

Globally-Guided Geometric Fabrics for Reactive Mobile Manipulation in Dynamic Environments

Tomas Merva , Graduate Student Member, IEEE, Saray Bakker , Graduate Student Member, IEEE, Max Spahn , Danning Zhao , Ivan Virgala , Member, IEEE, and Javier Alonso-Mora , Senior Member, IEEE

Abstract—Mobile manipulators operating in dynamic environments shared with humans and robots must adapt in real time to environmental changes to complete their tasks effectively. While global planning methods are effective at considering the full task scope, they lack the computational efficiency required for reactive adaptation. In contrast, local planning approaches can be executed online but are limited by their inability to account for the full task’s duration. To tackle this, we propose Globally-Guided Geometric Fabrics (G3F), a framework for real-time motion generation along the full task horizon, by interleaving an optimization-based planner with a fast reactive geometric motion planner, called Geometric Fabrics (GF). The approach adapts the path and explores a multitude of acceptable target poses, while accounting for collision avoidance and the robot’s physical constraints. This results in a real-time adaptive framework considering whole-body motions, where a robot operates in close proximity to other robots and humans. We validate our approach through various simulations and real-world experiments on mobile manipulators in multi-agent settings, achieving improved success rates compared to vanilla GF, Prioritized Rollout Fabrics and Model Predictive Control.

Index Terms—Mobile manipulation, constrained motion planning, geometric fabrics, collision avoidance.

I. INTRODUCTION

THE deployment of mobile manipulators in human-centered environments, including households and hospitals, relies on their ability to carry out object-focused tasks like navigation, grasping, transportation and placement. These long-horizon

Received 20 December 2024; accepted 11 April 2025. Date of publication 16 April 2025; date of current version 25 April 2025. This article was recommended for publication by Associate Editor K. Hang and Editor M. Vincze upon evaluation of the reviewers’ comments. This work was supported in part by European Union through ERC, INTERACT, under Grant 101041863, and in part by Slovak Grant Agency VEGA under Grant 1/0436/22, Grant 008TUKE-4/2024, and Grant 07/TUKE/2024. (Tomas Merva and Saray Bakker contributed equally to this work.) (Corresponding authors: Tomas Merva; Saray Bakker.)

Tomas Merva is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the Department of Industrial Automation and Mechatronics, Technical University of Kosice, 040 01 Kosice, Slovakia (e-mail: tomas.merva@tuke.sk).

Saray Bakker, Max Spahn, Danning Zhao, and Javier Alonso-Mora are with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: s.bakker-7@tudelft.nl; m.spahn@tudelft.nl; d.zhao-3@student.tudelft.nl; j.alonsomora@tudelft.nl).

Ivan Virgala is with the Department of Industrial Automation and Mechatronics, Technical University of Kosice, 040 01 Kosice, Slovakia (e-mail: ivan.virgala@tuke.sk).

Project website: https://autonomousrobots.nl/paper_websites/g3f

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3562005>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3562005

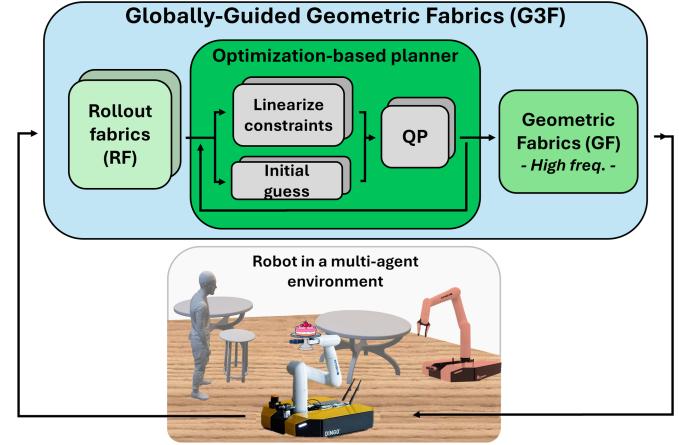


Fig. 1. A schematic overview of G3F. Given the current state of the environment, RF provides two distinct forward predictions for linearization and warm-starting the quadratic program (QP) solver. After refining the solution of the optimization-based planner iteratively, the path with the lower cost guides the geometric motion planner GF.

tasks can span several tenths of seconds or involve sequential processes, e.g. pick-and-place, where global guidance is essential for smooth execution. In dynamic environments, offline global planning alone is insufficient as humans and other robots might obstruct the predefined paths or end-effector target poses [1], while online local planning methods are limited to short-term planning and overlook the complete task horizon. This highlights a need for global methods capable of adapting online to the movements of other agents. Moreover, if other agents or obstacles block a precomputed target pose, the planner should identify an alternative end-effector pose to accomplish the task online. By addressing the global planning problem and object-centric tasks simultaneously, efficient solutions can be enabled for cluttered and multi-agent settings.

The most common way to adapt to environmental changes is through repeated replanning, often using optimization-based frameworks such as Model Predictive Control (MPC), e.g. [2], [3], [4]. However, the required computation for long-horizon tasks is a major bottleneck for high-dimensional systems in dynamic environments [5]. Data-based methods represent an attractive alternative since they can drastically reduce computational time. Nonetheless, these approaches require a lengthy process of data collection and subsequent training, while often lacking guarantees to satisfy all physical constraints [6]. Another

class of methods for fast replanning is built on the concept of geometry control, such as Geometric Fabrics (GF) [7] encoding the desired behavior of the system, including joint limits and collision avoidance, into geometries represented by second-order differential equations. As GF are inherently reactive, e.g. have no prediction horizon, Prioritized Rollout Fabrics (P-RF) were proposed in [8] to forward simulate multi-agent systems via Rollout Fabrics (RF) and resolve deadlocks based on prioritization.

Most online planning methods for object-focused tasks rely on a high-level planner providing global guidance toward a static target pose, such as a pick or place pose. However, in dynamic environments, a static target pose may become unreachable. By exploring alternative end-effector target poses online, mobile manipulators can adapt and complete their tasks even if the pre-computed pose is obstructed. Nevertheless, as discussed in Section II, most methods that investigate alternative pick-and-place poses either neglect collision avoidance or are computationally inefficient, making them unsuitable for reactive execution.

We propose Globally-Guided Geometric Fabrics (G3F), a framework for real-time motion generation achieving reactivity in dynamic environments while considering the full task horizon from start to goal. We approximate the trajectory optimization problem as a QP by leveraging RF for linearization of the constraints and providing an initial guess to the solver (Fig. 1). This allows us to consider the time horizon until completion of the task, whereas online optimization-based methods usually limit the time horizon up to a few seconds. To explore the multitude of acceptable target poses for task completion, we explicitly model the task's degree of freedom into the QP. By coupling the optimization-based planner with a lower-level motion generator, we achieve real-time performance that accounts for the whole-body kinematics of mobile manipulators while avoiding dynamic obstacles. The effectiveness of our approach is verified in several simulated and real-world multi-agent scenarios.

To this end, our contributions are as follows:

- Globally-Guided Geometric Fabrics (G3F): A real-time motion generation approach combining an optimization-based planner with Rollout Fabrics for adaptive motion planning in dynamic environments.
- Object-informed planning: G3F enables the robot to explore alternative target poses and account for collision avoidance and physical constraints, ensuring efficient and reactive execution of object-centric tasks.
- We experimentally evaluate the approach in simulated and real-world multi-agent scenarios, using qualitative examples and a quantitative comparison against GF, P-RF and MPC.

II. RELATED WORK

A. Motion Planning for Mobile Manipulators

Common approaches to motion planning for high-dimensional systems, such as mobile manipulators, are sampling-based and optimization-based methods. Sampling-based planners, including probabilistic road maps (PRM) [9], [10] and

variations of rapidly exploring random trees (RRT) [11], [12], are however primarily used for static environments where a plan is generated offline and then executed, as their computational complexity poses challenges for reactive execution [1]. In contrast, optimization-based motion planners incorporate predictions of other agents' behavior, including them as constraints within the optimization program. Model Predictive Control (MPC) solves the optimization problem over a receding horizon while replanning the solution based on the current configuration and environment. Whole-body MPC has been applied to a single mobile manipulator in several real-world settings, e.g. [2], [3], [5], [13]. Another strategy is to separate base and manipulator motion planning, where the base is moved during navigation and for a manipulation task only the arm is moved [14]. This simplifies planning but could lead to inefficient task execution, e.g. when the base cannot be adjusted to achieve an alternative end-effector pose. Therefore, this work focuses on whole-body motion generation leveraging all degrees of freedom (DoFs) for efficient navigation and manipulation in obstacle-rich dynamic environments.

Geometric motion planning offers a fast alternative to optimization-based motion planners, ensuring stable, convergent behavior for kinematically redundant robots via differential geometry [15]. Geometries shape a manifold of the configuration space in such a way that collision avoidance, limit avoidance and goal-reaching are achieved in a smooth manner. The recently introduced Riemannian Motion Policies are suitable for designing human-like motions [16], and are extended to Geometric Fabrics (GF) in [17]. These fabrics provide convergence to local minima using straightforward construction rules [7], [18] and can adapt to dynamic environments [19], while having a high planning frequency compared to optimization-based methods [19]. GF have been further expanded for multi-agent systems, introducing P-RF, resolving deadlocks based on prioritization by leveraging RF to forward simulate the multi-robot system. In RF, actions are computed via GF at every predicted step along the horizon given the current state and predicted goal of each agent [8]. More global solutions are provided in [20] by using fabric rollouts as a set of candidates for a sampling-based optimizer along with a learned warm-start. In contrast, our approach leverages fabric rollouts to warm-start the QP solver, enabling online trajectory planning for mobile manipulators in dynamic environments while considering a longer time horizon and being less computationally intensive.

B. Sequential Motion Planning

Sequential object-centric tasks, such as pick-and-place, are explored in Task and Motion Planning (TAMP) [21], [22], examining the intertwined relationship between long-horizon sequences of discrete actions and continuous motion parameters. However, the computational time required makes these methods often unsuitable for real-time adaptive replanning. For the reactive execution of TAMP, [23] formulates the problem using relative Cartesian coordinates alongside operational space controllers; however, it cannot handle situations where the original target pose is unreachable. SEC-MPC decomposes the control

of the sequential plan into three sub-problems using MPC [24], while we address the problem by leveraging geometry control. Even though the discussed TAMP frameworks exhibit enhanced task variability, our method contributes to TAMP research by enabling online generation of long-horizon motions for sequential pick-and-place tasks, where intermediate goals are reached.

C. Object-Centric Planning

Recent advances have shown that data-driven models are capable of determining feasible object-centric poses, such as grasp candidates, for several objects [25]. By processing sensory inputs from RGB-D cameras, data-driven models provide grasp candidates for the given objects, usually in the form of a grasp axis [26], [27], [28]. However, most methodologies are focused on determining the object-centric pose without considering the full environment, thereby overlooking blocking obstacles and other agents. This issue is especially common for mobile manipulators operating in multi-agent dynamic environments. The GOMP planner [29] exploits the grasp axis by creating an additional DoF around it to compute pick-and-place poses and trajectories that enable faster pick-and-place cycle time, considering collision avoidance on the height of the end-effector. To speed up the computation time, [6] trained a neural network (NN) on data from the GOMP-based planner to warm-start the optimization process. Our work shares similarities with [6], [29] in exploiting the target pose axis and incorporating it into a QP, but we achieve run-time performance incorporating whole-body collision avoidance in dynamic multi-agent scenarios, without relying on data-collection and training, by leveraging geometric fabrics.

III. PRELIMINARIES

In this section, we provide an introduction to geometric fabrics (Section III-B), a fundamental concept of our method, starting with the required notations (Section III-A).

A. Notation

A robot has a configuration \mathbf{q} in the configuration space \mathcal{C} with time-derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$. In the context of geometric control, *tasks* can be defined in different spaces, e.g. defining a task variable \mathbf{x} as the difference between the goal position and the current end-effector position. A task variable $\mathbf{x}_j \in \mathcal{X}_j$ is defined with dimension $m_j \leq n$, $\forall j \in [M]$ where M indicates the number of task spaces and $[M]$ denotes the shorthanded notation of the set $\{j \in \mathbb{Z}^+ : j \leq M\}$. For task variables indicating a pose, the notation ζ is used, consisting of a position $\mathbf{p} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^\top$ and orientation expressed as the rotation matrix \mathbf{R} , forming the transformation matrix \mathbf{T} . A twice-differential map $\phi_j : \mathcal{C} \rightarrow \mathcal{X}_j$ connects the configurations \mathbf{q} of the robot with the task variables \mathbf{x}_j . The map ϕ_j relating the configuration space to the end-effector pose ζ_{ee} is denoted as ϕ_{ee} provided by the forward kinematics (FK).

B. Geometric Fabrics

Geometric fabrics (GF) define the desired behavior of a system using second-order differential equations of the form $\ddot{\mathbf{x}} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})$ [7], [17]. These artificial dynamical systems, notated by their corresponding equations of motion $\mathbf{M}(\mathbf{x}, \dot{\mathbf{x}})\ddot{\mathbf{x}} + \boldsymbol{\xi}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{0}$, represent a task such as avoiding a collision between the end-effector of the robot and an obstacle. To ensure that trajectories generated by the dynamical systems are converging when forced, the system $\ddot{\mathbf{x}} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})$ is *energized* using a Finsler energy \mathcal{L}_e with the corresponding equations of motion $\mathbf{M}_{\mathcal{L}_e}\ddot{\mathbf{x}} + \boldsymbol{\xi}_{\mathcal{L}_e} = \mathbf{0}$,

$$\ddot{\mathbf{x}} = \text{energize}_{\mathcal{L}_e}[\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})] = \mathbf{h} - \frac{\dot{\mathbf{x}}^\top (\mathbf{M}_{\mathcal{L}_e}\mathbf{h} + \boldsymbol{\xi}_{\mathcal{L}_e})}{\dot{\mathbf{x}}^\top \mathbf{M}_{\mathcal{L}_e} \dot{\mathbf{x}}} \dot{\mathbf{x}} \quad (1)$$

The resulting system $\ddot{\mathbf{x}} = \tilde{\mathbf{h}}(\mathbf{x}, \dot{\mathbf{x}}) = \tilde{\mathbf{M}}^{-1}\tilde{\boldsymbol{\xi}}$ then forms a geometric fabric [17]. To ensure path consistency, e.g. energization only changes the speed along the path but not the path itself, the function $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})$ is designed to be homogeneous of order 2, $\mathbf{h}(\mathbf{x}, \gamma\dot{\mathbf{x}}) = \gamma^2\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})$, $\forall \gamma \geq 0$.

To combine all behaviors, the dynamical systems defined in their respective task space \mathcal{X} are pulled to the configuration space \mathcal{C} and summed. The *pullback operation* maps the energy-conserving fabric to the configuration space using a twice-differential map ϕ [7],

$$\text{pull}_\phi \left(\tilde{\mathbf{M}}, \tilde{\boldsymbol{\xi}} \right)_\mathcal{X} = \left(\mathbf{J}_\phi^T \tilde{\mathbf{M}} \mathbf{J}_\phi, \mathbf{J}_\phi^T (\tilde{\boldsymbol{\xi}} + \dot{\mathbf{J}}_\phi \dot{\mathbf{q}}) \right)_\mathcal{C}. \quad (2)$$

The resulting energy-conserving fabric can be forced by a navigation policy \mathbf{f} to the minimum of a potential function $\psi(\mathbf{q})$ when damped,

$$\ddot{\mathbf{q}} = \tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}). \quad (3)$$

Equation (3) is therefore a combination of all avoidance behaviors, e.g. collision avoidance and limit avoidance, defined as energy-conserving fabrics $\tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$ and a policy forcing the system to a desired goal, $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$.

IV. GLOBALLY-GUIDED GEOMETRIC FABRICS (G3F)

Geometric fabrics, being a local motion planning method, require global guidance to avoid local minima like deadlocks. Our proposed method, G3F, provides this guidance via a real-time adaptive reference path. Section IV-A introduces the nonlinear program (NLP) problem formulation, followed by G3F achieving real-time guidance by simplifying the NLP into a QP via RF, allowing for online target pose adaptations.

A. Problem Formulation

Consider a mobile manipulator that is required to complete its task, e.g. grasp a cup, while remaining collision-free in a dynamic environment. This can be expressed as an NLP formulation that minimizes the control objective $f(\tau)$ along the trajectory of length T , represented by a set of the robot's joint-space waypoints $\tau = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_T]^\top$, while satisfying

constraints,

$$\min_{\tau} f(\tau), \quad (4a)$$

$$\text{s.t. } \phi_{ee}(\mathbf{q}_k) \in \mathcal{P}, \quad (4b)$$

$$g(\mathbf{q}_t) \leq 0, \quad \forall t \in [0, T]. \quad (4c)$$

The constraints encapsulate that the robot's end-effector pose, obtained via forward kinematics $\phi_{ee}(\mathbf{q}_k)$, at waypoint \mathbf{q}_k , lies within the interval of feasible end-effector target poses \mathcal{P} , where $k = T$ for constraining the final waypoint in (4b). The inequality constraints in (4c) ensure joint limits and collision avoidance with obstacles and other agents along the discrete path τ are considered.

Solving the full NLP in (4) over the entire trajectory horizon is computationally costly, especially for high DoF-systems. We propose G3F for online trajectory generation by approximating the NLP as an informed QP by using RF.

B. Optimization-Based Planner

Reformulating an NLP into a QP, requires careful considerations to preserve the global nature of the solution. Therefore, we propose to leverage GF as a fast collision-free local motion planner, as well as its extension RF to provide an initial guess for the QP and joint-space waypoints for constraint linearization (Fig. 1). By using GF as the local motion planner, the QP optimizes a sparser set of waypoints and reduce the number of constraints to the core collision links of the robot. Our approach will jointly optimize the trajectory towards the target pose and the target pose itself, allowing exploration of alternative poses in complex environments. The resulting QP serves as an effective, sparse planner along the full task horizon that guides the low-level motion planner GF. The QP is solved sequentially at every iteration $h \in [1, \dots, H]$ with H the number of iterations, to refine the initial guess and linearized constraints,

$$\min_{\tau_h} f_{QP}(\tau_h), \quad (5a)$$

$$\text{s.t. } \mathbf{lb}_h \leq \mathbf{A}_h \mathbf{q}_t \leq \mathbf{ub}_h, \quad \forall \mathbf{q}_t \in \tau_h, \quad (5b)$$

where the constraints \mathbf{lb}_h , \mathbf{A}_h , and \mathbf{ub}_h are linearized using the fixed points determined by the solution at the previous iteration τ_{h-1} , where τ_0 is given by RF.

1) *Objective Function:* We formulate the objective function of the QP in convex quadratic form penalizing the sum of squared accelerations along the trajectory, as well as minimizing the difference between decision variables τ and the initial guess τ_0 ,

$$f_{QP}(\tau) = \tau^\top \mathbf{Q} \tau + (\tau - \tau_0)^\top (\tau - \tau_0), \quad (6)$$

where \mathbf{Q} is a positive semi-definite matrix expressed as $\mathbf{Q} = \mathbf{A}^\top \mathbf{A}$, with \mathbf{A} being a finite-difference matrix used to compute joint accelerations $\dot{\mathbf{q}}_t = \mathbf{A}_t \mathbf{q}_t$ from joint positions $\mathbf{q}_t \in \tau$. The vector τ_0 represents the initial guess provided by RF. The quadratic penalty $(\tau - \tau_0)^\top (\tau - \tau_0)$ minimizes the deviation from the initial guess τ_0 . By favoring a solution close to the initial guess, this approach enhances consistency and prevents the QP solver from generating large variations in consecutive iterations. The objective function enforces smooth motions to

achieve feasible trajectories, with the joint limits for the first waypoint \mathbf{q}_0 set to the current state of the robot.

2) *Target-Pose Adaptation:* Since the target pose might be unreachable due to the manipulation task conflicting with collision avoidance, we explore alternative target poses by modeling the task's degrees of freedom (DOF) into the QP explicitly. We assume that a high-level subsystem produces a precomputed target pose candidate with respect to the world frame W , denoted as ${}^W \mathbf{T}_g$. To incorporate a multitude of acceptable target poses into (5b), the target pose constraint in (4b) is formulated as separate inequality constraints for position and orientation. The end-effector position constraint ensures that the end-effector position remains within a bounding sphere, $\|\mathbf{p}_{ee} - \mathbf{p}_g\|_2 \leq \alpha_g$, where α_g represents the task-dependent positional tolerance.

The orientation constraint is expressed as a set of rotational constraints on only two axes of the end-effector frame, since constraining two axes restricts the third axis as well. For illustration, we explore the scenario where the z-axis of the desired end-effector frame should align with the z-axis of the target frame, while rotations around the x-axis are allowed. The first rotational constraint enforces the difference in rotational angle θ_z along the z-axis, between the end-effector frame \mathbf{T}_{ee} and target frame ${}^W \mathbf{T}_g$, to be zero with some small tolerance α_z ,

$$\cos(\alpha_z) \leq (\bar{\nu}_{ee}^z)^\top \bar{\nu}_g^z, \quad (7)$$

where $\bar{\nu}_{ee}^z$ and $\bar{\nu}_g^z$ are the normalized rotational z-axis column-vectors contained in the transformation matrices \mathbf{T}_{ee} and ${}^W \mathbf{T}_g$ respectively.

The second rotational constraint defines the additional DoF as a relative rotation θ around the x-axis with respect to the original target pose, constrained within a specified interval $\pm \theta_{max}$,

$$\cos(\theta_{max}) \leq (\bar{\nu}_{ee}^x)^\top \bar{\nu}_g^x, \quad (8)$$

where $\bar{\nu}_{ee}^x$ and $\bar{\nu}_g^x$ are the normalized rotational x-axis column-vectors extracted from \mathbf{T}_{ee} and ${}^W \mathbf{T}_g$ respectively. These constraints are linear as the forward kinematics relating the waypoint $q_i \in \tau$ towards \mathbf{T}_{ee} , are linearized using the previous solution of τ . The combination of (7) and (8) allows for alternative end-effector poses on selected waypoints of τ , e.g. on the final waypoint at $t = T$.

3) *Collision Avoidance:* By integrating collision avoidance in the long-horizon planner, we can ensure that the path is collision-free if a solution is found and we can reactively adapt to the movements of the other agents. As the fast local motion planner GF includes whole-body collision avoidance, the collision-avoidance constraints in the QP can consider a sparser set of links on the ego-agent. Collision avoidance is represented as inequality constraints, with these links and obstacles approximated as a set of spheres, expressed as follows for one collision pair:

$$\|\mathbf{p}_{link} - \mathbf{p}_{obst}\|_2 \geq r_{link} + r_{obst}, \quad (9)$$

where \mathbf{p}_{link} and \mathbf{p}_{obst} are the center position of spheres of the robot's links and obstacles respectively and r_{link} and r_{obst} their radii [30]. To compensate for the reduced number of collision constraints, we use larger radii than for GF.

C. Rollout Fabrics

We employ RF to initialize the aforementioned QP, as RF provides collision-free initial guesses for the QP and configuration-space paths along which the constraints are linearized. RF provide forward predictions of the robot's movements where actions at each prediction step are computed efficiently via GF, similar to [8]. Unlike [8], we predict the ego-robot's state-action pairs over a horizon K , assuming a quasi-static environment at each time-step t , thereby allowing for extended prediction horizons. To avoid forward simulating complex dynamics, a second-order integrator is used as the joint-space prediction model, with a low-pass filter applied to the velocity, $\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \Delta t(\alpha_v \ddot{\mathbf{q}}_k + (1 - \alpha_v) \dot{\mathbf{q}}_k)$, $\forall k \in K$, with time-step Δt and constant α_v . After each timestep, we compute the Euclidean norm between the predicted end-effector position, obtained via $\phi_{ee}(\mathbf{q}_{k+1})$ and the target position to stop the rollout early if the distance to the target pose falls within the tolerance.

As mentioned in Section III-B, a fabric consists of a combination of several tasks, shaped via second-order differential equations. In the following, detailed design guidelines are provided for each component within GF and RF.

1) *Repulsion Dynamics:* For collision avoidance and joint limit avoidance, the dynamical system repels the robot's body from the undesired region. A possible dynamical system for a repelling motion is of the form, $\ddot{\mathbf{x}} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}) = -\frac{\alpha_0}{\mathbf{x}^z} \dot{\mathbf{x}}^2$, where $\dot{\mathbf{x}}^2$ and $(\frac{\alpha_0}{\mathbf{x}^z})$ are element-wise operations, and the barrier $(\frac{\alpha_0}{\mathbf{x}^z})$ with order $z \in [1, 2, \dots]$ enforces repulsion with task variable \mathbf{x} expressing the distance between an undesired region and a position along the robot's body. The system $\ddot{\mathbf{x}} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}})$ is homogeneous of order 2 in $\dot{\mathbf{x}}$.

2) *Energization:* A geometric fabric $\tilde{\mathbf{h}}$ conserves a Finsler energy, and therefore all repulsive terms are energized as in (1). A limit or collision Finsler that adheres to these rules is $\mathcal{L}_e = \frac{\alpha_0}{\mathbf{x}^z} \dot{\mathbf{x}}^2$ where $z \in [2, 4, \dots]$. The combined energy-conservative fabrics $\tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$ are formed by pulling all task-specific fabrics using the pullback operation in (2) and summing them in configuration space \mathcal{C} .

3) *Attractor Dynamics:* To ensure convergence towards a goal, the energy-conservative fabric is forced towards the minimum of a potential function [7], $\psi(\mathbf{x})$, e.g.

$$\psi(\mathbf{x}) = \alpha_0 \left(\|\mathbf{x}\| + \frac{1}{\alpha_1} \ln \left(1 + e^{-2\alpha_1 \|\mathbf{x}\|} \right) \right) \quad (10)$$

where a straightforward choice for the task variable \mathbf{x} is the difference between the end-effector pose of the robot ζ_{ee} and the goal pose ζ_g . The navigation policy \mathbf{f} is formed by the derivative of the potential function and additional damping $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$ with $\mathbf{B} > 0$, e.g. $\mathbf{f} = \tilde{\mathbf{M}}^{-1}(-\partial\psi - \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}})$, which ensures convergence to the local minimum of ψ under the assumption that $\tilde{\mathbf{M}}^{-1}$ is a positive definite matrix [17].

4) *Speed-Control Via Execution Energy Regulation:* The speed profile of the energized geometry is determined by the Finsler energy \mathcal{L}_e . Often, it is desired to regulate a so-called *execution energy* \mathcal{L}_{ex} . The forced fabric in (3) can therefore be altered by an additional term α_{ex} ,

$$\ddot{\mathbf{q}} = \tilde{\mathbf{h}} + \mathbf{f} + \alpha_{ex} \text{ where } \alpha_{ex} = \eta \alpha_{ex}^0 + (1 - \eta) \alpha_{ex}^\psi \quad (11)$$

Algorithm 1: Globally-Guided Geometric Fabrics.

```

Input :  $(\mathbf{q}, \dot{\mathbf{q}})$ ,  ${}^W\mathbf{T}_{obsts}$ ,  ${}^W\mathbf{T}_g$   $\triangleright$  Current state of ego-robot
           $\tau_{RF, coll}$ ,  $\tau_{RF, free}$   $\triangleright$  Obstacles' and target poses
1  $\tau_{result} \leftarrow \emptyset$ 
2  $\tau_{RF, coll}, \tau_{RF, free} \leftarrow$  compute RF
3 for  $\tau_0 \in \{\tau_{RF, coll}, \tau_{RF, free}\}$  do
4   for  $h = 1, \dots, H$  do
5      $\mathbf{A}_h, \mathbf{l}_h, \mathbf{u}_h \leftarrow$  linearized using  $\tau_{h-1}$ 
6      $\tau_h \leftarrow$  QP, as in (5)
7      $\tau_{result} \leftarrow \tau_{result} \cup \{\tau_H\}$ 
8    $\{\tau_{coll}, \tau_{free}\} \leftarrow \tau_{result}$ 
9   if  $f_{QP}(\tau_{coll}) \leq f_{QP}(\tau_{free})$  then
10    |  $\Xi_{ee} \leftarrow \phi_{ee}(\tau_{coll})$ 
11   else
12    |  $\Xi_{ee} \leftarrow \phi_{ee}(\tau_{free})$ 
13    $\zeta_g =$  reference_tracker( $\Xi_{ee}$ )
14    $\ddot{\mathbf{q}} \leftarrow \tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \zeta_g)$ ,  $\triangleright$  GFs, (3)

```

where α_{ex}^0 and α_{ex}^ψ are the energization coefficients to maintain a constant execution energy \mathcal{L}_{ex} under a zero potential $\psi = 0$ and when forced $\psi > 0$ respectively, and η is a constant between 0 and 1. For more details on energy regulation, we refer to [7], [18].

D. G3F in Multi-Agent Scenarios

In Sections IV-B and IV-C, we describe G3F, a framework for real-time motion generation including target-pose optimization, solving a QP sequentially informed by RF. Algorithm 1 summarizes G3F for a multi-agent scenario, where every robot is controlled in a decentralized manner, observing the other agents as a set of collision sphere poses, ${}^W\mathbf{T}_{obsts}$, and radii, where the poses are updated at every time step. RF provides a collision-free initial guess for the optimization-based planner, $\tau_{RF, coll}$. In some cases, achieving a collision-free trajectory via RF can be challenging due to the local nature of fabrics, and RF might be biased towards a locally optimal plan. Therefore, the optimization-based planner is initialized with two types of joint-space trajectories for τ_0 in (5): 1) Trajectories by RF respecting collision avoidance, $\tau_{RF, coll}$, 2) trajectories generated by RF without collision avoidance, while still respecting kinematics and joint limits, $\tau_{RF, free}$. Using GF for initialization improves task success and the QP solver performance over linear interpolation or the zero vector. We solve the QP sequentially with these two different initializations for $\tau_0 \in \{\tau_{RF, coll}, \tau_{RF, free}\}$, resulting in two different sequences, τ_{coll} and τ_{free} (Line 4-7). The sequence with the lowest objective cost, provided by (6), is mapped to its corresponding sequence of end-effector poses Ξ_{ee} via the forward kinematics ϕ_{ee} (Line 11, 12). Via a reference tracker (Line 13), a local goal in the form of an end-effector pose, $\zeta_{ee} \in \Xi_{ee}$, is provided to the low-level motion planner, i.e. GF (Line 14). When the QP cannot find a feasible solution, the final pose from the last feasible solution of the QP is supplied to GF.

TABLE I
STATISTICS FOR FOUR SIMULATED MULTI-AGENT SCENARIOS COMPARING THE PROPOSED G3F AGAINST GF, P-RF, AND MPC

Scenario	Method	Success rate [%]	Time-to-Success [s]	Collision rate [%]	Computational time [ms]	
					Low level	High level
Scenario 1: Two tables	G3F (ours)	100	21.34 \pm 3.47	0	1.15 \pm 0.32	329.1 \pm 128.2
	GF	70	20.96 \pm 6.41	0	1.14 \pm 0.42	—
	P-RF	70	20.96 \pm 6.41	0	1.04 \pm 0.32	128.5 \pm 152.6
	MPC	45	10.08 \pm 0.73	0	14.51 \pm 1.33	—
Scenario 2: One table	G3F (ours)	85	32.79 \pm 5.92	0	1.09 \pm 0.42	296.2 \pm 119.1
	GF	35	25.50 \pm 3.83	0	1.07 \pm 0.37	—
	P-RF	35	25.40 \pm 4.14	0	0.95 \pm 0.25	176.7 \pm 143.9
	MPC	65	14.98 \pm 9.15	0	13.90 \pm 1.26	—
Scenario 3: Single agent	G3F (ours)	100	24.25 \pm 4.27	0	1.18 \pm 0.43	315.3 \pm 117.6
	GF	45	31.54 \pm 8.38	0	1.17 \pm 0.39	—
	MPC	35	19.24 \pm 7.46	0	14.30 \pm 0.75	—
Scenario 4: Three agents	G3F (ours)	80	35.53 \pm 7.64	0	1.12 \pm 0.37	361.4 \pm 157.9
	GF	30	37.92 \pm 7.59	10	1.07 \pm 0.33	—
	P-RF	30	38.68 \pm 6.65	10	0.98 \pm 0.29	274.0 \pm 297.4
	MPC	20	10.68 \pm 0.77	15	34.69 \pm 3.93	—

Each scenario consists of 20 randomized environments with varying target poses, obstacle poses and initial configurations.

V. EXPERIMENTS

A. Experimental Setup

The presented framework is evaluated in both simulation and on real hardware. In each setting, we use a 9-DoF mobile manipulator consisting of a Clearpath Dingo holonomic mobile base and a Kinova Gen3 lite 6-DoF manipulator. The optimization-based planner is solved using the QP solver OSQP [31] in conjunction with the symbolic framework CasADi [32] without parallelization or use of a GPU. Due to the complexity of the simulated environments, the long-horizon planner, including RF and the optimization-based planner, were configured to operate at a frequency of 2 Hz and GF at 100 Hz. In real-world experiments, the long-horizon planner operates at 4 Hz and GF at 25 Hz, primarily constrained by communication limitations with the robotic hardware. The computation time of RF is influenced by the time-step $\Delta t = 0.05\text{s}$, the maximum horizon length $K = 500$, goal weighting of RF and a distance to a target pose. The simulation experiments run on a laptop with an Intel Core i7 and 16 GB RAM, whereas in real-world experiments, all is executed on the onboard NVIDIA AGX Orin of the robots. To detect the poses of static obstacles, humans, robots and target objects, we employ the Vicon motion capture system. More details and videos can be found on our website.

B. Simulation Experiments

In the simulation experiments, we compare our method G3F across four realistic scenarios against three state-of-the-art methods, namely vanilla GF [17], P-RF [8], and a MPC for whole-body control of a mobile manipulator [2] which we extend for a decentralized multi-agent scenario with a horizon of 2s and $\Delta t_{\text{MPC}} = 0.1\text{s}$. Each scenario is simulated in 20 randomized environments, varying in terms of the robots' starting configurations and the poses of obstacles and objects, Fig. 2. Poses of the holonomic bases are uniformly sampled within scenario-specific ranges for (x, y, θ) , with the Euclidean distance of (x, y) from the tables varying between 6.4m and 3m, and the rotation $\theta \in [-2.0, 2.0]\text{rad}$. Object positions are sampled within annular regions around the centers of the tables. In all scenarios, we consider the task successfully accomplished if all agents reached

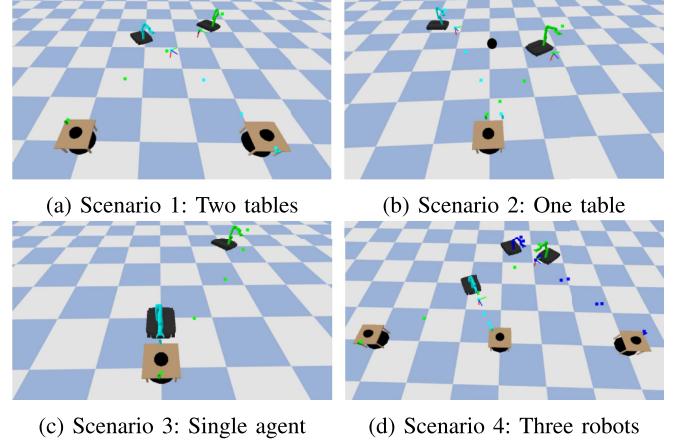


Fig. 2. Example environments of the simulated scenarios.

their target position within a task-dependent tolerance, given by a Euclidean distance of 0.07m. For comparison, we evaluate the success rate, time-to-success, collision rate during execution, and computation time of the low-level planner and high-level planner if applicable. The ego robot is approximated by two or five collision spheres, for the long-horizon planner and GF respectively, while other robots are represented by two spheres encapsulating the mobile base and the upper wrist link of the arm. Each table is modeled by two collision spheres. Collision violations are detected by checking contacts between all rigid bodies in the Pybullet physics engine.

The results in Table I show that the proposed G3F outperforms GF, P-RF and MPC in terms of success rate across all scenarios. The computational effort of the low-level planner within G3F is approximately 1 ms, achieving high reactivity, while the high-level planner is online adaptive in real-time as well. Although not optimized over, each robot achieves its task within reasonable time, while remaining collision-free in all scenarios compared to GF, P-RF and MPC.

In *Scenario 1*, the mobile manipulators are tasked to pick a cup placed on two different tables, forcing the robots to cross paths while avoiding each other, as illustrated in Fig. 2(a). G3F achieves a 100% success rate, improving significantly over GF

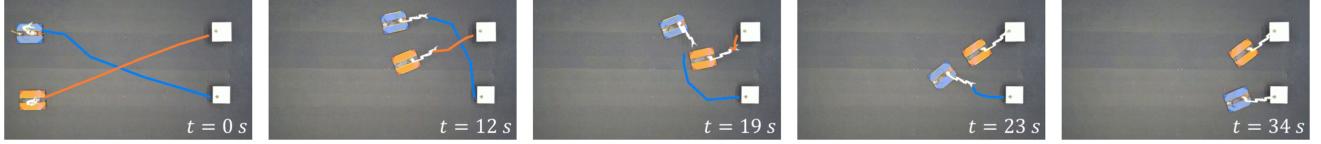


Fig. 3. Selected time frames for resolving a crossover in a multi-robot scenario using G3F. The optimized reference waypoints for the end-effectors, Ξ_{ee}^i , for the robots $i \in [1, 2]$ are shown as blue and orange lines respectively.

(70%) and P-RF (70%) and MPC (45%), showcasing the ability of G3F to solve long-horizon planning problems in dynamic multi-agent environments.

In *Scenario 2*, two mobile manipulators navigate closely to grasp assigned cups on the same table. As illustrated in Fig. 2(b), our method shows an 85% success rate, compared to the 35%, 35%, and 65% success rates achieved by GF, P-RF, and MPC, respectively. The benefit of optimizing the grasp pose (Fig. 4(a)) over a static grasp pose (Fig. 4(c)) is observed as both agents adapt their grasp pose given the current state of the other agent. Most G3F failures arise when the high-level QP cannot find a feasible solution due to local minima, which could be improved in future work by diversifying RF.

In *Scenario 3*, an agent is blocking the path to the table without adapting its pose to make room for the ego agent, shown in Fig. 2(c). Note that P-RF is omitted in Table I, as it requires a multi-agent setting to assign priorities. By considering the full task horizon to the target pose and adaptability of the grasp pose, our method achieved a 100% success rate, outperforming the other methods. In particular, MPC suffers from a short horizon, resulting in deadlock scenarios in local minima caused by collision avoidance.

To demonstrate the scalability of our method to an increased number of agents, we extended *Scenario 1* to three decentralized mobile manipulators in *Scenario 4* (Fig. 2(d)). The increased complexity leads to a performance drop for all methods. However, G3F still significantly outperformed the baseline methods without introducing additional computational overhead compared to *Scenario 1*. In contrast to geometric methods, MPC experienced a notable increase in computational time as the number of agents increased. GF, MPC and P-RF suffer from collisions, which could be mitigated by more conservative tuning, albeit at the cost of success rate. Future work could enhance social compliance by incorporating intention or goal estimation within the QP-horizon for navigation in dense multi-environments.

C. Real-World Experiments

We evaluate our method in several real-world scenarios including environments shared with humans, other mobile manipulators and an illustrative comparison against GF.

1) *Experiment 1. An Illustrative Comparison:* In Fig. 4, the advantage of globally-guided motion generation via G3F over vanilla GF is observed. In Fig. 4(c), the proposed G3F explores alternative grasp poses, thereby allowing both robots to reach their target. In addition, G3F avoids local minima (Fig. 4(d)), whereas GF halts at a local minimum caused by two spherically modeled obstacles (Fig. 4(b)).

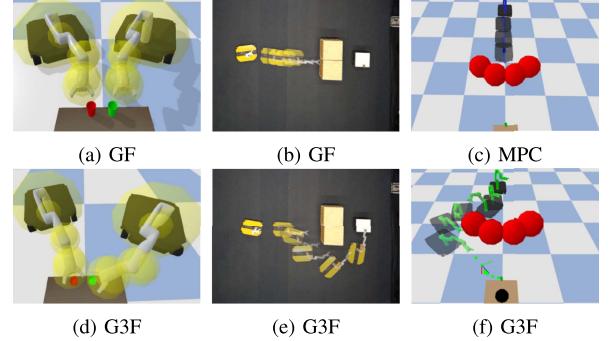


Fig. 4. Comparative examples of G3F versus GF and MPC illustrating the benefit of grasp-pose adaptation (Fig. 4(d)) and globally-guided motions (Fig. 4(e)-4(f)).

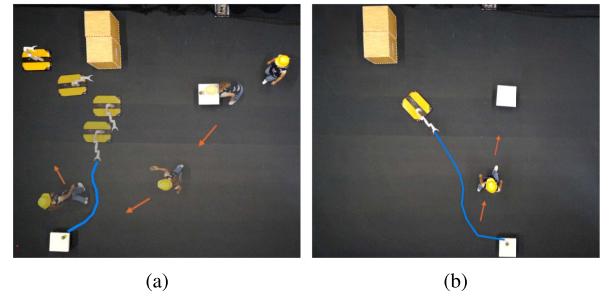


Fig. 5. In (a) and (b), G3F is demonstrated in two environments shared with a human moving the target object to another table. The blue trajectory shows the optimized path computed in real-time for the last visible frame, based on the current state of the environment.

2) *Experiment 2. Multi-Agent Scenarios:* In Figs. 3 and 5, mobile manipulators are tasked to grasp an object in an environment shared with another mobile manipulator or human respectively. In these decentralized multi-agent settings, we observe that the robot finds a globally-guided solution via G3F, by finding the solution to cross behind the other robot if crossing in front is no longer suitable in Fig. 3. In the presence of a human agent, G3F is able to react to the human changing the target pose online by moving the object, while avoiding collisions with the human, Fig. 5.

3) *Experiment 3. Pick and Place:* The proposed approach is suitable for optimizing the path from start to goal for a pick-and-place task, adapting the grasp and place pose accordingly. Currently half of all waypoints are used for the grasping stage and half for the placing stage. After grasping the object, we optimized only the placing manoeuvre, following the same approach as in previous experiments. We verified this approach in three different environments Fig. 6, where the other

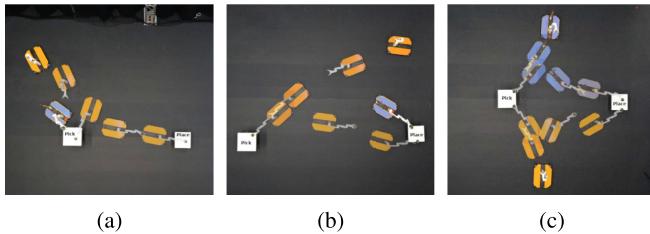


Fig. 6. Pick-and-place scenarios where G3F explores alternative grasp and place poses. Fig. 6(a) and (b) illustrate scenarios where the blue agent obstructs the precomputed place and grasp pose respectively, while in Fig. 6(c) both robots execute the proposed G3F.

robot is stationary in Fig. 6(a)-(b) or also controlled via G3F, Fig. 6(c). The mobile manipulators successfully execute the pick-and-place task, demonstrating the benefit of jointly optimizing the grasp and place pose based on the robot's and environment's current state.

VI. CONCLUSION

This work presents a framework for reactive object-informed whole-body motion generation, denoted as Globally-Guided Geometric Fabrics (G3F), allowing for collision-free motions in multi-agent settings, while considering the full task horizon from start to goal.

By utilizing RF to warm-start and linearize the optimization problem, we achieved real-time execution of the planner guiding the low-level Geometric Fabrics. The framework's performance was demonstrated on 9-DoF mobile manipulators in a decentralized manner, in a quantitative comparison against GF, P-RF and MPC, as well as in qualitative real-world experiments. Future work could reduce computation time through parallelization and improve robustness via diverse forward predictions via RF. Vision-based grasp prediction could provide static grasp poses for a wider range of objects.

ACKNOWLEDGMENT

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

- [1] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Sci. Robot.*, vol. 8, no. 84, 2023, Art. no. eadf7843.
- [2] A. Heins and A. P. Schoellig, "Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator," *IEEE Robot. Automat. Lett.*, vol. 8, no. 12, pp. 7986–7993, Dec. 2023.
- [3] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6177–6184, Oct. 2020.
- [4] J. Haviland and P. Corke, "NEO: A novel expeditious optimisation algorithm for reactive motion control of manipulators," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1043–1050, Apr. 2021.
- [5] M. Spahn, B. Brito, and J. Alonso-Mora, "Coupled mobile manipulation via trajectory optimization with free space decomposition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 12759–12765.
- [6] J. Ichnowski, Y. Avigal, V. Satish, and K. Goldberg, "Deep learning can accelerate grasp-optimized motion planning," *Sci. Robot.*, vol. 5, 2020, Art. no. eabd7710.
- [7] N. D. Ratliff, K. V. Wyk, M. Xie, A. Li, and M. A. Rana, "Optimization fabrics," 2020, *arXiv:2008.02399*.
- [8] S. Bakker, L. Knödler, M. Spahn, J. Böhmer, and J. Alonso-Mora, "Multi-robot local motion planning using dynamic optimization fabrics," in *Proc. IEEE Int. Symp. Multi-Robot Multi-Agent Syst.*, 2024, pp. 149–155.
- [9] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [10] G. Sanchez and J. C. Latombe, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2002, pp. 2112–2119.
- [11] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat. Symp.*, 2000, pp. 995–1001.
- [12] R. Shome, K. Solovey, A. Dobson, D. Halperin, and K. E. Bekris, "DRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning," *Auton. Robots*, vol. 44, no. 3, pp. 443–467, 2020.
- [13] A. Gawel et al., "A fully-integrated sensing and control system for high-accuracy mobile robotic building construction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2300–2307.
- [14] J. Wu et al., "TidyBot: Personalized robot assistance with large language models," *Auton. Robots*, vol. 47, pp. 1087–1102, 2023.
- [15] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*. Berlin, Germany: Springer, 2004.
- [16] H. Klein, N. Jaquier, A. Meixner, and T. Asfour, "A Riemannian take on human motion analysis and retargeting," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 5210–5217.
- [17] N. Ratliff and K. Van Wyk, "Fabrics: A foundationally stable medium for encoding prior experience," 2023, *arXiv:2309.07368*.
- [18] M. Xie et al., "Geometric fabrics for the acceleration-based design of robotic motion," 2020, *arXiv:2010.14750*.
- [19] M. Spahn, M. Wisse, and J. Alonso-Mora, "Dynamic optimization fabrics for motion generation," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2684–2699, Aug. 2023.
- [20] W. Zhi, I. Akinola, K. Van Wyk, N. D. Ratliff, and F. Ramos, "Global and reactive motion generation with geometric fabric command sequences," in *Proc. 2023 IEEE Int. Conf. Robot. Automat.*, 2023, pp. 939–945.
- [21] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *Proc. Int. Conf. Automated Planning Scheduling*, 2020, vol. 30, pp. 440–448.
- [22] V. N. Hartmann, A. Orthey, D. Driess, O. S. Oguz, and M. Toussaint, "Long-horizon multi-robot rearrangement planning for construction assembly," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 239–252, Jan. 2022.
- [23] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 844–851, Apr. 2020.
- [24] M. Toussaint, J. Harris, J. S. Ha, D. Driess, and W. Höning, "Sequence-of-constraints MPC: Reactive timing-optimal control of sequential manipulation," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13753–13760.
- [25] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6222–6227.
- [26] J. Mahler et al., "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau4984.
- [27] R. Newbury et al., "Deep learning approaches to grasp synthesis: A review," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3994–4015, Oct. 2023.
- [28] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.*, vol. 39, no. 2–3, pp. 183–201, 2020.
- [29] J. Ichnowski, M. Danielczuk, J. Xu, V. Satish, and K. Goldberg, "GOMP: Grasp-Optimized Motion Planning for bin picking," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 5270–5277.
- [30] S. Duenser, J. M. Bern, R. Pórrané, and S. Coros, "Interactive robotic manipulation of elastic objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3476–3481.
- [31] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, no. 4, pp. 637–672, 2020.
- [32] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi—A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.