{d}\|}\right)\right)^2. \tag{7}$$

For computational efficiency, both factors are only implemented for the closest obstacle.

CFs by themselves are local minima free as shown in [47]. However, the combination with the virtual attractor dynamics could induce new local minima. Thus, we deactivate the attractive force when it would decelerate the control point velocity below a defined limit $v_{\min}$. That is, when it is not in the vicinity ξ of the goal position, the following aggregate goal scaling factor is induced combining $k_{g_{\mathrm{cl}}}$ and $k_{g_{\mathrm{oc}}}$ as

$$k_g = \begin{cases} 0 & \text{if } \dot{\boldsymbol{p}}_a \cdot \mathbf{F}_{\mathrm{g}} \leq 0 \wedge \|\dot{\boldsymbol{p}}_a\| \leq v_{\min} \wedge \left\|\boldsymbol{p}_g - \boldsymbol{p}_a\right\| > \xi \\ k_{g_{\mathrm{cl}}} k_{g_{\mathrm{oc}}} & \text{otherwise.} \end{cases} \tag{8}$$

In the presence of multiple obstacles, the forces of all obstacles are superposed. These combined measures result in the following control point dynamics

$$\ddot{\hat{\boldsymbol{p}}}_a = k_g \mathbf{F}_{\mathrm{g}} + \sum_{i=1}^{n_{\mathrm{o}}} \mathbf{F}_{\mathrm{cf}_i}, \tag{9}$$

where n_{o} is the number of obstacles.

C. Repulsive Potential Field Force for Self-Collision Avoidance

To avoid self-collisions, we used the classical APF forces from [42] and approximated the structure of the robot via spherical obstacles

$$\mathbf{F}_{\mathrm{r}} = k_r \boldsymbol{d} \left(\frac{\frac{1}{\|\boldsymbol{d}\|} - \frac{1}{r_d}}{\|\boldsymbol{d}\|^2} \right), \tag{10}$$

where k_r is the scaling factor of the repulsive force, which is applied only if the control point of the robot is within the detection shell radius r_d of the respective obstacle[3]. Similar to the CF forces, the repulsive forces of all n_{sc} self-collision obstacles are superposed, leading to the following updated control point dynamics

$$\ddot{\hat{\boldsymbol{p}}}_a = k_g \mathbf{F}_{\mathrm{g}} + \sum_{i=1}^{n_{\mathrm{o}}} \mathbf{F}_{\mathrm{cf}_i} + \sum_{j=1}^{n_{\mathrm{sc}}} \mathbf{F}_{\mathrm{r}_j}. \tag{11}$$

D. Cooperative Guidance Strategies based on Artificial Forces

In order to extend the capabilities of the resulting vector-field planner and make it robust to issues such as manipulability, undesired contact, and joint limits, we introduce additional task space forces that steer the cooperative system to more favorable directions. First, we illustrate how undesirable singular configurations can be avoided locally during motion execution by exploiting the singular values of the motion Jacobian. Next, we describe the concept of repulsive contact wrenches. Finally, we demonstrate how the motion policy can be further revamped by leveraging the joint bounds of the system and designing artificial forces that reject configurations closer to the limits.

CDTS Manipulability Force Guidance: In contrast to the standard circular field approach, the integration of reactive techniques into the Cooperative Dual Task Space (CDTS) framework requires further knowledge of the existing constraints. The planning design must ensure that undesirable configurations that could degrade performance are avoided, e.g., joint limits and singular configurations. Even though we can explore task-priority and the CDTS self-motion to this aim, as we will show in the following subsection, in this work, we propose a strategy to leverage such information to the circular field. This allows for the motion generation to account for the CDTS constraints in joint space and, consequently, to guide the desired motion through regions of larger manipulability.

First, note that the reactive motion should be integrated into either the absolute or relative position cooperative primitives as described in (43). Details on the integration follow in Section V as the task execution is defined. Now, from the desired motion Jacobian, consider the singular value decomposition of $\boldsymbol{J}_{\boldsymbol{p}_\chi}$,

$$\boldsymbol{J}_{\boldsymbol{p}_\chi} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \sum\nolimits_i \sigma_i \boldsymbol{w}_i \boldsymbol{v}_i^T,$$

where $\boldsymbol{U} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \boldsymbol{w}_3]$, $\boldsymbol{V} = [\boldsymbol{v}_1\ \cdots\ \boldsymbol{v}_n]$, are orthogonal matrices and $\boldsymbol{S} = [\mathrm{diag}\,(\sigma_1, \cdots, \sigma_{\min})\ \ \mathbf{0}]$. Taking the output singular vector $\boldsymbol{w}_{\min}$ associated with the singular minimum value $\sigma_{\min}$, we can design a repulsive field that guides the system away from poor manipulability regions, that is, from regions where the resulting reactive motion would require a larger motion in joint-space. Hence, the updated vector field becomes,

$$m\ddot{\hat{\boldsymbol{p}}}_a = k_g \mathbf{F}_{\mathrm{g}} + \sum_{i=1}^{n_{\mathrm{o}}} \mathbf{F}_{\mathrm{cf}_i} + \sum_{j=1}^{n_{\mathrm{sc}}} \mathbf{F}_{\mathrm{r}_j} + \mathbf{F}_\sigma \tag{12}$$

$$\mathbf{F}_\sigma = \lambda \boldsymbol{w}_{\min}, \tag{13}$$

where $\mathbf{F}_{\mathrm{g}}$ is the attractor force, $\mathbf{F}_{\mathrm{cf}}$ is defined in (4), and λ is a gain defined similar to [69], however with positiveness defined in order to build an opposite force to the direction of $\mathbf{F}_{\mathrm{cf}} + \mathbf{F}_{\mathrm{g}}$.

$$\lambda = \begin{cases} (1 - \frac{\sigma_{\min}}{\epsilon_\sigma})\lambda_{\max}, & \text{if } \sigma_{\min} < \epsilon_\sigma; \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $\lambda_{\max} > 0$ and $\epsilon_\sigma > 0$ defines the size of the singular region we want our force to act. Notice that outside the region, the repulsive force is void. Furthermore, the motion Jacobian can also be weighted by the distance to joint limits to leverage such information to the resulting force field.

The force $\mathbf{F}_\sigma$ adds repulsive dynamics that drives the robot to directions further away from the task space vector associated with the minimum singular value. In long horizon tasks where large workspace distances need to be traversed, (13) can explore a fixed value of $\boldsymbol{w}$ which is then updated whenever the projection of the vector over $\boldsymbol{w}_{\min}$ is higher than a given angle threshold.

CDTS Joint-Limit Force Guidance: Similarly to the CDTS manipulability force guidance, which aims to "push" the trajectory to regions of the cooperative workspace with higher manipulability, in this subsection we also propose a novel virtual force guidance in task-space aiming to improve motion generation from a joint-limit perspective.

The main concept is the following. We take motions in the joint space leading to joint-limit avoidance, e.g., a first-order system $\dot{\boldsymbol{q}}_{jg} = \boldsymbol{q}_c - \boldsymbol{q}$, where $\boldsymbol{q}_c$ is the center joint position. This leads to large velocities whenever the joint position is far away from the center, and smaller ones otherwise. This motion is mapped to task-space motion through a task-space velocity Jacobian $\boldsymbol{J}_{\boldsymbol{p}_\chi}(\boldsymbol{q})$. Then, we build a virtual force proportional to this motion, i.e., $\mathbf{F}_{\mathrm{j}} \approx \dot{\boldsymbol{q}}_{jg}$, that generates accelerations in

[3]This was crucial in scenarios with the additional KoBo setup (see sec. IX), where we placed a virtual repulsive sphere in the center of the robot.

such direction with a proportional gain, that is,

$$\mathbf{F}_{\mathrm{j}} = k_{\mathrm{jg}} \frac{\boldsymbol{J}_{\boldsymbol{p}_\chi}(\boldsymbol{q}_c - \boldsymbol{q})}{\|\boldsymbol{J}_{\boldsymbol{p}_\chi}(\boldsymbol{q}_c - \boldsymbol{q})\|}, \tag{15}$$

where k_{jg} is a scalar gain to prescribe a given behaviour, e.g., an hyperbolic tangent in the proximity of joint-limits, an inertia-dependent gain, among others. In this work, we consider a proportional gain scaled by distance to the center joint position, that is,

$$k_{\mathrm{jg}} = \overline{k}_{\mathrm{jg}} \tfrac{1}{q_r} \max_{i=1,..n} dz\Big(\tfrac{\|q_{i,c} - q_i\|}{q_{i,\mathrm{range}}}, 1 - q_r \Big), \tag{16}$$

where $dz(a,b)$ is a dead-zone correction given by

$$\mathrm{dz}(a,b) := \begin{cases} a+b, & a < -b, \\ 0, & |a| < |b|, \\ a-b, & a > b, \end{cases} \tag{17}$$

with q_r being the percentage of the range of motion, i.e., $q_{i,\mathrm{range}}$ for all $i-$joints, the forces will start to act to avoid limits. For instance, when $q_r = 0.5$, the $\mathbf{F}_{\mathrm{j}}$ is always zero until one of the joints reaches the outer 50% of its range. The upper-bound $\overline{k}_{\mathrm{jg}}$ defines the force intensity and acts as a stiffness gain for values above the deadzone.

Finally, the updated force field becomes

$$m\ddot{\hat{\boldsymbol{p}}}_a = k_g \mathbf{F}_{\mathrm{g}} + \sum_{i=1}^{n_{\mathrm{o}}} \mathbf{F}_{\mathrm{cf}_i} + \sum_{j=1}^{n_{\mathrm{sc}}} \mathbf{F}_{\mathrm{r}_j} + \mathbf{F}_\sigma + \mathbf{F}_{\mathrm{j}}. \tag{18}$$

Although we do not consider the concept of *repulsive contact wrenches* in this work, as shown in our previous study [9], other virtual guidance forces similar to the ones mentioned can be integrated straightforwardly in this framework.

E. Reference Trajectory Calculation

For the calculation of the reference trajectory, we use the previously calculated desired control point acceleration and point mass dynamics with bounds defined by the maximum velocity $v_{\max}$ and maximum acceleration $a_{\max}$, i.e.,

$$\dot{\hat{\boldsymbol{p}}}_a(t+1) = k_{vb} \left(\dot{\hat{\boldsymbol{p}}}_a(t) + k_{ab} \ddot{\hat{\boldsymbol{p}}}_a(t) T_c \right), \tag{19}$$

$$\hat{\boldsymbol{p}}_a(t+1) = \hat{\boldsymbol{p}}_a(t) + \dot{\hat{\boldsymbol{p}}}_a(t) T_c + k_{ab} \ddot{\hat{\boldsymbol{p}}}_a(t) \frac{T_c^2}{2}, \tag{20}$$

where

$$k_{vb} = \begin{cases} 1 & \text{if } \left\| \dot{\hat{\boldsymbol{p}}}_a(t) \right\| \le v_{\max}, \\ \frac{v_{\max}}{\left\| \dot{\hat{\boldsymbol{p}}}_a(t) \right\|} & \text{otherwise}, \end{cases} \tag{21}$$

$$k_{ab} = \begin{cases} 1 & \text{if } \left\| \ddot{\hat{\boldsymbol{p}}}_a(t) \right\| \le a_{\max}, \\ \frac{a_{\max}}{\left\| \ddot{\hat{\boldsymbol{p}}}_a(t) \right\|} & \text{otherwise}, \end{cases} \tag{22}$$

and T_c is the step time for the control command calculation.

F. Predictive Multi-Agent Framework

One of the limitations of traditional CFs, which is predominantly a vector field based planning approach, is the omission of global environment information. Thus, the application generally leads to globally suboptimal paths and it is possible to design trap scenarios where the point-mass agent can get stuck. Note that this is not equivalent to the local minima of potential field approaches as the robot does not stop moving but is trapped in limit cycles which can be avoided with appropriate parametrization. Therefore, we employ our predictive multi-agent exploration approach, the circular field predictions (CFP), which was introduced in our previous publication [10].

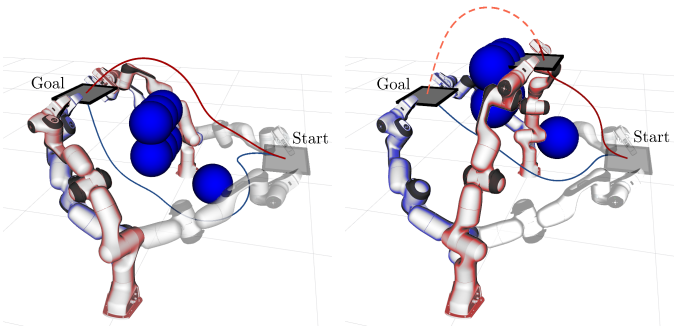
Our method allows for the generation of multiple trajectories towards the goal pose by different virtual agents, each characterized by a specific parameter set $\tilde{\mathfrak{P}}$ including, e.g. the circular field gain, the artificial magnetic field vectors, the maximum velocity and the safety margin. Additional parameters can be incorporated depending on the environment under consideration. In the planning stage, numerous virtual agents are created, each with distinct dynamic parameters, to explore different trajectories in the known environment. Each agent is steered by the same forces as the real robot, but with its individual parameter set. Simultaneously, the motions of known obstacles in the environment are simulated using a prediction model of choice. In our implementation, we make use of a constant velocity model, in which the velocity was inferred from the last known positions. More sophisticated models would enhance the prediction and can also be adjusted to the available sensor data if required.

We assess the performance of each agent using a predefined step time, which can be adapted to a particular scenario, and select the most favorable agent based on a specific cost function (cf. Sec. IV-H). The parameter set $\tilde{\mathfrak{P}}_b$ of this best agent is used to calculate the immediate next desired position for the real robot to allow reactive responses to dynamic unforeseen environment changes. Note that this control architecture allows different sampling times for the control command calculation T_c and environment exploration of the predictive agents $T_p = n_p T_c$, with $n_p \in \mathbb{Z}^+$. Thus, reactive motion planning remains unaffected by (potentially long) global environment explorations. In this study, we focus on the artificial current vector because it defines the direction in which an obstacle is avoided.

In contrast to [10], where we used a brute-force approach to find the best parameter sets, herein, we define several heuristics that are executed and evaluated in parallel. This significantly reduces the number of necessary agents and therefore enhances the computation performance, especially in more complex environments, which is crucial for controlling complex systems with high degrees of freedom. The number of maximum agents n_a can be set beforehand and should reflect the computing power of the hardware, particularly the number of available hyperthreads. The procedure of the multi-agent framework is shown in the upper part of Fig. 4.

G. Artificial Current Vector Heuristics

In the following, we introduce five different heuristics for calculating the artificial current vector and thus define the direction of the robot's avoidance maneuver. Further heuristics can be added depending on the desired behavior or task. We also introduce an approach for utilizing randomized current vectors if more hyper-threads are available on the computing hardware. The calculation of the robot avoidance direction for each heuristic is schematically illustrated in Fig. 3. The figure presents all the vectors necessary for generating the artificial current vectors, which will be explained comprehensively in the following subsections. Note that the artificial current for



(a) (b)

Fig. 2: Exemplary comparison of two heuristics in similar environments. In (a), the *Goal Vector Heuristic* (red) results in a shorter and more easily traversable path than the *Obstacle Distance Heuristic* (blue). However, a small change of the obstacle placement, where all obstacles are elevated by $0.3\,\mathrm{m}$ (subfigure (b)) results in a collision of the red heuristic due to a violation of the workspace limits and in a better trajectory of the blue heuristic. The remaining planned path of the *Goal Vector Heuristic* in (b) is depicted with a dashed orange line.

calculating the CF forces needs to be perpendicular to the obstacle surface normal.

a) Velocity Heuristic: The first heuristic is inspired by [68], where the current robot velocity is projected onto the obstacle surface, resulting in an avoidance direction which is close to the current direction of the robot motion

$$\boldsymbol{c}_{vel} = \dot{\boldsymbol{d}} - \frac{\boldsymbol{d}_{oc}}{\|\boldsymbol{d}_{oc}\|}\left(\frac{\boldsymbol{d}_{oc}}{\|\boldsymbol{d}_{oc}\|}\cdot\dot{\boldsymbol{d}}\right). \tag{23}$$

Here, again, $\dot{\boldsymbol{d}}$ is the relative velocity between the robot control point and the obstacle and $\boldsymbol{d}_{oc}$ is the vector pointing from the robot control point to the obstacle geometric center.

b) Path Length Heuristic: The next heuristic aims to minimize the path length for circumventing an obstacle. Towards this end, we use the vector pointing from the robot to the goal $\boldsymbol{d}_g = \boldsymbol{p}_g - \boldsymbol{p}_a$ and project it onto the vector pointing from the robot to the geometric center of the obstacle $\boldsymbol{d}_{oc} = \boldsymbol{p}_{o_{gc}} - \boldsymbol{p}_a$ resulting in the following artificial current vector

$$\boldsymbol{c}_{pl} = \boldsymbol{d}_g - \frac{\boldsymbol{d}_{oc}}{\|\boldsymbol{d}_{oc}\|}\left(\frac{\boldsymbol{d}_{oc}}{\|\boldsymbol{d}_{oc}\|}\cdot\boldsymbol{d}_g\right). \tag{24}$$

Note that this heuristic might still result in globally suboptimal path lengths as the artificial current is calculated for each obstacle individually.

c) Goal Vector Heuristic: This heuristic is taken from our previous work [46] and will lead to similar avoidance maneuvers as the path length heuristics. Differences in the resulting paths will occur in particular when dynamic obstacles are considered, because we calculate a fixed avoidance direction when the robot enters the detection sphere of an obstacle

$$\boldsymbol{c}_{0_{gv}} = \frac{\boldsymbol{d}_g}{\|\boldsymbol{d}_g\|}\left(\frac{\boldsymbol{d}_g}{\|\boldsymbol{d}_g\|}\cdot\boldsymbol{d}_{oc}\right) - \boldsymbol{d}_{oc}. \tag{25}$$

This initial current vector is used for calculating an artificial magnetic field vector

$$\boldsymbol{b}_{gv} = \boldsymbol{c}_{0_{gv}} \times \boldsymbol{d}_{oc}, \tag{26}$$

which is used for the calculation of the real current in each time step

$$\boldsymbol{c}_{gv} = \frac{\boldsymbol{d}_{oc}\times\boldsymbol{b}_{gv}}{\|\boldsymbol{d}_{oc}\|\,\|\boldsymbol{b}_{gv}\|}. \tag{27}$$

Note that we can save computational resources by omitting the subtraction of $\boldsymbol{d}_{oc}$ in Eq. (25), which is rendered redundant due to the cross product in Eq. (26).

d) Obstacle Distance Heuristic: Using this heuristic the robot will navigate around obstacles by always choosing a path that leads further away from next closest obstacle of the active obstacle. To prevent oscillations the avoidance direction is defined when the robot first enters the detection shell of the active obstacle by calculating an initial artificial current vector

$$\boldsymbol{c}_{0_{od}} = \frac{\boldsymbol{d}_{oc}}{\|\boldsymbol{d}_{oc}\|}\left(\frac{\boldsymbol{d}_{oc}}{\|\boldsymbol{d}_{oc}\|}\cdot\boldsymbol{d}_{oo}\right) - \boldsymbol{d}_{oc}, \tag{28}$$

where $\boldsymbol{d}_{oo}$ denotes the vector pointing from the geometric center of the active obstacle to the geometric center of its closest next obstacle. This vector is used to calculate an artificial magnetic field vector

$$\boldsymbol{b}_{od} = \boldsymbol{c}_{0_{od}} \times \boldsymbol{d}_{oc}, \tag{29}$$

which defines the avoidance direction. The artificial current in each time step can then be calculated with

$$\boldsymbol{c}_{od} = \frac{\boldsymbol{d}_{oc}\times\boldsymbol{b}_{od}}{\|\boldsymbol{d}_{oc}\|\,\|\boldsymbol{b}_{od}\|}. \tag{30}$$

In the case of only a single obstacle, the current vector is defined randomly as described in Eq. (34). In the same way as in Eq. (25), we save computational resources by not performing the subtraction of $\boldsymbol{d}_{oc}$ in Eq. (28) in our implementation.

e) Path Length - Obstacle Heuristic: This heuristic will make the robot choose a path between the paths of the previous two heuristics resulting in an avoidance direction, which is a tradeoff between leading away from the other obstacle and minimizing the path length. In the same way as the obstacle distance heuristic, we use an initialization when the robot enters the obstacle's detection shell:

$$\boldsymbol{c}_{0_{pl-od}} = \boldsymbol{c}_{pl} + \boldsymbol{c}_{od}, \tag{31}$$

with $\boldsymbol{c}_{pl}$ from eq. (24) and $\boldsymbol{c}_{od}$ from eq. (30). This is again used to calculate the artificial magnetic field vector

$$\boldsymbol{b}_{pl-od} = \boldsymbol{c}_{0_{pl-od}} \times \boldsymbol{d}_{oc} \tag{32}$$

which allows the calculation of the actual artificial current vector

$$\boldsymbol{c}_{pl-od} = \frac{\boldsymbol{d}_{oc}\times\boldsymbol{b}_{pl-od}}{\|\boldsymbol{d}_{oc}\|\,\|\boldsymbol{b}_{pl-od}\|}. \tag{33}$$

Similarly to the obstacle distance heuristic, if only one obstacle is present, the randomized approach is used for calculating $\boldsymbol{c}_{od}$.

f) Random Artificial Current: As a fallback and in case of free computation resources, we introduce an approach for calculating random artificial current vectors. As mentioned before, the artificial current must be perpendicular to the obstacle surface. Additionally, the avoidance direction should be fixed to prevent oscillations, thus we only use randomized vectors to define the artificial magnetic field vectors for each obstacle, which in turn can be used for calculating the current:

$$\boldsymbol{c}_{ran} = \frac{\boldsymbol{d}_{oc}\times\boldsymbol{b}_{ran}}{\|\boldsymbol{d}_{oc}\|\,\|\boldsymbol{b}_{ran}\|}, \tag{34}$$

where $\boldsymbol{b}_{ran}$ is a random vector.

Note that each of the heuristics is designed for specific configurations of the environment (see Fig. 2). Consequently, the quality of the resulting avoidance behavior of each heuristic highly depends on the placement of the obstacles. We illustrate

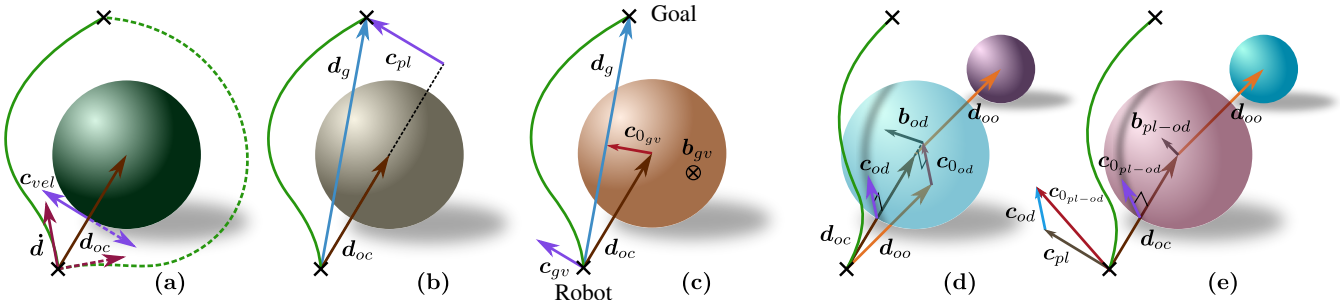


Fig. 3: Schematic illustration depicting the essential vectors involved in calculating the heuristics for determining an avoidance direction. In each figure, the purple vector denotes the artificial current vector, defining the direction for evading obstacles. Note that the resulting trajectory (green) is expected to be comparable among all heuristics in the illustrated scenario featuring only one primary obstacle. Labels (a),(b),(c),(d), and (e) follow the description in sec. IV-G.

these characteristics by means of an exemplary comparison of two heuristics in Fig. 2, where we show that a slight change in the environment has a significant impact on the choice of the heuristics. Moreover, we would like to emphasize that only the combination of the different heuristics results in appropriate trajectories in a multitude of environments. This always leads to better or equally good results than applying the most suitable heuristic individually, as we show later in sec. IX-B.

H. Cost Function for Agent Assessment

In this paper, the cost function $\tilde{\mathfrak{F}}$ is defined with the following criteria for evaluating each predictive agent:

- The path length of the agent trajectory $c_{pl} = w_{pl} \sum_{i=t+1}^{t+T_p} \|\boldsymbol{p}_a(i) - \boldsymbol{p}_a(i-1)\|$;
- The remaining distance to the goal position $c_{gd} = w_{gd} \left\|\boldsymbol{p}_g - \boldsymbol{p}_a(t+T_p)\right\|$;
- The minimal distance of the agent trajectory to all obstacles

$$d_{\min} = \min_{i\in[t+1,t+T_p],j\in[1,n_o],\boldsymbol{p}_{o_j}(i)\in\mathbb{O}_j} \left\|\boldsymbol{p}_{o_j}(i) - \boldsymbol{p}_a(i)\right\|, \quad (35)$$

$$c_{od} = \frac{w_{od}}{d_{\min}} \quad (36)$$

  where $\boldsymbol{p}_{o_j}(i)$ is the closest point of obstacle j to the robot control point at the time step $t = i$;
- The deviation of the trajectory from predefined workspace limits p_k^+ and p_k^-

$$c_{ws} = w_{ws} \sum_{i=t+1}^{t+T_p} \begin{cases} \left(p_{a,k}(i) - p_k^+\right)^2 & \text{if } p_{a,k}(i) > p_k^+, \\ \left(p_{a,k}(i) - p_k^-\right)^2 & \text{if } p_{a,k}(i) < p_k^-, \\ 0 & \text{otherwise}, \end{cases}$$

  for all $k \in [1, 2, 3]$.

Here, $w_{pl}, w_{gd}, w_{od}, w_{ws} > 0$ are the respective weighting factors and the total cost of an agent trajectory is defined as

$$\tilde{\mathfrak{F}} = c_{pl} + c_{gd} + c_{od} + c_{ws}. \quad (37)$$

Depending on the desired robot behavior and task, other criteria can be also added to extend the definition.

V. The Cooperative Dual Task Space (CDTS)

This section elucidates fundamental concepts regarding the cooperative task space and the various controllable cooperative primitives. The CDTS framework exploits $\mathrm{Spin}(3) \ltimes \mathbb{R}^3$ transformations to describe the cooperative space shared by both arms. To start with, let us consider a two-arm system, as the one shown in Fig. 1. This shared cooperative space can be described by the variables $\underline{\boldsymbol{x}}_r$ and $\underline{\boldsymbol{x}}_a$ that respectively define the relative pose, between left and right end-effectors, and the absolute pose of a frame located between end-effectors w.r.t. to a common coordinate system (see Fig 4). Without loss of generality, the absolute pose can be shifted by means of a constant transformation.

Definition 1. *The CDTS variables, that is, the relative and absolute poses, are defined as*

$$\underline{\boldsymbol{x}}_r = \underline{\boldsymbol{x}}_2^* \underline{\boldsymbol{x}}_1, \quad (38)$$

$$\underline{\boldsymbol{x}}_a = \underline{\boldsymbol{x}}_2 \underline{\boldsymbol{x}}_{r/2}. \quad (39)$$

where $\underline{\boldsymbol{x}}_1$ and $\underline{\boldsymbol{x}}_2$ represent the end-effector poses of the two arms, and $\underline{\boldsymbol{x}}_{r/2}$ is the transformation that corresponds to half of the angle ϕ_r around the axis $\boldsymbol{n}_r = \hat{i}n_x + \hat{j}n_y + \hat{k}n_z$ of the quaternion $\mathcal{P}(\underline{x}_r)$ and half of the translation between the two arms [63].

The goal from here on is to define task-specific geometric primitives and corresponding Jacobians for controlling them in the sense that these cooperative Jacobians map robot joint velocities to the CDTS variables and vice-versa. Note that, in this case, the joint vector $\boldsymbol{q}_{\mathrm{cp}} \in \mathbb{R}^n$ represents the entire system i.e., it is the stacked vector of the joint variables of the individual arms that produce the dual-arm system and is given by, $\boldsymbol{q}_{\mathrm{cp}} = [\boldsymbol{q}_1^T \;\; \boldsymbol{q}_2^T]^T$. Analytically, the primitive $\boldsymbol{u}$ and joint $\boldsymbol{q}_{\mathrm{cp}}$ are related by

$$\dot{\boldsymbol{u}} = \boldsymbol{J}_{\mathrm{task}} \dot{\boldsymbol{q}}_{\mathrm{cp}}, \quad (40)$$

where $\boldsymbol{J}_{\mathrm{task}}$ is the Jacobian capturing the first-order task dynamics. The following subsections detail the various primitives (see Table II for the list) that we consider for effective bimanual manipulation for our use case.

A. Relative Dual Pose Control:

The joint vector $\boldsymbol{q}_{\mathrm{cp}}$ is related to the task cooperative variable $\underline{\boldsymbol{x}}_r$ by the relative dual quaternion Jacobian $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$ as conveyed by

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_r} = \left[\overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_2^*)\boldsymbol{J}_{\underline{\boldsymbol{x}}_1} \;\; \overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_1)\boldsymbol{J}_{\underline{\boldsymbol{x}}_2}^*\right], \quad (41)$$

where $\boldsymbol{J}_{\underline{\boldsymbol{x}}_i} = \partial f_i / \partial \boldsymbol{q}_i$ is the analytical Jacobian, which can be easily derived using dual quaternion algebra as in [70]–[72]. The matrix $\overset{+}{\boldsymbol{H}}$ and $\overset{-}{\boldsymbol{H}}$ are Hamilton operators that can be used to commute terms when performing dual quaternions multiplications.[4] Considering the mapping of the

[4]Similar to $SE(3)$, unit dual quaternion multiplication is not commutative.

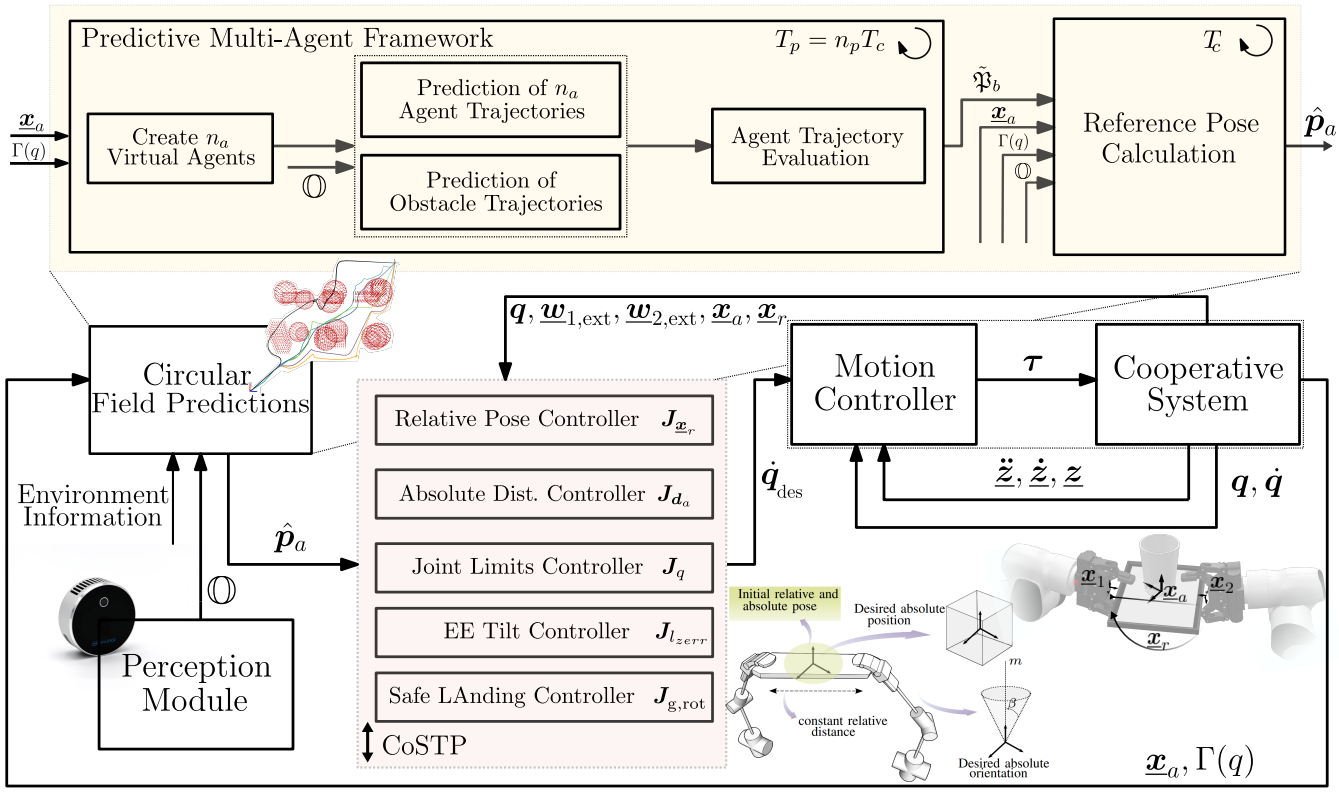


Fig. 4: Overview of our planning and control architecture: The main focus in this work are the highlighted (i) Predictive Multi-Agent Framework (PMAF) block which spawns different virtual agents and evaluates them, finally leading to the agent with the best parameter set $\tilde{\mathfrak{P}}_b$ which is then used to compute the reference pose $\hat{\boldsymbol{p}}_a$, the (ii) Cooperative Set-based Task Priority (CoSTP) block with switching equality and set-based constraints, and the information exchange among both. The perception module for obstacle information and the admittance-based inner motion controller along with the dual-arm set-up complete the system. The CoSTP controller, along with the lower level motion controller, runs at 1KHz dispensing effective torque commands for the dual-arm system.

dual quaternion set $\underline{\boldsymbol{S}}$ into $\mathbb{R}^8$, that is, $\text{vec} : \underline{\boldsymbol{S}} \to \mathbb{R}^8$ and the dual quaternion $\underline{\boldsymbol{z}}=\underline{\boldsymbol{x}}\underline{\boldsymbol{y}}$, the Hamilton operators, $\overset{+}{\boldsymbol{H}}$ and $\overset{-}{\boldsymbol{H}}$, satisfy $\text{vec}\,\underline{\boldsymbol{z}}=\overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}})\,\text{vec}\,\underline{\boldsymbol{y}}=\overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{y}})\,\text{vec}\,\underline{\boldsymbol{x}}$, [70]–[72].

B. Absolute Dual Pose Control:

In a similar fashion, the absolute dual quaternion Jacobian can be expressed as (refer to appendix for definitions)

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_a} = \left[\overset{-}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_{r/2})\boldsymbol{J}_{\underline{\boldsymbol{x}}_{2ext}} + \overset{+}{\boldsymbol{H}}(\underline{\boldsymbol{x}}_2)\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}}\right], \tag{42}$$

where $\boldsymbol{J}_{\underline{\boldsymbol{x}}_{2ext}} = [\boldsymbol{0}\ \boldsymbol{J}_{\underline{\boldsymbol{x}}_2}]$, and $\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}}$ is given by

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}} = \begin{bmatrix} \frac{1}{2}\overset{-}{\boldsymbol{H}}_4(\boldsymbol{r}^*_{r/2})\boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_r)} \\ \frac{1}{4}\left(\overset{-}{\boldsymbol{H}}_4(\boldsymbol{r}_{r/2})\boldsymbol{J}_{\boldsymbol{p}_r} + \overset{+}{\boldsymbol{H}}_4(\boldsymbol{p}_r)\boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_{r/2})}\right) \end{bmatrix}$$

in which $\boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_{r/2})}$ refers to the first four rows of the relative dual quaternion Jacobian $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$.

As impressed upon in earlier instances, we now exemplify the trivial extraction of the cooperative geometric task primitives. These stem from the relative and absolute variables defined in (38)-(39) and the corresponding Jacobians (41)-(42) which then makes the way for the (local) task-space to joint-space mapping of these task primitives. Following the demonstration in [8], the *relative translation* between arms and the *absolute translation* in coordinate-frame is defined by $\boldsymbol{p}_\chi = 2\,\mathcal{D}\left(\underline{\boldsymbol{x}}_\chi\right)\mathcal{P}\left(\underline{\boldsymbol{x}}_\chi\right)^*$, where $\chi = \{r, a\}$ refers to the relative or absolute variables.

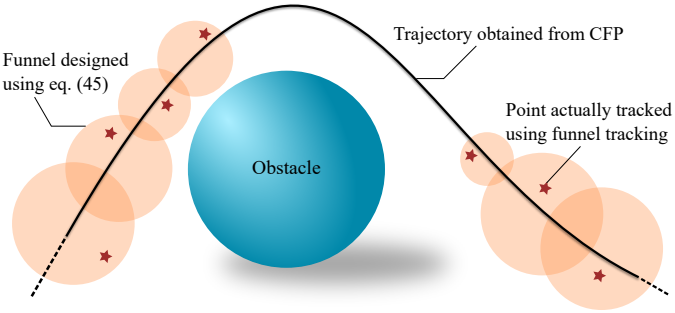


Fig. 5: Funnel tracking which tracks not the 3 DoF CFP trajectory through the absolute position of the cooperative system but only a scalar distance. For brevity, funnels are only shown at the beginning and end of the trajectory.

The translation Jacobians are therefore

$$\boldsymbol{J}_{\boldsymbol{p}_\chi} = 2\overset{-}{\boldsymbol{H}}_4\left(\mathcal{P}\left(\underline{\boldsymbol{x}}^*_\chi\right)\right)\boldsymbol{J}_{\mathcal{D}(\underline{\boldsymbol{x}}_\chi)} + 2\overset{+}{\boldsymbol{H}}_4\left(\mathcal{D}\left(\underline{\boldsymbol{x}}_\chi\right)\right)\boldsymbol{J}^*_{\mathcal{P}(\underline{\boldsymbol{x}}_\chi)}. \tag{43}$$

with $J_{\mathcal{P}(\underline{\boldsymbol{x}}_\chi)}$ and $J_{\mathcal{D}(\underline{\boldsymbol{x}}_\chi)}$ being the four upper and lower rows of $J_{\underline{\boldsymbol{x}}_\chi}$, respectively.

The geometric *relative distance* and *absolute distance* define the square distance, that is, the radius of the ball given by $c \triangleq \left\|\boldsymbol{p}_\chi\right\|^2$. The corresponding Jacobian is given by

$$\boldsymbol{J}_{d_\chi} = 2\left(\text{vec}_4\,\boldsymbol{p}_\chi\right)^T\boldsymbol{J}_{\boldsymbol{p}_\chi}. \tag{44}$$

The *relative* and *absolute orientation* are given by the unit quaternion $\boldsymbol{r}_\chi = \mathcal{P}\left(\underline{\boldsymbol{x}}_\chi\right)$. The corresponding mapping can therefore be trivially derived through $\boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_\chi)}$.

C. Reactive Funnel Tracking:

As discussed before, the absolute position primitive requires the definition of 3 DoF. Herein, we bring in the concept of *funnel trajectory* which is generated from the force-field output together with the information of the closest obstacle from the n_o present, given by eq. (35). In this context, the desired motion coordination satisfying the reactive approach leads to a collision free trajectory and is given by $\mathfrak{R}_d(\mathfrak{R}_a)$ with $\overline{d} \leftarrow d_{\min}$ defined from (35) and distance given by

$$d_{\text{CF}} = \|\hat{\boldsymbol{p}}_a - \boldsymbol{p}_a\|, \tag{45}$$

where $\hat{\boldsymbol{p}}_a$ is the desired CF trajectory at a given instant and the $\boldsymbol{p}_a$ the current absolute position of the cooperative system. As illustrated in Fig. 5, this leads to an improved task defined within 1 DoF instead of the usual 3 required from a naive integration. More specific to the planning context, we define the desired trajectory from the CF output to be within a desired distance, instead of specific Cartesian position. Thus, the control primitive is $d \triangleq \|\boldsymbol{p}_\chi\|^2$.

D. Cooperative Tilt Control:

As depicted in Fig. 1 we are also interested in controlling the tilt of the coordinated object (e.g., the tray with the glass for instance) and therefore it is necessary to bring to light the Jacobian mapping from $\dot{\boldsymbol{q}}$ w.r.t. the line passing through the z-axis of the absolute/relative reference frame (see Fig. 6). This changes according to the robot's (held object's) motion and mathematically, that line $\underline{\boldsymbol{l}}_z$ can be encoded as,

$$\underline{\boldsymbol{l}}_z = \boldsymbol{l}_z + \varepsilon \boldsymbol{m}_z, \tag{46}$$

where $\boldsymbol{l}_z = \boldsymbol{r}\hat{k}\boldsymbol{r}^*$ is the frame transformation of the z-axis and $\boldsymbol{m}_z$ is the moment of the line. The end-effector constraint Jacobian can then be defined as,

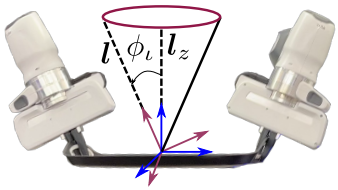


Fig. 6: Cooperative inclination control of the tray using Plücker lines defined on the CDTS frame(s).

$$\boldsymbol{J}_{r_z} = \left[\bar{\boldsymbol{H}}_4(\hat{k}\boldsymbol{r}^*)\boldsymbol{J}_r + \overset{+}{\boldsymbol{H}}_4(\boldsymbol{r}\hat{k})\boldsymbol{C}_4\boldsymbol{J}_r\right], \tag{47}$$

in which $\boldsymbol{J}_{\boldsymbol{r}_\chi} = \boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_\chi)}$ is the rotational Jacobian and $\boldsymbol{C}_4 = \text{diag}(1,-1,-1,-1)$. Therefore, in order to avoid undesired end-effector configurations we want to control the angles between the static Plücker line and $\underline{\boldsymbol{l}}_z$. This distance can be represented by [73], $f = (\boldsymbol{l} - \boldsymbol{l}_z)^T(\boldsymbol{l} - \boldsymbol{l}_z)$. The end effector constraint Jacobian error definition can then be,

$$\boldsymbol{J}_{l_{zerr}} = -2\text{vec}_4(\boldsymbol{l} - \boldsymbol{l}_z)^T \boldsymbol{J}_{r_z}. \tag{48}$$

This constructs the primitive definition as $\tilde{\mathbf{l}} = (\boldsymbol{l} - \boldsymbol{l}_z)^T(\boldsymbol{l} - \boldsymbol{l}_z)$.

E. Joint Limits Control:

To ensure that the computed joint velocities do not push the system beyond the limits, we enforce joint limit control through the Jacobian $\boldsymbol{J}_q$. We define the middle of the joint bounds using $\boldsymbol{q}_{\text{mid}} = \frac{(\underline{q_i} + \overline{q_i})}{2}$, where $\underline{q_i}$ and $\overline{q_i}$ represent the lower and upper limits of joint i, respectively. In our quest to drive the joints to their center, we define $s = 0.5\sum(q_i - q_{i,\text{mid}})^2$. $\boldsymbol{J}_q$ is then generated by the derivative $\frac{\delta s}{\delta q}$.

TABLE II: MAIN GEOMETRIC TASKS PRIMITIVES AND TASK JACOBIANS.

Task Primitive $\boldsymbol{u}$	Contr.Sets	DOF	Task
Rel/Abs distance ($d \in \mathfrak{R}_d$)	$d \preceq \overline{d}$	1	$\boldsymbol{J}_{d_\chi}$ in (44)
Rel/Abs position ($\boldsymbol{p} \in \mathfrak{R}_p$)	$\underline{\boldsymbol{p}} \preceq \boldsymbol{p} \preceq \overline{\boldsymbol{p}}$	3	$\boldsymbol{J}_{\boldsymbol{p}_\chi}$ in (43)
Rel/Abs orientation ($\boldsymbol{r} \in \mathfrak{R}_o$)	$\underline{\boldsymbol{r}_\ell} \preceq \log \boldsymbol{r} \preceq \overline{\boldsymbol{r}_\ell}$	3	$\boldsymbol{J}_{\boldsymbol{r}_\chi} = \boldsymbol{J}_{\mathcal{P}(\underline{\boldsymbol{x}}_\chi)}$
Rel/Abs tilt ($\phi_\iota \in \mathfrak{R}_{\phi_\iota}$)	$\underline{\phi_\iota} \preceq \phi_\iota \preceq \overline{\phi_\iota}$	1	$\boldsymbol{J}_{\phi_\chi}$ in (48)
Rel/Abs Sing. ($\sigma_{min} \in \mathbb{R}$)	$\underline{\sigma} \preceq \sigma_{min}$	1	$\boldsymbol{J}_\sigma = \frac{\partial \sigma_{\min}(\chi)}{\partial \boldsymbol{q}}$
Joint limits ($q_i \in \mathbb{R}^n$)	$\underline{q_i} \preceq q_i \preceq \overline{q_i}$	1	$\boldsymbol{J}_q = q_i - \frac{(\underline{q_i} + \overline{q_i})}{2}$

VI. COOPERATIVE SET-BASED TASK PRIORITY (COSTP)

This section describes the switching strategy employed for flexible control of the geometric task primitives introduced in the previous section. First, we mathematically define controllable sets originating from geometric primitives. Second, we incorporate convergence guarantees for these hybrid sets in the form of task priorities, similar to the work in [74], combined with hysteresis-based switching of the controlled cooperative tasks [61]. These enable robust prioritized handling of reference trajectory tracking generated by the vector fields as well as the other defined CDTS primitives.

In an ideal scenario, during task execution, we do not want the system to depart from the geometric constraint subsets $\mathfrak{R}$, that govern the complex cooperative manipulation task. In other words, the objective is to ensure that the tasks remain restricted in pre-defined sets such that successful achievement of several tasks is possible. This is similar to the concept of area of satisfaction from [75], where the motion is defined over desired feasible intervals along the controlled primitives. Keeping that in mind, we now elucidate the several primitives in connection with the Cooperative Dual Task Space.

Definition 2. *Given a set $\mathfrak{S} \subseteq Spin(3) \ltimes \mathbb{R}^3$ in the cooperative space and a dual quaternion pose $\underline{\boldsymbol{x}}_g$, we delineate the following proper geometrical subsets with regards to $\mathfrak{S}$,*

$$\mathfrak{R}_d(\mathfrak{S}) = \left\{\underline{\boldsymbol{x}}_g \in \mathfrak{S} \mid d = \left\|\mathfrak{T}\left(\underline{\boldsymbol{x}}_g\right)\right\|, \ d \in \mathbb{R}\right\},$$
$$\mathfrak{R}_p(\mathfrak{S}) = \left\{\underline{\boldsymbol{x}}_g \in \mathfrak{S} \mid \boldsymbol{p} = \mathfrak{T}\left(\underline{\boldsymbol{x}}_g\right), \ \boldsymbol{p} \in \mathbb{H}_0\right\},$$
$$\mathfrak{R}_o(\mathfrak{S}) = \left\{\underline{\boldsymbol{x}}_g \in \mathfrak{S} \mid \boldsymbol{r} = \mathcal{P}\left(\underline{\boldsymbol{x}}_g\right), \ \boldsymbol{r} \in Spin(3)\right\},$$
$$\mathfrak{R}_{\phi_\iota}(\mathfrak{S}) = \left\{\underline{\boldsymbol{x}}_g \in \mathfrak{S} \mid \phi_\iota = \cos^{-1}(\langle \boldsymbol{l}_z, \boldsymbol{l}\rangle), \boldsymbol{r} = \mathcal{P}(\underline{\boldsymbol{x}}_g), \boldsymbol{l}_z = \boldsymbol{r}\hat{k}\boldsymbol{r}^*\right\},$$

where $\mathfrak{T}\left(\underline{\boldsymbol{x}}_g\right) \triangleq 2\mathcal{D}\left(\underline{\boldsymbol{x}}_g\right)\mathcal{P}\left(\underline{\boldsymbol{x}}_g\right)^*$ is the translation corresponding to $\underline{\boldsymbol{x}}_g$ and $\mathbb{H}_0$ is the set of all pure quaternions, isomorphic to $\mathbb{R}^3$. The subscripts d, p, o, and ϕ_ι denote distance, translation, rotation, and tilt angle respectively. More in detail, ϕ_ι describes the opening angle of a solid cone defined by the rotation of the body $z-$axis (Plücker line) to the absolute coordinate frame, i.e., $\boldsymbol{l}_z$ around a desired line $\boldsymbol{l}$. (Fig. 6).

Remark 1. *These definitions of basic sets allow us to introduce the various controllable subsets as shown in Table II.*[5] *As far as cooperative manipulation tasks are concerned, our focus in this paper is limited to these introduced subsets which are utilized for flexible control of the dual-arm system. For further details regarding cooperative primitives, we refer readers to [63] and also to [70], [72]. For a general treatment of geometric feature extraction from dual quaternions, we allude to [61], [62], and [73].*

[5]Joint limits is an exceptional case where joint and task-space coincide.

Now that the constraint subsets $\mathfrak{R}_\zeta$ (ζ can be d, p, o, and ϕ_ι) are defined respecting the controllable primitives, we can demarcate the domains corresponding to different subsets. For instance, the absolute/relative position $\mathfrak{R}_p$ which has 3 degrees of freedom and resides in $\underline{\boldsymbol{p}} \preceq \boldsymbol{p} \preceq \overline{\boldsymbol{p}}$. Comparably, the relative/absolute distance $\mathfrak{R}_d$ controls the scalar distance $d_r \in \mathfrak{R}_d$ that enables reactive funnel tracking. That is, in this case, the feasibility of the task requires that primitives operate within their valid domain D defined as $\tilde{\boldsymbol{d}}_{\min} \preceq \tilde{\boldsymbol{d}} \preceq \tilde{\boldsymbol{d}}_{\max}$, where $\tilde{\boldsymbol{d}}_{\min}$ and $\tilde{\boldsymbol{d}}_{\max}$ are the boundaries of the feasible set. It is important to note that each geometric primitive requires different degrees of freedom from the CDTS, which can be further explored to enhance flexibility and in complex cooperative planning scenarios. The primitives that we consider along with their domain sets and degrees of freedom are reported in Table II.[6] Here it can be seen that according to the definition, the task is still feasible and valid at the boundary domain, however control intervention is needed once it departs the defined sets. We address this by employing the multi-priority switching strategy as described in the next subsection.

A. Switching Strategy for Multi-Priority Control

For each of the defined cooperative tasks, the goal is to design a control scheme such that set-based tasks are satisfied. As encoded in the first order differential equation (40), the error $\boldsymbol{e}$ towards a desired equilibrium can be defined as $\boldsymbol{e} = \boldsymbol{u} - \hat{\boldsymbol{u}}$ where $\hat{\boldsymbol{u}}$ denotes the current state of the system. The resulting acceleration is given by the expression $\ddot{\boldsymbol{e}} = \dot{\boldsymbol{J}}_u \dot{\boldsymbol{q}} + \boldsymbol{J}_u \ddot{\boldsymbol{q}}$. For the purpose of accomodating all other tasks in this formulation, we explore the redundant space accessible within the CDTS framework. More specifically, we can define priorities by exploiting the nullspace of the main task $\boldsymbol{u}$. This is captured mathematically in the equation

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_u^+ \dot{\hat{\boldsymbol{e}}} + \boldsymbol{P}_u \dot{\boldsymbol{q}}_{\text{null}}, \tag{49}$$

where $\boldsymbol{q}_{\text{null}}$ is the lower priority task that is executed on the condition that the nullspace has the freedom, $\boldsymbol{P}_u = (\boldsymbol{I} - \boldsymbol{J}_u^+ \boldsymbol{J}_u)$ is the linear operator that projects $\dot{\boldsymbol{q}}_{\text{null}} \in \mathbb{R}^n$ in the nullspace of $\boldsymbol{J}_u$, and $\dot{\hat{\boldsymbol{e}}}$ is the desired error convergence [76]. It is a known fact that the evolution of the stack of the active tasks (set-based) is affected by the higher priority main task [77].

The arbitrary joint velocity $\dot{\boldsymbol{q}}_{\text{null}}$ can be explored for multiple tasks following a priority order—since it acts in the nullspace of the task $\boldsymbol{u}$.[7] Hence, given an order of the cooperative primitives defined in Section V, which we can rename as $\boldsymbol{u}_1 \cdots \boldsymbol{u}_\eta$ for expositional reasons, the resulting joint velocities should be

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{u_1}^+ \kappa_1 \dot{\hat{\boldsymbol{e}}}_1 + \boldsymbol{P}_{u_1} \kappa_2 \dot{\hat{\boldsymbol{e}}}_2 + \boldsymbol{P}_{u_{1:2}} \kappa_3 \dot{\hat{\boldsymbol{e}}}_3 + \cdots + \boldsymbol{P}_{u_{1:\eta-1}} \kappa_\eta \dot{\hat{\boldsymbol{e}}}_\eta,$$

where κ_i is the positive gain which is defined accordingly to [77] such to ensure asymptotic convergence of the resulting tasks—if they are defined independently, i.e., if the one task does not disturb a higher priority one (in which case this should not be satisfied). The matrix $P_{u_{1:i}}$ indicates the nullspace projector for an augmented Jacobian $\boldsymbol{J} = [J_{u_1} \ \cdots \ J_{u_i}]$.

In practice, however, a closer look at the task definition and the cooperative primitives in Table II, reveals that most tasks in CDTS can be satisfied even if the exact configuration is not met, i.e., it does not converge to an equilibrium but rather remains stable/converge to a set around it. In this context, the main idea is to explore bounded geometric regions from Table II. The stability and convergence follows from [74], [77], yet we introduce a linear blending based strategy that avoids chattering and ensures continuity in the joint space. The switching is performed based on the activation/deactivation of tasks in accordance with the set-based condition and the time evolution of each individual task in the current mode, i.e.,

$$\dot{\boldsymbol{e}}_i = \boldsymbol{J}_{ui} \dot{\boldsymbol{q}} = \boldsymbol{J}_u \left(\boldsymbol{J}_{u_1}^+ \kappa_1 \dot{\hat{\boldsymbol{e}}}_1 + \cdots + \boldsymbol{P}_{u_{1:\eta-1}} \kappa_\eta \dot{\hat{\boldsymbol{e}}}_\eta \right).$$

Hence, a set-based task is activated if it lies outside the set region defined in Table II and if its evolution pushes it away from the boundaries. In contrast, if any of the conditions

$$\begin{cases} (i) & \boldsymbol{e}_i \in \left(\underline{\boldsymbol{e}_i}, \overline{\boldsymbol{e}_i} \right), \\ (ii) & \boldsymbol{e}_i \succeq \overline{\boldsymbol{e}_i} \text{ and } \dot{\boldsymbol{e}}_i \preceq 0, \\ (iii) & \boldsymbol{e}_i \preceq \overline{\boldsymbol{e}_i} \text{ and } \dot{\boldsymbol{e}}_i \succeq 0, \end{cases}$$

is satisfied, then the task will be deactivated—that is, it will only act in the lower priority of the self-motion. The activation and deactivation of tasks, with all being defined in the lower-priority level yields the desired switching strategy. That is, if for instance, a task with priority 2 is deactivated, this will be pushed below all activated tasks.

The asymptotic stability of active tasks follows from the fact that all possible N modes are asymptotically stable. This has been proved in [74] with specific control gains, which we use in the paper. Hence, by building a continuous function over the dynamics of each mode, i.e., $f \triangleq \lambda_1 f_1 + \lambda_2 f_2 \cdots + \lambda_N f_N$, where $\sum_i \lambda_i = 1$ for $\lambda_i > 0$, we can build a quadratic Lyapunov function for the augmented vector which guarantees global asymptotic stability. As active tasks reach equilibrium or are within their desired sets, they will be deactivated, which in turn opens space for the subsequential tasks to reach equilibrium. Further details of the augmented dynamics for any task and the resulting Lyapunov-Krasovskii analysis are shown in [74].

Illustrative Example: Here, we present a didactic overview of our switching strategy by analyzing the scenario illustrated in Fig. 1. The first main task, defined as an equilibrium, is the control of the tightly coupled fixed relative orientation between

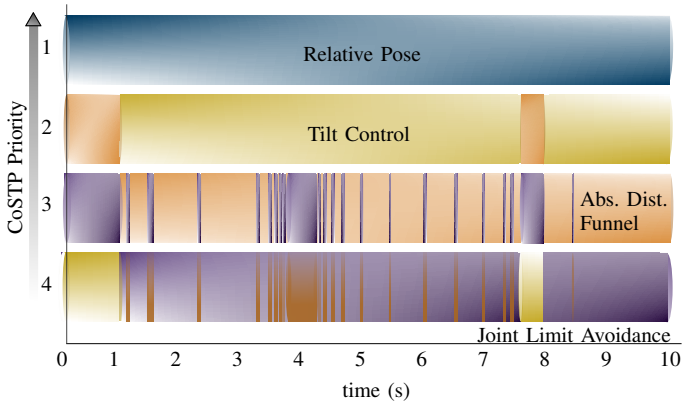


Fig. 7: Switching between the different priority tasks during the evolution of the dual-arm use-case on the KoBo setup. Blue denotes relative pose, orange is absolute distance, yellow depicts tilt control, and purple represents joint-limit avoidance. As expected, the relative pose between the two arms is set as the default first task (top priority) which cannot be disturbed at any given instance.

[6] Notice that the min SVD $\sigma_{\min}$ refers to both relative and absolute tasks which define the CDTS, and not to individual manipulators.

[7] Herein, we define $\dot{\boldsymbol{q}}_{\text{null}}$ based on [69] in order to be free of algorithmic singularities.

the two end-effectors, which is encoded by $\dot{\underline{\boldsymbol{x}}}_r = \boldsymbol{J}_{\underline{\boldsymbol{x}}_r}\dot{\boldsymbol{q}} = 0$. Any motion or task should not disturb this main cooperative task. Our integrated (multi-agent based) feedback planner generates the reference trajectory, which is then followed through the absolute pose funnel (45). During tracking, we also want the system not to tilt the tray beyond a certain safety threshold, which is regulated by the tilt control task. This initially operates in the nullspace of the tracking, however if it is closer to its limit then tracking becomes lower priority and operates in the nullspace of the tilt-control (see Fig. 7).

It is important to note that the above-introduced tasks require only 8 DoFs for successful execution (in contrast to the trivial 12 from a complete modeling). Subsequently, tasks such as joint limit avoidance and local manipulability optimization are accomplished in the augmented nullspace of the above mentioned three main tasks. In this regard, deactivating specific tasks and pushing lower tasks up in the order as the satisfaction conditions are violated and re-activating them back is a critical feature of our framework and is essential for real-time planning success.

Continuity at the Joint Level: Our linear blending strategy allows a continuous joint-space velocity profile during switching. This improves previous results, e.g., [74], which could lead to undesirable residual torques and degrade robot performance. When a task is pushed inside the corresponding controllable sets, its deactivation is started. Instead of using a harsh switch, we introduce linear blending, which outputs a convex combination between tasks gaining a higher priority $\boldsymbol{e}_{\uparrow}$ and those losing priority $\boldsymbol{e}_{\downarrow}$, that is, $\boldsymbol{\Lambda} e_{\uparrow} + (1-\boldsymbol{\Lambda}) e_{\downarrow}$, where $\boldsymbol{\Lambda}$ is a state-dependent variable that evolves within the region. This ensures a smooth and continuous transition of tasks in the joint-space which may otherwise result in jerky profiles.

VII. Contact Handling and Landing Controller

In this section, we illustrate the bimanual contact detection scheme as well as our landing controller which helps to provide a safe and human-like placement of objects with a sense of touch in non-static environments.

A. Bimanual Collision Detection and Wrench Estimation

For successful completion of the landing task, the first challenge that needs to be addressed is to detect the contact. Usually, classical approaches to detect contacts based on external torques and momentum observers will fail to provide robust detection in bimanual tasks as the internal strain of the system even with low errors in the relative control will easily exceed the torque thresholds. We therefore developed the following approach for bimanual contact detection utilizing the estimation of the external wrenches at the end-effectors of both arms. The contact wrenches $\boldsymbol{\mathcal{F}}_{\text{ext,l}} = \begin{pmatrix}\boldsymbol{f}_{\text{ext,l}}^{\text{T}} & \boldsymbol{m}_{\text{ext,l}}^{\text{T}}\end{pmatrix}^{\text{T}}$ and $\boldsymbol{\mathcal{F}}_{\text{ext,r}} = \begin{pmatrix}\boldsymbol{f}_{\text{ext,r}}^{\text{T}} & \boldsymbol{m}_{\text{ext,r}}^{\text{T}}\end{pmatrix}^{\text{T}}$ for left and the right arm, respectively, are calculated internally in the robot controller based on the external torque estimation. We use them to calculate the estimated external wrench $\boldsymbol{\mathcal{F}}_{\text{ext}} = \begin{pmatrix}\boldsymbol{f}_{\text{ext}}^{\text{T}} & \boldsymbol{m}_{\text{ext}}^{\text{T}}\end{pmatrix}^{\text{T}}$ on the intermediate frame between the two arms (absolute pose) as

$$\boldsymbol{\mathcal{F}}_{\text{ext}} := \begin{pmatrix} \boldsymbol{f}_{\text{ext,l}} + \boldsymbol{f}_{\text{ext,r}} \\ \boldsymbol{m}_{\text{ext,l}} + \boldsymbol{m}_{\text{ext,r}} - \frac{1}{2}\boldsymbol{v}_{\text{lr}} \times \boldsymbol{f}_{\text{ext,l}} + \frac{1}{2}\boldsymbol{v}_{\text{lr}} \times \boldsymbol{f}_{\text{ext,r}} \end{pmatrix}, \quad (50)$$

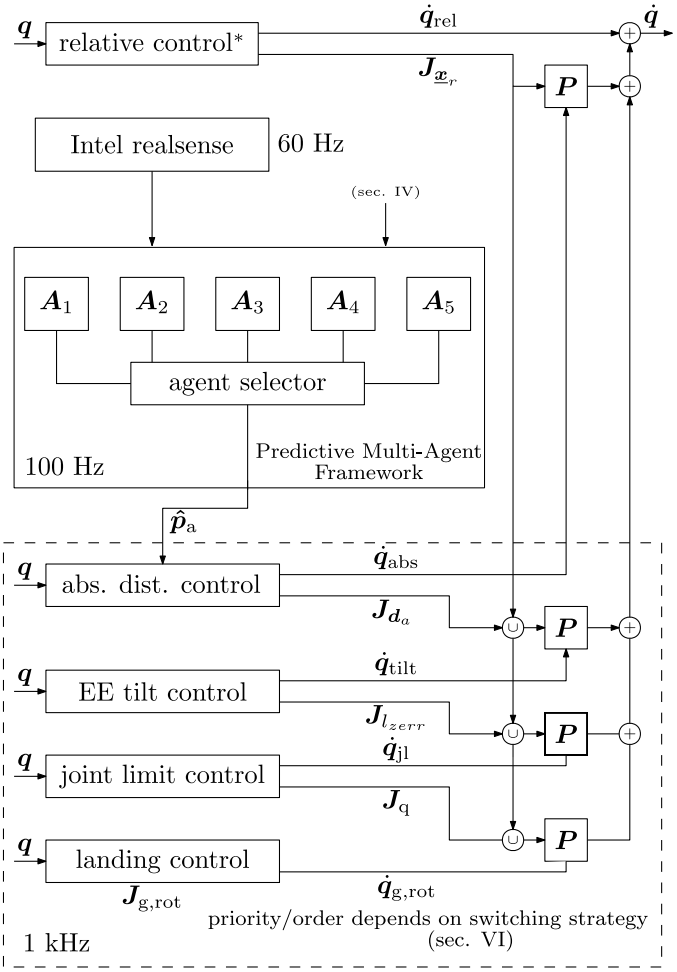


Fig. 8: Overview of our novel integrated planning and control framework. The symbol $\cup$ denotes the concatenation of the matrix inputs into the block for creation of the augmented jacobians. The $\boldsymbol{P}$ block depicts the nullspace projection of the input matrix $\boldsymbol{M}$ applied to the input vector $\boldsymbol{v}$, i.e. the output vector $\boldsymbol{o}$ would be $\boldsymbol{o} = (\boldsymbol{I} - \boldsymbol{M}^{\#}\boldsymbol{M})\boldsymbol{v}$. The # symbol denotes the robust pseudo inverse defined later in the section. For illustrative purpose, this picture only uses 5 agents for environment exploration and a fixed order of task-priorities. The $*$ symbolizes the fact that rel. control is always the first.

where $\boldsymbol{v}_{\text{lr}}$ is the vector from the left to the right arm end-effector. We test $\boldsymbol{\mathcal{F}}_{\text{ext}}$ against a fixed threshold $\boldsymbol{\mathcal{F}}_{\text{ext,th}}$ element-wise for contact detection. The advantage of using $\boldsymbol{\mathcal{F}}_{\text{ext}}$ for contact detection is the elimination of false detections due to internal strain, as any internal strain will show up in both arms wrenches but with a different sign and therefore does not influence $\boldsymbol{\mathcal{F}}_{\text{ext}}$. A dead-zone function is also applied to the elements of $\boldsymbol{f}_{\text{ext}}$ in the experiments so that the system does not react to sensor noise or modelling errors. More specifically, the estimated force $f^*_{\text{ext},i}$ can be given by:

$$f^*_{\text{ext},i} := \begin{cases} f_{\text{ext},i} + f_o, & f_{\text{ext},i} < -f_o \\ 0, & -f_o <= f_{\text{ext},i} <= f_o \\ f_{\text{ext},i} - f_o, & f_{\text{ext},i} > f_o \end{cases} \quad (51)$$

where $i \in \{x, y, z\}$ and f_o is the half size of the dead zone. Furthermore, one can also utilize this information as an input to the planner for integrating sudden contact and reflex reactions.

B. Safe LAnding Controller (SLAC)

When the tray is close to the landing location/surface of roughly known position and orientation, the robotic system

will establish contact with that surface. At this point, using the wrench estimation technique explained above it is straightforward to design a rotational admittance controller to be able to adapt to unknown surfaces. We design and implement the following admittance control law

$$\dot{\boldsymbol{q}}_{\mathrm{rot}} := \boldsymbol{J}^{+}_{\mathrm{g,rot}} \int_0^t \boldsymbol{m}^{*}_{\mathrm{ext}}(\tau) - k_\omega \boldsymbol{\omega}(\tau) d\tau, \tag{52}$$

where $\boldsymbol{J}^{+}_{\mathrm{g,rot}}$, $\boldsymbol{\omega}$ and k_ω denote a robust pseudo inverse of the rotational geometric Jacobian, the rotational velocity of the intermediate frame, and the damping gain of the admittance control, respectively. Similar to the force, the dead zone function from (51) is also applied to $\boldsymbol{m}_{\mathrm{ext}}$. This approach allows the robot to adapt to unknown surface inclinations and establish steady contact before releasing the object.

VIII. Additional Implementation Details

The overall control scheme is depicted in Fig. 8. As an illustrative example, if we assume that the priority order is keeping the relative pose first, and then the absolute position control followed by the absolute tilt and joint limit avoidance, the closed loop control equation is given by

$$\begin{aligned}\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\mathrm{rel}} + \boldsymbol{P}\left(\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}\right)\dot{\boldsymbol{q}}_{\mathrm{abs}} + \boldsymbol{P}\left(\begin{bmatrix}\boldsymbol{J}^T_{\underline{\boldsymbol{x}}_r} & \boldsymbol{J}^T_{\boldsymbol{p}_a}\end{bmatrix}^T\right)\dot{\boldsymbol{q}}_{\mathrm{tilt}} \\ + \boldsymbol{P}\left(\begin{bmatrix}\boldsymbol{J}^T_{\underline{\boldsymbol{x}}_r} & \boldsymbol{J}^T_{\boldsymbol{p}_a} & \boldsymbol{J}^T_{\underline{\boldsymbol{x}}_{l_{\mathrm{zerr}}}}\end{bmatrix}^T\right)\dot{\boldsymbol{q}}_{\mathrm{jl}},\end{aligned} \tag{53a}$$

$$\dot{\boldsymbol{q}}_{\mathrm{rel}} = \boldsymbol{J}^{\#}_{\underline{\boldsymbol{x}}_r}\left(\mathrm{vec}\,\underline{\boldsymbol{x}}_{r,0} - \mathrm{vec}\,\underline{\boldsymbol{x}}_r\right), \quad \dot{\boldsymbol{q}}_{\mathrm{abs}} = \dot{\boldsymbol{q}}_{\mathrm{c}} + \dot{\boldsymbol{q}}_{\mathrm{ff}}, \tag{53b}$$

$$\dot{\boldsymbol{q}}_{\mathrm{tilt}} = \boldsymbol{J}^{\#}_{\underline{\boldsymbol{x}}_{l_{\mathrm{zerr}}}}\left|\mathrm{vec}\left(\boldsymbol{l} - \boldsymbol{l}_z\right)\right|^2, \quad \dot{\boldsymbol{q}}_{\mathrm{jl}} = \frac{1}{4}\boldsymbol{J}^{\#}_{\underline{\boldsymbol{x}}_{q_i}}\left(\boldsymbol{J}^{\mathrm{T}}_{\underline{\boldsymbol{x}}_{q_i}}\boldsymbol{J}_{\underline{\boldsymbol{x}}_{q_i}}\right)$$

The # symbol defines the robust pseudo-inverse of a matrix $\boldsymbol{M}$, i.e., $\boldsymbol{M}^{\#} = \boldsymbol{M}^{\mathrm{T}}\left(\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}} + \lambda_\epsilon \boldsymbol{I}\right)^{-1}$, where $\boldsymbol{I}$ is the identity matrix weighted by $\lambda_\epsilon > 0$ for damping the pseudo-inverse. Furthermore, $\underline{\boldsymbol{x}}_{r,0}$ denotes the relative pose at the beginning of the task.

The control joint velocity $\dot{\boldsymbol{q}}_{\mathrm{c}}$ and feed-forward joint velocity $\dot{\boldsymbol{q}}_{\mathrm{ff}}$ are defined as follows (see (56) and (57)). The resulting joint velocity $\dot{\boldsymbol{q}}$ is integrated and the resulting desired position is controlled by a low level joint impedance controller. As the distance and velocity between two consecutive goals can vary, the instantaneous Cartesian control goal $\boldsymbol{p}_{\mathrm{i}}$ is interpolated according to the actual velocity v_{act} of the robot, in order to ensure a smooth movement

$$\boldsymbol{p}_{\mathrm{i}} := \boldsymbol{p}_{\mathrm{i}} + v_{\mathrm{act}} T \frac{\boldsymbol{p}_d - \boldsymbol{p}_{\mathrm{i}}}{\left|\boldsymbol{p}_d - \boldsymbol{p}_{\mathrm{i}}\right|_2}, \tag{54}$$

and whenever $|\boldsymbol{p}_d - \boldsymbol{p}_{\mathrm{i}}|_2/T < v_{act}$, $\boldsymbol{p}_d$ is updated. If v_{act} is below the commanded velocity for the current segment

$$v_{\mathrm{ts}} = \frac{\boldsymbol{p}_{\mathrm{d}}(t) - \boldsymbol{p}_{\mathrm{d}}(t-T)}{T}, \tag{55}$$

where T is the planning cycle time, it is increased in each time step by a constant acceleration (we used a constant increment of 0.05 m/s) until v_{ts} is reached to ensure a smooth transition to higher velocities.

The feed-forward joint velocity

$$\dot{\boldsymbol{q}}_{\mathrm{ff}} := v_{\mathrm{act}} \dot{\tilde{\boldsymbol{q}}}_{\mathrm{ff}} \frac{1}{\left|\dot{\tilde{\boldsymbol{q}}}_{\mathrm{ff}}\right|_2}, \tag{56}$$

with $\dot{\tilde{\boldsymbol{q}}}_{\mathrm{ff}} := \left(\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}\boldsymbol{J}_{\boldsymbol{p}_a}\right)^{\#}\begin{pmatrix}\boldsymbol{0}^{\mathrm{T}} & \boldsymbol{p}^{\mathrm{T}}_{\mathrm{i}}(t+T) - \boldsymbol{p}^{\mathrm{T}}_{\mathrm{i}}(t)\end{pmatrix}^{\mathrm{T}}$ and the control velocity

$$\begin{aligned}\dot{\boldsymbol{q}}_{\mathrm{c}} &:= k_{\mathrm{pos}} \dot{\tilde{\boldsymbol{q}}}_{\mathrm{c}} \frac{1}{T\left|\dot{\tilde{\boldsymbol{q}}}_{\mathrm{c}}\right|_2}, \\ \text{where} \quad \dot{\tilde{\boldsymbol{q}}}_{\mathrm{c}} &:= \boldsymbol{J}^{\#}_{\boldsymbol{p}_a}(\boldsymbol{p}_{\mathrm{i}} - \boldsymbol{p}),\end{aligned} \tag{57}$$

enable the robot to reach the instantaneous control goal $\boldsymbol{p}_{\mathrm{i}}(t)$ in each time step. The relative pose and the absolute position Jacobians are denoted by $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$ and $\boldsymbol{J}_{\boldsymbol{p}_a}$, and $\boldsymbol{p}$ is the current position of the robot according to the commands, i.e. assuming perfect low-level control. The parameter $0 \leq k_{\mathrm{pos}} \leq 1$ is the position control gain. Due to the use of $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$ in (56), the feed-forward velocity respects the relative orientation of the robot and is thus not reduced when being projected into the nullspace of the relative pose controller.

IX. Simulations and Quantitative Analyses

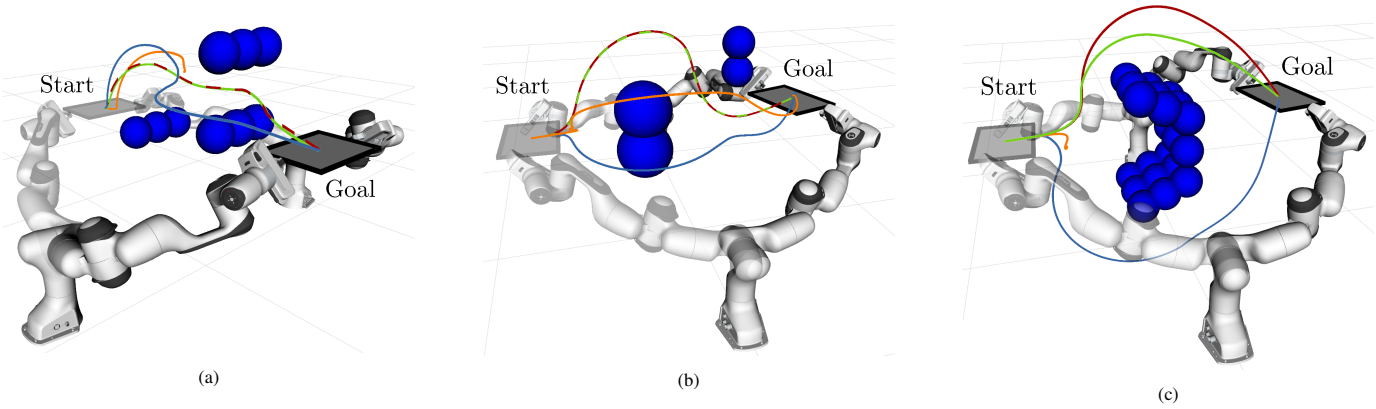
To demonstrate the effectiveness of our planning approach, we compared our proposed motion planner with other state-of-the-art reactive planning methods [8], [46], [68] including the work from [42] in several challenging scenarios. These planning scenarios include multiple random-placed dynamic obstacles as well as static and dynamic trap-like barriers obstacles, which generally pose major challenges for reactive planning approaches. Additionally, to test portability of our framework we considered different dual-arm robot setups, for instance, a humanoid-like KoBo robot and a cooperative setup with two table-mounted Franka-Emika research robots [78].

A. Implementation Setup

For the simulations, a laptop running Ubuntu 18.04 (power-save mode) with an Intel® Core i9-9880H CPU with 8 cores and a base frequency of 2.30 GHz was used. All controllers and robot kinematic models were implemented using the DQ-Robotics library [79]. The low-level controller, running at 1 kHz, is programmed in unoptimized C++ using libraries provided by the robot manufacturer, and exploiting the ROS control framework [80]. The planner, on the other hand, runs at a frequency of 100 Hz in a different process and exchanges information with the control in the form of messages. Note that due to the restrictive time constraints within our robot control framework (time step requirements: $T_c < 10\,\mathrm{ms}$) comparisons with classical sense-plan-act approaches are not feasible.

The cooperative control priorities deployed for all simulations are maintaining the relative pose, the absolute distance to the prescribed Cartesian path – that is the abs. distance funnel, the tilt-angle of the shared tray, and remaining within the joint-limits with gains of κ_{rp}=0.005, κ_{ee}=0.1, κ_{ap}=1.0, and κ_{jl}=5.0, respectively. The gains and planner parameters were not optimized and perform similarly throughout multiple scenarios. To ensure replicability, all simulation parameters are available in our accompanying data repository [81]. Overall, parameters are $n_a = 10$, $r_r = 0.05$, $r_d = 0.35$, $k_{\mathrm{cf}} = 0.015$, $k_a = 4.0$, $k_d = 4.0$, $k_r = 0.08$, $v_{\max} = 0.2\,\mathrm{m/s}$, $a_{\max} = 13.0\,\mathrm{m/s^2}$. The cost function for evaluating the agents uses the weights $w_{pl} = 10.0$, $w_{gd} = 100.0$, $w_{od} = 0.001$, $w_{ws} = 1.0$.

Detailed information on the scenarios and parameters for all simulations is available in [81]. This data repository for the paper also includes videos for all generated simulations.



(a) (b) (c)

Fig. 9: Comparison of the different planners in the three static environments with two Franka Emika robots cooperatively grasping a tray. In subfigure (a), the narrow passage scenario *Dual Arm Static 1* is depicted. The scenarios *Dual Arm Static 2* and *Dual Arm Static 3* with disparate obstacle arrangements are shown in subfigure (b) and (c), respectively. The trajectory of the CFP planner is depicted in green, the CF planner in red, MFI in blue, and APF in orange.

TABLE III: QUANTITATIVE COMPARISON OF THE VARIOUS REACTIVE PLANNING APPROACHES IN 30 RANDOM DYNAMIC SCENARIOS.

Environments (Scenes)	**Planner**	**Number of runs**	**Success Rate** [%]	**Path Length** [m]	**Tracking Error** [mm]	**Best Agent Switches**
Dual Arm Dynamic	CFP	30	96.67 %	1.39 ± 0.07	2.7 ± 1.2	0.57 ± 0.60
	CF	30	53.33 %	1.32 ± 0.03	2.2 ± 0.4	-
	MFI	30	23.33 %	1.96 ± 0.22	63.7 ± 40.3	-
	APF	30	3.33 %	1.99 ± 0.00	11.1 ± 0.0	-

TABLE IV: COMPARISON WITH STATE-OF-THE-ART PLANNING APPROACHES IN VARIOUS SCENARIOS.

Envs. (Scenes)	**Planner**	**Suc.**	**Path Len.** [m]	**Tracking Err.** [mm]	**B. Agent Switches**
Dual Arm Static 1	CFP	Yes	1.37	1.9	1
	CF	Yes	1.37	1.3	-
	MFI	No	-	-	-
	APF	No	-	-	-
Dual Arm Static 2	CFP	Yes	1.66	2.0	0
	CF	Yes	1.66	1.6	-
	MFI	No	-	-	-
	APF	No	-	-	-
Dual Arm Static 3	CFP	Yes	1.51	2.4	1
	CF	Yes	1.67	2.2	-
	MFI	Yes	2.35	0.9	-
	APF	No	-	-	-
KoBo Dynamic 1	CFP	Yes	1.22	0.3	4
	CF	Yes	1.18	0.4	-
	MFI	Yes	1.31	0.2	-
	APF	Yes	1.32	0.2	-
KoBo Dynamic 2	CFP	Yes	1.12	0.2	3
	CF	Yes	1.21	0.4	-
	MFI	Yes	1.27	0.3	-
	APF	Yes	1.36	0.2	-
KoBo Dynamic 3	CFP	Yes	1.22	0.1	5
	CF	No	-	-	-
	MFI	Yes	1.26	0.2	-
	APF	Yes	1.51	0.2	-

B. Results and Discussion

The results for the expository scenarios consisting of trap-like barrier obstacles, three examples are shown in Fig. 9 (rest can be found in [81]), are summarized in Table IV. We compare our proposed framework (**CFP**) against state-of-the-art circular field approaches (**CF**), e.g., [8], [9], [46], magnetic field (**MFI**), e.g., [68], as well as standard artificial potential fields (**APF**) [42].

The results for the illustrative cases clearly show that our proposed CFP planner, is the only planner that successfully finds a collision-free path across all the designed environments. This result emphasizes the reliability of the proposed framework based on global environment exploration with local control relaxations. The tracking flexibilities, e.g., following a funnel rather than a straight rigid path is highlighted in the slighter higher tracking error. On the other hand, the CFP planner finds the shortest path in almost all scenarios mostly due to its larger exploration space.[8] For instance, consider the KoBo Dynamic 2 scenario, where all the planners were able to compute a path to the goal. Our CFP planner generated the shortest path with a length of $1.12\mathrm{m}$ while APF had the longest ($1.36\mathrm{m}$). The proposed framework is also capable of running well below the real-time limits. The average step time for calculating the next reference position is 695.18 µs. The tracking error of $0.2\mathrm{mm}$ was also the lowest in this environment with our CFP, which is the same as using APF. We also report that the average step time for the controller, that is, preparing and computing the augmented Jacobians, projectors, inverses, and the final desired joint velocities for the cooperative system is $641.56(\pm 396)$ µs. As far as agent switching is concerned, the most switches (5) occurred for the KoBo Dynamic 3 scene.

Our quantitative analysis supports these findings (cf. Tab. III). Herein, we additionally analyzed the different planning approaches in 30 distinct complex environments with three dynamic obstacles. In each trial, we randomly changed the size, initial position, and/or velocity of the obstacles. Two of these obstacles were always placed to form a barrier and posed a major challenge for the APF approach, which only succeeded in a single trial out of the 30 runs that were conducted per planner. Notably, the CF planner resulted in lowest tracking error while MFI planners were the worst. Regarding path lengths, CFP and CF planners resulted in lengths of $\approx 1.3\mathrm{m}$ whereas MFI and AFP planners had lengths of $\approx 1.9\mathrm{m}$. Overall, the advantages

[8]Note that the multi-agent simulation for finding the best agent settings (which corresponds to a global path planning query) is called with $T_p = 10\,\mathrm{ms}$ in the current implementation and while most of the agent simulations are done in parallel, there is still a considerable computation overhead. Thus, the calculation step time can be lowered even more by calling the multi-agent simulation less frequently.

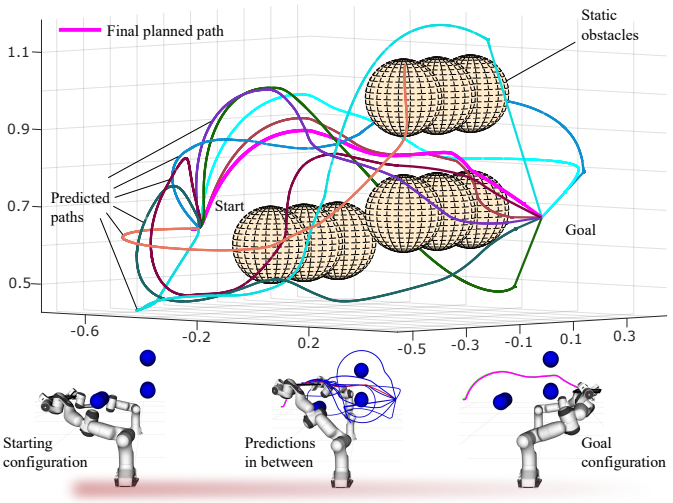


Fig. 10: Predicted and planned paths for a static narrow passage scenario used in the simulation studies. Here, the number of explorer agents, n_a, was set to 10. The Cartesian absolute position of the cooperative system during starting is at $(-0.7567, 0.0063, 0.6185)$ and the goal is defined to be $(0.5, 0.0, 0.7)$. It can be seen that the final planned path evades the different obstacles in a sub-optimal fashion and also respects the constraints of the dual-arm system.

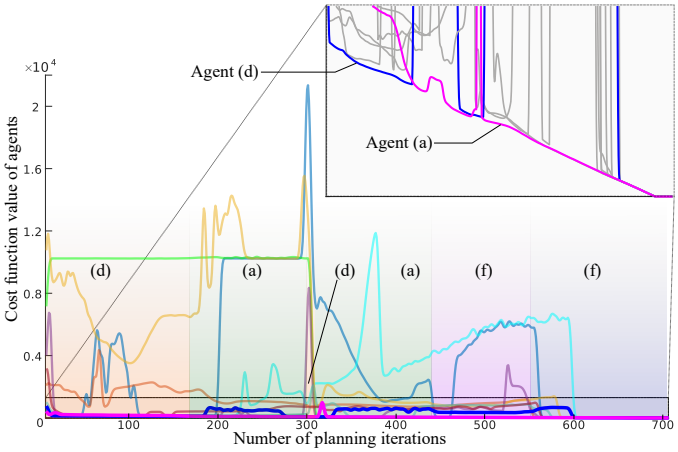


Fig. 11: The cost function value of all agents used in the scenario *KoBo Dynamic 3*. For better visualization purposes the original data has been smoothed out with a Gaussian-weighted moving average filter with a window of length 10. It is important to note that, as the planning horizon comes to an end, the multi-agents converge. The orange area denotes the approximate time *agent (d)* was selected as best agent, and the green area was dominated by *agent (a)*. In the purple and grey areas the parameter sets of two different agents of type *agent (f)* were used.

and mainly reliability of the proposed framework are reflected in the considerably higher success rate compared to other approaches. While CF planners reach 53%, our proposed CFP framework achieves a success rate of $\approx 97\%$. The only failure case with our approach occurred due to joint limits in a multiple simultaneous boundary constraint scenario, in which all other planners also failed.

Note that the CFP exhibited slightly longer path lengths on average than the CF planner, despite having a higher overall success rate. This observation can be attributed to the fact that the CF planner only succeeds in simpler scenarios, which require shorter paths, skewing its average path length towards shorter values.

In Fig. 10, we visualized the predicted paths of all explorer agents at the initial time step. The presented visualization clearly shows the ability of the diverse heuristics and randomized agents to comprehensively cover and evaluate numerous paths for obstacle avoidance.

The predictive multi-agent framework demonstrates an additional advantage in the scenarios *Dual Arm Static 3* (cf. Fig. 9 (c)) and *KoBo Dynamic 3* (cf. Fig. 11). Both scenarios show that switching between agent parameter sets during the execution yields improved results compared to solely evaluating an initial cost function and utilizing a single agent parameter set. In Fig. 9 (c) the CFP planner initially adopts the same parameter set as the CF planner but switches to a more optimal parameter set just before passing the obstacles. More parameter set switches in one run are also possible as can be observed in Fig. 11, which shows the costs of all agents at each time step. Here, the real robot switches between the parameter sets of *agent (d)*, *agent (a)* and two distinct random agents, i.e., *agent (f)*. The switching occurs when the priority of the absolute pose in the CoSTP controller is low and the real robot is no longer able to follow the predicted trajectory. Consequently, the cost of the current best agent increases and the parameter set of the new best agent is adopted (cf. iterations 180, 298, 316, 449 and 559 in Fig. 11). More detailed comparisons of our reactive planner CFP, including comparisons with global, non-reactive planners are in our previous publications [10], [57].

C. Ablation Studies

Furthermore, in order to test the robustness of the task hierarchy choice and the switching strategy, we perform ablation studies. For this, we use the constrained *Dual Arm Static* 1 scenario described in Fig. 9 (a) of the paper. All scenarios are run with 5 explorer agents. A comparison was made of the following two control strategies:
Exp 1: Task switching is **active** through CoSTP: This is the basic scenario and as expected, the system is able to reach the goal without any difficulty.
Exp 2: Task switching is **inactive**: The task hierarchy, maintained from the commencement to the end, in this case was: (1) Relative pose (2) Joint limits (3) Absolute Pose (4) EE Tilt. We found that the system was unable to track the absolute pose of the cooperative task space after a certain point in time. The explanation would be that during the evolution of the complex task, the number of degrees of freedom in the (static) nullspace gets diminished significantly, and the 3rd and 4th tasks get executed only with a reduced priority.

Note that the gains for tasks $1, 2, 3,$ and 4 were fixed throughout the experiments and deployed several times to ensure consistency with values of $0.005, 2.0, 1.0,$ and 0.1, respectively. The key point is that static solutions lead to infeasible results, and are dependent on tightly tuned control construction which might fail with slight change in scenarios. *Reduced Joint-limits:* To further highlight and evaluate the performance of the CoSTP we perform a series of experiments where we artificially reduce the joint limit span. The range of joints $1, 2, 3, 4, 5,$ and 7 are reduced by a factor of $\approx 12\%$. Joint 6 has a reduced limit originally and therefore was lowered by a factor of $\approx 5\%$. Even in this highly constrained setup, the framework is still able to generate feasible control signals that

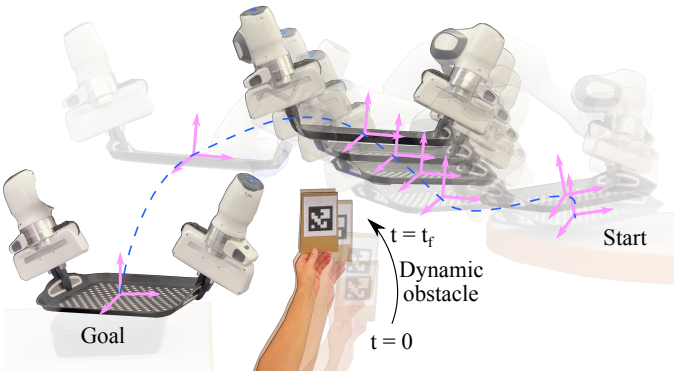


Fig. 12: Non-stationary environments: Illustration demonstrating dynamic obstacle avoidance (the ArUco marker denotes the centre of the obstacle).

complete the task successfully. Using the suitable gains that we choose, the controller reaches a globally asymptotically stable equilibrium point in all modes. Our algorithm switches effectively within modes that fulfill the equality tasks (such as maintaining the relative pose) while ensuring the set-based tasks (such as the joint limits) are not violated.

Although harsher penalty functions can be used for the joint limit avoidance task (cf. subsection V-E), this experiment demonstrates that with robust task-switching, the different tasks we consider can be achieved in a dual-arm scenario. We consider the choice of the function an exciting topic for future investigation, including selective activation and de-activation of joints instead of pushing all joints toward their centers.

X. Real Robot Experiments

In this section, we present thorough real-world experimental evaluation of the planner and control framework. As the experimental test bed, we used the KoBo setup with two Franka-Emika research robots, as shown in Fig. 1. The experiments were executed using an Intel® Core i7-7700 processor with 4 CPUs and 2 threads per core, having a clock speed of 3.60 GHz and running Ubuntu 18.04 LTS operating system with a real-time kernel. Otherwise, the software structure and implementation are identical to the simulations. The vision thread, explained later, runs in a separate computer for computational load reduction and due to the restrictions stemming from the use of a real-time kernel. Indeed, we emphasize the fact that the controller for the Franka Emika robots needs access to a real-time kernel with a thread running below 1 kHz; otherwise, there would be a communication failure. Additionally, no code optimization was done for the computer system which was set to the default powersave mode. We report and inspect 2 case studies and the additional 10+ trials in the form of videos can be found in [81]. Herein, we focus on 2 distinct case studies, as well as an assessment of a landing experiment, to demonstrate the real-time evasive strategies in fast-changing environments as well as in highly constrained static scenes.

A. Case Study #1: Reactive planning with dynamic obstacles

The central intention of this experiment is to demonstrate the reactive capability and, mainly, the reliability of the proposed framework. As shown in Fig. 12 and the accompanying video

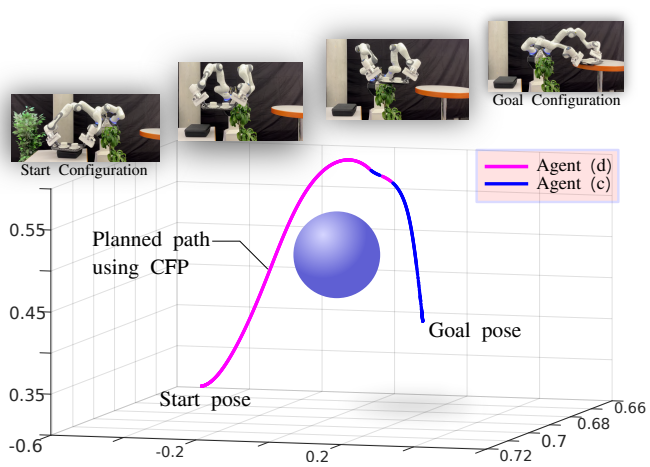


Fig. 13: Planning in static environments: As can be seen in the snapshots at the top, the dual arm system successfully avoids the obstacle (plant) and releases the tray on the table-top. The plot below exemplifies our agent switching strategy during the final planned path for the constrained scenario.

attachment, the robot successfully executes a pick and place task after the human places the tray at the pick-up location. The non-stationary obstacle is then introduced midway along the path which is detected by a real-sense D435 through an attached aruco-marker. We use fiducial markers to simplify the visual sensing aspect of robotic experiments. For detection, we use the ArUco module of OpenCV [82]. For the calibration process we obtain the marker poses in the robot frame by estimating the fixed SE(3) transform between the robot base and the camera frame. To solve this estimation problem, we use the ViSP package [83].

The cooperative control primitives deployed in this study are maintaining the relative pose between arms, the absolute distance to the originally planned trajectory, i.e., abs. distance funnel, the tilt-angle of the shared tray, and remaining within the joint limits with gains of κ_{rp}=0.005, κ_{ap}=0.9, κ_{ee}=0.1, and κ_{jl}=5.0, respectively. The planning parameters are $n_a = 4$, $r_r = 0.05$, $r_d = 0.3$, $k_{\text{cf}} = 0.001$, $k_a = 4.0$, $k_d = 4.0$, $k_r = 0.08$, $k_\sigma = 0.0$, $k_\omega = 10.0$, $f_o = 1.5$ for forces and $f_o = 0.0$ for torques. The remaining parameters are the same as the simulations in the previous section. Our robust strategy ensures that the set-based constraints remain satisfied at all times. Despite the fact that continued disturbance is added in the form of an unforeseen obstacle, as is illustrated in Fig. 12, the evasion for the elaborate system is made possible through intelligent agent switching as well as flexible task adjustment.

B. Case Study #2: Planning - static scenario

In this second scenario, the robotic task is to plan a path in a highly constrained scenario, as depicted in Fig. 13. Note the main obstacle, i.e., the plant, covers most of the reachable space of the cooperative system.

After reaching the goal, we also want the system to detect uncertain landing surfaces and release the tray in a human-like fashion. More specifically, our objective here is to examine the system in a case where the workspace is limited and constrained dual-arm manipulation is not trivial. The planning and control parameters are the same as in the last experiment except for the rotational admittance in SLAC which is k_ω=0.01. The plots in Fig. 16 indicate that the joints of the left arm of

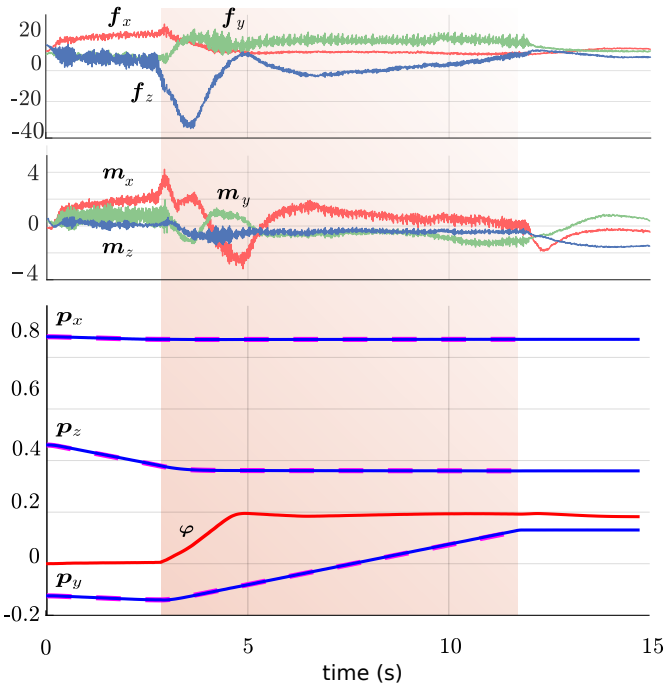


Fig. 14: Trajectory for a landing ($\boldsymbol{p}$) with a sliding motion along the y axis, and normal angle (φ) along z-axis, and external forces ($\boldsymbol{f}$) and torques ($\boldsymbol{m}$) according to (50). The landing takes place in the marked area of the plot – starting with the contact detection and ending when the final goal is reached by sliding the object over the table.

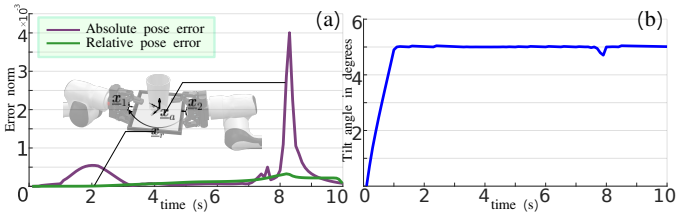


Fig. 15: Data from real robot experiments: (a) Absolute and relative pose errors for the trial explained in Sec. X-B. Both error norms decay to 0 in the end. (b) Tray tilt (reference was set to 5°) during the entire task horizon.

the KoBo system remain within the safe bounds and Fig. 15 reports the error in the cooperative variables (absolute and relative pose) and the evolution of the tray tilt during the task horizon. The two peaks in for the absolute pose error reflect the fact that the absolute distance to the trajectory task was the second priority task at those points. Furthermore, the abs. pose combines both Cartesian and orientation errors, and the second peak matches the titl error due to the contact instant with the table. Overall, the peaks are also in accordance with the switching illustration of the different controllers in Fig. 7. The tilt angle is commanded and maintained at $5°$ for feasible task completion. Fig. 13 reinforces our claim of adaptive agent switching between type 4 and type 3 as the robot avoids static obstacles in the scene.

C. Analysis of the Safe Landing Capability

This experiment was conducted together with previous ones, and highlights the scenario where the dual-arm system is supposed to safely release the cup-tray system after reaching the landing location. The goal here is to validate our SLAC strategy. The final landing act is executed by sliding on the surface of the table along the y-axis of the world frame after which the system comes to a rest. When the contact is detected, the controller switches between the tilt controller and instead deploys the rotational admittance controller. As it lands by sliding on the table (along the y-axis), the planner needs to adapt its orientation, as it starts with an angle of $15°$ towards the table. The entire duration of the event is 15 seconds, yet the actual landing strategy happens around the marked area and takes roughly 8 seconds. The control parameters used are $k_\omega = 0.5$, $f_o = 6$ for the force, and $f_o = 1.5$ for the torque.

Fig. 14 illustrates the behavior of the system during this task. The top panel demonstrates the fact that the external forces and moments (wrenches) adapt accordingly in order to bring about a stable landing. As expected, the initial high force of -40 N in the z direction during the contact phase can be seen to decrease asymptotically as soon as the planner switches priority and focuses on surface adaptation. Moments on the other hand, stay approximately in the range of -4 to 4 Nm with the moment in the x direction reacting the most to the sensed torques. The expected human-like alignment of the tray to the table is demonstrated with the translation $\boldsymbol{p}$ and normal angle φ data in the bottom panel. After the detection of the contact, the tilt controller is swapped with the rotational admittance controller.

XI. Conclusion, Discussions, and Future Outlook

This work introduces a novel real-time planner-controller framework for coordinated dual-arm tasks in fast-changing environments. Our vector-field-based planning strategy exploits multi-agent exploration of the environment, thereby resulting in robust global exploration proficiency with high reliability. Further, our framework records and integrates environmental wrenches which are fed back into the planner and also allows us to achieve a common bimanual task like placing a tray in a chosen location in a safe manner. The distinct modeling strategy that we follow utilizes coupled analysis of the task space and captures constraints with fewer degrees of freedom compactly, thanks to our usage of unit dual quaternions. We then systematically formalize geometric subsets of interest (primitives) in the shared task space that we control flexibly by implementing a switching scheme. The extra degrees of freedom of the system are thoroughly used to design nullspace projectors to accommodate these additional primitives and ease task execution. Our approach can also be extended to n manipulators given the CDTS abstraction [27]. The modifications that would facilitate this are additional frames in the CDTS, the corresponding new analytical Jacobians, and the augmented joint space.

Although control strategies based on quadratic programming (QP) bring in interesting perspectives in this regard [84]–[86], they are not necessarily better. Our CoSTP framework provides a closed-form solution to a highly constrained problem and does not suffer from the exponential costs arising from inequality constraints. The CFP combined with the CoSTP form an anytime algorithm, resulting in better solutions with time. Moreover, we have feedback at each step of the planning and control cycle which is not possible with QP-based methods.

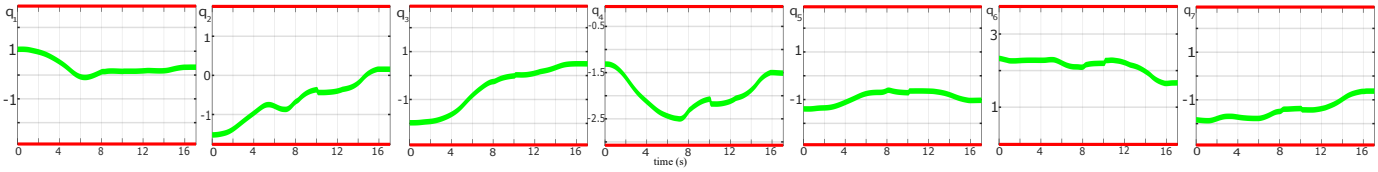


Fig. 16: Commanded joint positions of the left arm (depicted in green) along with their limits (red) from the real robot experiments described in section X. Our joint limit controller ensures that all the joints are always pushed toward the center of their ranges.

Comprehensive analyses demonstrate that our proposed framework achieves better performance than current state-of-the-art reactive planners in the context of dual-arm manipulation in stationary and non-stationary settings. To this end, we fuse the environment information through vision sensing, which is later used for effective lightweight prediction computation. The evasive maneuvers generated from the predictions help us choose globally sub-optimal paths for reaching a specific target location. Experiments also show that our multi-agent strategy eliminates the need for constant parameter tuning with the change in environments. Furthermore, the problem of local minima is also addressed by the predictive switching scheme. Lastly, through admittance control, we are able to adapt to different landing surfaces and release objects intuitively. Overall, the main contribution of the proposed framework is the design of a reliable reactive planner for dual-arm and high-DoF systems – in terms of timely finding feasible solutions in different setups without any redesign or parameter tuning.

Some of the limitations of our approach include local adaptation of the manipulability of the entire system through artificial forces. In the future, we aim to alleviate this problem by incorporating embodied manipulability maps which help to pass global manipulability information to field planners, in turn facilitating the analysis of geometric capability of the robotic system at task-specific locations [87]. Also, a more insightful analysis of the environment representation is expected to handle complex coordinated maneuvers in a way that object affordances are intrinsically encoded. The introduction of obstacle detection uncertainty and testing planner robustness is yet another avenue for extensions in the future, along with supplements for handling complex uncoordinated tasks.

APPENDIX A
MATHEMATICAL PRELIMINARIES: DUAL QUATERNIONS

The dual quaternion algebra [88] describes rigid body motion algebraically in a similar way that quaternions represent rotations in three-dimensional space. A unit quaternion $\boldsymbol{r} = \cos(\phi/2) + \sin(\phi/2)\boldsymbol{n}$ represents a rotation with angle ϕ around the axis $\boldsymbol{n}$ [89]. The inverse operation is given by the conjugate $\boldsymbol{r}^* = \cos(\phi/2) - \sin(\phi/2)\boldsymbol{n}$. The one-to-one mapping from the quaternion space to $\mathbb{R}^4$ is governed by the $\mathrm{vec}\,4 : \mathbb{H} \to \mathbb{R}^4$ operator. Essentially, the operator just takes the coefficients of the quaternion and stacks them in a vector. The algebra of dual quaternions is an algebra over $\mathbb{R}^8$ composed of $\mathbb{H} \otimes \mathbb{D}$ where $\mathbb{H}$ represents the set of quaternions and $\mathbb{D}$ denotes the set of dual numbers. In this paper, we used $\underline{\boldsymbol{S}}$ to refer to the set of unit dual quaternions which can be mathematically defined as follows,

$$\underline{\boldsymbol{S}} \triangleq \{\underline{\boldsymbol{x}} \in \mathrm{Spin}(3) \ltimes \mathbb{R}^3 : \|\underline{\boldsymbol{x}}\| = 1\} \tag{58}$$

where elements of $\underline{\boldsymbol{S}}$ furnished with the operation of multiplication represent elements of $\mathrm{Spin}(3) \ltimes \mathbb{R}^3$. Analogous to quaternion algebra [90], dual quaternions are also non-commutative, that is, for two dual quaternions $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$, $\underline{\boldsymbol{x}}\underline{\boldsymbol{y}} \neq \underline{\boldsymbol{y}}\underline{\boldsymbol{x}}$. Therefore, one way is to exploit the matrix form of the algebra to commute terms when performing dual quaternion multiplications (as in the case of Clifford algebra).

REFERENCES

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[2] F. Krebs and T. Asfour, "A Bimanual Manipulation Taxonomy," *IEEE Robot. Autom. Lett. (RAL)*, vol. 7, no. 4, 2022.

[3] S. P. Swinnen and J. Gooijers, *Bimanual Coordination*. Elsevier Inc., 2015, vol. 2.

[4] R. N. Colomba, R. Laha, L. F. Figueredo, and S. Haddadin, "Adaptive admittance control for cooperative manipulation using dual quaternion representation and logarithmic mapping," in *Proc. IEEE Conf. Dec. and Contr.* IEEE, 2022, pp. 107–144.

[5] W. Thibault, V. Rajendran, and K. Mombaur, "Bimanual manipulation workspace analysis of humanoid robots with object specific coupling constraints," in *Proc. IEEE-RAS Int. Conf. Human. Robots*. IEEE, 2022, pp. 141–148.

[6] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wuthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg, "Real-time perception meets reactive motion generation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1864–1871, 2018.

[7] D. Kragic, J. Gustafson, H. Karaoguz, P. Jensfelt, and R. Krug, "Interactive, collaborative robots: Challenges and opportunities," in *IJCAI International Joint Conference on Artificial Intelligence*, 2018, pp. 18–25.

[8] R. Laha, L. F. Figueredo, J. Vrabel, A. Swikir, and S. Haddadin, "Reactive cooperative manipulation based on set primitives and circular fields," in *2021 Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2021, pp. 6577–6584.

[9] R. Laha, J. Vorndamme, L. F. Figueredo, Z. Qu, A. Swikir, C. Jähne, and S. Haddadin, "Coordinated motion generation and object placement: A reactive planning and landing approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2021.

[10] M. Becker, T. Lilge, M. A. Müller, and S. Haddadin, "Circular fields and predictive multi-agents for online global trajectory planning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2618–2625, 2021.

[11] R. Laha, M. Becker, J. Vorndamme, J. Vrabel, L. F. Figueredo, M. A. Müller, and S. Haddadin, "Repository: Predictive multi-agent based planning and landing controller for reactive dual-arm manipulation." 2023. [Online]. Available: https://github.com/riddhiman13/predictive-multi-agent-framework

[12] J. Złotowski, D. Proudfoot, K. Yogeeswaran, and C. Bartneck, "Anthropomorphism: opportunities and challenges in human–robot interaction," *International journal of social robotics*, vol. 7, no. 3, pp. 347–360, 2015.

[13] C. Bartneck, T. Belpaeme, F. Eyssel, T. Kanda, M. Keijsers, and S. Šabanović, *Human-robot interaction: An introduction*. Cambridge University Press, 2020.

[14] E. Roesler, D. Manzey, and L. Onnasch, "A meta-analysis on the effectiveness of anthropomorphism in human-robot interaction," *Science Robotics*, vol. 6, no. 58, p. eabj5425, 2021.

[15] D. Almeida and Y. Karayiannidis, "Asymmetric dual-arm task execution using an extended relative jacobian," in *Robotics Research: The 19th International Symposium ISRR*. Springer, 2022, pp. 18–34.

[16] B. Nemec, N. Likar, A. Gams, and A. Ude, "Adaptive human robot cooperation scheme for bimanual robots," *Advances in Robot Kinematics 2016*, pp. 371–380, 2018.

[17] F. Caccavale and M. Uchiyama, "Cooperative manipulation," *Springer handbook of robotics*, pp. 989–1006, 2016.

[18] D. Surdilovic, Y. Yakut, T. M. Nguyen, X. B. Pham, A. Vick, and R. Martin-Martin, "Compliance control with dual-arm humanoid robots: Design, planning and programming," in *Proc. IEEE-RAS Int. Conf. Human. Robots*. IEEE, 2010, pp. 275–281.

[19] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation—a survey," *Robot. Auton. Syst.*, vol. 60, no. 10, pp. 1340–1353, 2012.

[20] H. A. Park and C. G. Lee, "Extended cooperative task space for manipulation tasks of humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2015, pp. 6088–6093.

[21] H. Park and C. G. Lee, "Dual-arm coordinated-motion task specification and performance evaluation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2016, pp. 929–936.

[22] R. S. Jamisola, P. S. Kormushev, R. G. Roberts, and D. G. Caldwell, "Task-space modular dynamics for dual-arms expressed through a relative jacobian," *J. Intell. Robot. Sys.*, vol. 83, pp. 205–218, 2016.

[23] A. Sinha and N. Chakraborty, "Robust relative hand placement for bi-manual tasks," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 83990. American Society of Mechanical Engineers, 2020, p. V010T10A057.

[24] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr 1988, pp. 350–356 vol.1.

[25] C. L. Lewis and A. A. Maciejewski, "Trajectory generation for co-operating robots," in *1990 IEEE International Conference on Systems Engineering*. IEEE, 1990, pp. 300–303.

[26] P. Chiacchio, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system," *Journal of Dynamic Systems, Measurement, and Control*, 1996.

[27] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2010, pp. 3955–3960.

[28] L. F. C. Figueredo, B. V. Adorno, J. Y. Ishihara, and G. A. Borges, "Switching strategy for flexible task execution using the cooperative dual task-space framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1703–1709.

[29] S. Chakravorty and S. Kumar, "Generalized sampling-based motion planners," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 855–866, 2011.

[30] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *Ieee access*, vol. 2, pp. 56–77, 2014.

[31] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[32] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom..*, vol. 2, 2000, pp. 995–1001 vol.2.

[33] M. Bonilla, L. Pallottino, and A. Bicchi, "Noninteracting constrained motion planning and control for robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2017.

[34] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 625–632.

[35] F. Burget, M. Bennewitz, and W. Burgard, "Bi2rrt*: An efficient sampling-based path planning framework for task-constrained mobile manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 3714–3721.

[36] M. Otte and E. Frazzoli, "Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.

[37] M. P. Strub and J. D. Gammell, "Advanced bit (abit): Sampling-based planning with advanced graph-search techniques," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 130–136.

[38] F. Grothe, V. N. Hartmann, A. Orthey, and M. Toussaint, "St-rrt*: Asymptotically-optimal bidirectional motion planning through space-time," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3314–3320.

[39] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using rrt* based approaches: a survey and future directions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

[40] S. Li and N. T. Dantam, "Sample-driven connectivity learning for motion planning in narrow passages," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2023.

[41] M. Braquet and E. Bakolas, "Vector field-based collision avoidance for moving obstacles with time-varying elliptical shape," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 587–592, 2022.

[42] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2. IEEE, 1985, pp. 500–505.

[43] J.-O. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Trans. Robot. Autom.*, 1992.

[44] L. Singh, H. Stephanou, and J. Wen, "Real-time robot motion control with circulatory fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 1996, pp. 2737–2742 vol.3.

[45] L. Singh, J. Wen, and H. Stephanou, "Motion planning and dynamic control of a linked manipulator using modified magnetic fields," in *Proceedings of the 1997 IEEE International Conference on Control Applications*. IEEE, 1997, pp. 9–15.

[46] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6842–6850, 2011.

[47] M. Becker, J. Köhler, S. Haddadin, and M. A. Müller, "Motion planning using reactive circular fields: A 2D analysis of collision avoidance and goal convergence," *IEEE Transactions on Automatic Control*, 2023, early access. [Online]. Available: https://doi.org/10.1109/TAC.2023.3303168

[48] A. M. Hussein and A. Elnagar, "Motion planning using maxwell's equations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3. IEEE, 2002, pp. 2347–2352.

[49] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[50] Y. Yang, V. Ivan, Z. Li, M. Fallon, and S. Vijayakumar, "idrm: Humanoid motion planning with realtime end-pose selection in complex environments," in *Proc. IEEE-RAS Int. Conf. Human. Robots*, 11 2016, pp. 271–278.

[51] H. Liu, D. Qu, F. Xu, Z. Du, K. Jia, J. Song, and M. Liu, "Real-time and efficient collision avoidance planning approach for safe human-robot interaction," *J. Intell. Robot. Syst.*, vol. 105, no. 4, p. 93, 2022.

[52] B. Lacevic and P. Rocco, "Kinetostatic danger field-a novel safety assessment for human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2010.

[53] B. Lacevic, P. Rocco, and A. M. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1257–1270, 2013.

[54] A. M. Zanchettin, B. Lacevic, and P. Rocco, "A novel passivity-based control law for safe human-robot coexistence," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2012, pp. 2276–2281.

[55] A. Sinha, R. Laha, and N. Chakraborty, "Oc3: A reactive velocity level motion planner with complementarity constraint-based obstacle avoidance for mobile robots," in *IEEE Int. Conf. Autom. Sci. Eng.* IEEE, 2023.

[56] N. Malone, H.-T. Chiang, K. Lesser, M. Oishi, and L. Tapia, "Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field," *IEEE Transactions on Robotics*, 2017.

[57] M. Becker, P. Caspers, T. Hattendorf, T. Lilge, S. Haddadin, and M. A. Müller, "Informed circular fields for global reactive obstacle avoidance of robotic manipulators," in *22nd International Federation of Automatic Control World Congress (IFAC WC 2023)*. IFAC-PapersOnLine, 2023.

[58] J. Funda and R. Paul, "A computational analysis of screw transformations in robotics," *IEEE Trans. Robot. Autom.*, 1990.

[59] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robotics and Autonomous Systems*, vol. 77, pp. 66 – 73, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889015301184

[60] N. T. Dantam, "Robust and efficient forward, differential, and inverse kinematics using dual quaternions," *The International Journal of Robotics Research*, vol. 40, no. 10-11, pp. 1087–1105, 2021.

[61] L. Figueredo, B. V. Adorno, J. Y. Ishihara, and G. Borges, "Switching strategy for flexible task execution using the cooperative dual task-space framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2014, pp. 1703–1709.

[62] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, oct 2019.

[63] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Taipei, Oct. 2010.

[64] Q. Gongye, P. Cheng, and J. Dong, "Image-based visual servoing with depth estimation," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 9, pp. 1811–1823, 2022.

[65] A. Chakravarthy and D. Ghose, "Collision cones for quadric surfaces," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1159–1166, 2011.

[66] Y.-K. Choi, J.-W. Chang, W. Wang, M.-S. Kim, and G. Elber, "Continuous collision detection for ellipsoids," *IEEE Transactions on visualization and Computer Graphics*, vol. 15, no. 2, pp. 311–325, 2008.

[67] H. C. Verma, *Concepts of Physics, Part 2*. Bharti Bhavan, 2011.

[68] A. Ataka, H.-K. Lam, and K. Althoefer, "Reactive magnetic-field-inspired navigation method for robots in unknown convex 3-d environments," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3583–3590, 2018.

[69] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[70] B. V. Adorno, "Two-arm manipulation: From manipulators to enhanced human-robot collaboration," Ph.D. dissertation, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) - Université Montpellier 2, Montpellier, France, 2011.
[71] B. Adorno, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra – Part I: Fundamentals," *HAL Archives*, p. 47, 2017.
[72] L. F. C. Figueredo, "Kinematic control based on dual quaternion algebra and its application to robot manipulators," Ph.D. dissertation, University of Brasilia, Brazil, 2016.
[73] J. J. Quiroz-Omaña and B. V. Adorno, "Whole-body control with (self) collision avoidance using vector field inequalities," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4048–4053, 2019.
[74] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers Robotics AI*, vol. 3, 2016.
[75] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming," *International Journal of Robotics Research*, 2014.
[76] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.
[77] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, 2009.
[78] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.
[79] B. V. Adorno and M. Marques Marinho, "Dq robotics: A library for robot modeling and control," *IEEE Robot. Autom. Mag.*, 2020.
[80] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke *et al.*, "ros_control: A generic and simple control framework for ros," *The Journal of Open Source Software*, 2017.
[81] R. Laha, M. Becker, J. Vorndamme, J. Vrabel, L. F. Figueredo, M. A. Müller, and S. Haddadin, "Dataset: Predictive multi-agent based planning and landing controller for reactive dual-arm manipulation." 2023. [Online]. Available: https://doi.org/10.25835/7mvvk9qg
[82] G. Bradski, "The opencv library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.
[83] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.
[84] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.
[85] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, 2011.
[86] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming," *International Journal of Robotics Research*, 2012.
[87] H. Yao, R. Laha, L. F. Figueredo, and S. Haddadin, "Enhanced dexterity maps (edm): A new map for manipulator capability analysis," *IEEE Robot. Autom. Lett.*, 2024.
[88] E. Study, "Von den bewegungen und umlegungen," *Springer*, vol. 39, no. 4, pp. 441–565, 1891.
[89] J. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 1999.
[90] J. Groß, G. Trenkler, and S.-O. Troschke, "Quaternions: further contributions to a matrix oriented approach," *Linear Algebra and its Applications*, vol. 326, pp. 205–213, 2001.

Riddhiman Laha received his B.Tech. degree (2015) in Mechanical Engineering from the Kalinga Institute of Industrial Technology (KIIT) in Bhubaneswar, India, and earned his M.S. degree (2018), also in Mechanical Engineering, from the State University of New York (SUNY) at Stony Brook, USA. He is currently working toward his Ph.D. degree in Electrical Engineering and Computer Science at the Technical University of Munich (TUM) in Germany, where he is also affiliated with the Munich Institute of Robotics and Machine Intelligence (MIRMI).

Marvin Becker received his Master's degree in Mechanical engineering from the Technical University Munich, Germany, in 2016. He is currently working toward his Ph.D. degree in Electrical Engineering at the Institute of Automatic Control (IRT) at Leibniz University Hannover. His current research interests are in the area of motion planning and collision avoidance for robotic manipulators.

Jonathan Vorndamme received his M.Sc. degree in mechatronics in 2012 and his B.Sc. degree in mathematics in 2014 from the Gottfried Wilhelm Leibniz Universität Hannover. He started working towards a Ph.D. at the Gottfried Wilhelm Leibniz Universität Hannover in 2012 and is currently continuing this endeavor at the Technical University of Munich (TUM), where he is affiliated with the Munich School of Robotics and Machine Intelligence. His current research interests are multi-contact detection and estimation in complex robotic systems as well as robotic reflexes.

Juraj Vrabel received his B.Sc. degree in Physics from Charles University in Prague, Czech Republic, in 2014 and his M.Sc. degree in Theoretical and Mathematical Physics from Ludwig Maximilian University and the Technical University of Munich, Germany, in 2017. He then pursued various engineering and research roles, including Robotics Researcher at the Technical University of Munich, Robotics Software Engineer at Agile Robots AG, and Senior Software Engineer at Numfum GmbH.

Luis F.C. Figueredo received his Bachelor's and Master's degrees in Electrical Engineering from the University of Brasilia, Brazil and earned his Ph.D. degree in Robotics also from the University of Brasilia. He is currently an Assistant Professor at the University of Nottingham, United Kingdom. He has received multiple awards including the Best Ph.D. Thesis at the University of Brasilia and IEEE ICRA 2022 Conference Editorial Board Outstanding Reviewer Award.

Matthias A. Müller received a Diploma degree in Engineering Cybernetics from the University of Stuttgart, Germany, an M.Sc. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign, US (both in 2009), and a Ph.D. from the University of Stuttgart in 2014. Since 2019, he is director of the Institute of Automatic Control and full professor at the Leibniz University Hannover, Germany. His research interests include nonlinear control and estimation, model predictive control, and data- and learning-based control, with application in different fields. He has received various awards for his work, including the inaugural Brockett-Willems Outstanding Paper Award, and the IEEE CSS George S. Axelby Outstanding Paper Award 2022.

Sami Haddadin received the Dipl.-Ing. degree in Electrical engineering in 2005, the M.Sc. degree in Computer Science in 2009 from the Technical University of Munich (TUM), Munich, Germany, the Honours degree in Technology Management in 2007 from Ludwig Maximilian University, Munich, Germany, and TUM, and the Ph.D. degree in safety in robotics from RWTH Aachen University, Aachen, Germany, in 2011. He is currently a full professor and chair of Robotics and Systems Intelligence at the Technical University of Munich (TUM) and the founding di- rector of the Munich Institute of Robotics and Machine Intelligence (MIRMI). He has received numerous awards for his scientific work, including the George Giralt Ph.D. Award (2012), the German President's Award for Innovation in Science and Technology (2017), and the Leibniz Prize (2019).