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

Safety-Critical Dynamic Motion Generation for Manipulators Using Differentiable Distance Fields in Configuration Space

Xuemin Chi[1,2*], Jihao Huang[1*], Yiming Li[2], Bolun Dai[3], Zhitao Liu[1†], Sylvain Calinon[2†]

***Abstract*— Generating collision-free motions in dynamic environments is a challenging problem for high-dimensional robotics, particularly under real-time constraints. Control Barrier Functions (CBFs), widely utilized in safety-critical control, have shown significant potential for motion generation. However, for high-dimensional robot manipulators, existing QP formulations and CBF-based methods rely on positional information, overlooking higher-order derivatives such as velocities. This limitation may lead to reduced success rates, decreased performance, and inadequate safety constraints. To address this, we construct time-varying CBFs (TVCBFs) that consider dynamic obstacles. Our approach leverages recent developments on distance fields for articulated manipulators, a differentiable representation that enables the mapping of objects' position and velocity into the robot's joint space, offering a comprehensive understanding of the system's interactions. This allows the manipulator to be treated as a point-mass system thus simplifying motion generation tasks. Additionally, we introduce a time-varying control Lyapunov function (TVCLF) to enable whole-body contact motions. Our approach integrates the TVCBFs, TVCLF, and manipulator physical constraints within a unified QP framework. We validate our method through simulations and comparisons with state-of-the-art approaches, demonstrating its effectiveness on a 7-axis Franka arm in real-world experiments. Source codes, experimental data and videos are available on the project webpage: `https://sites.google.com/view/sdfcdf-tvcbfs-qp`.**

I. Introduction

Robots are expected to respond quickly and safely in dynamic environments. For high-dimensional systems, generating safe motions at high frequencies is particularly challenging [1]. A hierarchical approach is predominantly applied in static environment scenarios, where a globally feasible path is planned offline, followed by full-horizon trajectory optimization to smooth the path while accounting for more complex constraints. The trajectory optimization process is commonly formulated as a nonlinear programming problem (NLP) and is solved using gradient-based solvers. Techniques such as CHOMP [2] and TrajOpt [3] are effective but may not respond quickly enough to the demands of dynamic environments which require faster reactions.

Model predictive control (MPC) methods can address this issue by using shorter horizons. The formulation is often an NLP problem that requires an initial guess. This includes traditional gradient-based MPC [4]–[6] and recent gradient-free MPC using sampling-based methods, e.g. MPPI [7] and VP-STO [8]. However, the replanning frequencies of these algorithms are struggling to balance safety and efficiency. The computation of system dynamics and nonlinear constraints for each rollout state grows exponentially. A longer horizon allows a higher chance of finding a feasible solution, as MPC can ensure safety within the preview window if a solution exists, but this comes at the cost of higher computational demands.

The quadratic programming (QP) formulation provides real-time efficiency for high-dimensional systems and has become popular in recent years [9], [10]. Recent works have explored the formulation of QP controllers for motion generation tasks [1], [11], by modeling the geometry of the robot as a signed distance field (SDF) [12], [13] or robot distance field (RDF) [13], achieving reactive motion policies with a high frequency of over 200 Hz. This concept has also been explored in autonomous navigation [14] and drones [15] in generating collision-free trajectories. These methods typically consider the positional relationship of objects and the robot, ignoring high-order information such as velocities. Such limitation prevents the robot from reacting appropriately to objects. For instance, the robot's motion should differ based on the object's speed and direction, particularly when the object is approaching or away from the robot. As a result, the robot's behavior usually relies on manually set parameters, limiting its applicability in dynamic scenarios and the safety behavior is not guaranteed.

To address this challenge, it is crucial to describe the object's dynamics, including both position and velocity information. In this paper, we propose to exploit the paradigm of control barrier functions [16] to address this challenge, allowing the robot to react safely based on the objects' dynamic states. Such an approach has been explored in task space-based planning fields such as autonomous driving [17], providing a simple and efficient approach for safety control synthesis. However, this task becomes more complex when applied to high-dimensional robotic manipulators due to the nonlinear mapping of geometries in task and joint spaces. Thus, previous studies typically construct control barrier functions (CBFs) based on the distance information [18], [19].

We propose to construct time-varying control barrier functions (TVCBFs [20]) that incorporate the velocity infor-

∗ Equal Contribution

This work was supported in part by National Key R&D Program of China (Grant NO. 2021YFB3301000); National Natural Science Foundation of China (NSFC:62173297), Zhejiang Key R&D Program (Grant NO. 2022C01035). (†Corresponding author)

1 Authors are associated with the Department of Control Science and Engineering, Zhejiang University.

2 Authors are associated with Idiap Research Institute, 1920 Martigny, Switzerland, and also with Ecole Polytechnique Federale de Lausanne (EPFL), 1015 Lausanne, Switzerland.

3 Authors are associated with Fauna Robotics, New York, New York, United States of America.

mation of objects. This is achieved by leveraging recent advancements in distance fields for robot manipulators, a differentiable representation that can reason the geometry of obstacles in robot configuration space [11], [13]. The derivative of the distance field corresponds to the velocity of obstacles in joint space. This representation allows us to fully consider the articulated robot and the surrounding environment as a point-mass system, simplifying the motion generation problem. We demonstrate a paradigm that synthesizes a safety-critical controller in configuration space. Additionally, we design a time-varying control Lyapunov function (TVCLF) to address a whole-body dynamic reaching task. The primary contributions of this work are as follows:

- We introduce an approach to map the position and velocity of the objects to robot configuration by leveraging the differentiable distance field representation of robot manipulators. TVCBFs are developed using the differentiable distance field to ensure safety-aware motion generation.
- We construct a singularity-free TVCLF to facilitate whole-body reaching tasks in dynamic configuration space. The constraints of TVCLFs, TVCBFs and physical limits of robotic arms are systematically formulated as a QP.
- We validate our approach through numerical simulations on planar arms, with benchmarks against baselines providing a detailed analysis. To further validate the proposed approach, we performed simulations and real-robot experiments using a 7-axis Franka robotic arm.

II. PRELIMINARIES

Our safety-critical controller is built upon CBFs and CLFs, which are formulated using differentiable distance fields. This section presents the necessary preliminaries.

A. Differentiable Distance Fields for Robotic Manipulators

SDFs are popular representations in robotics for modeling environments. Recent work extends this concept to articulated robots by either exploiting neural networks to approximate the distance function [12] or leveraging the kinematics chain of the robot [13]. Consider an n-DOF manipulator with states x_r described by joint angles $q = [q_1, \dots, q_n] \in \mathbb{R}^n$, we have an SDF expressed as:

$$\Gamma_s(q, p) = \pm \min_{p' \in \partial r(q)} \|p - p'\|_2 , \tag{1}$$

where $p \in \mathbb{R}^3$ is a spatial point in the task space. p' is the point on the manipulator's surface, denoted by the zero level contact set $\partial r(q) = \left\{p \in \mathbb{R}^3 \mid \Gamma_s(q, p) = 0\right\}$. $\Gamma_s(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^3 \to \mathbb{R}$ provides an SDF value.

In [11], the authors introduce configuration space distance fields (CDFs), which also estimate the distance given a specific point and the robot's configuration. However, instead of modeling the task space distance from a point to the robot's surface, CDFs measure the distance that indicates the minimal joint motion required by the robot to establish contact with the point. In other words, this representation measures the distance in joint space by implicitly solving a whole-body inverse kinematics task. The CDF is defined as:

$$\Gamma_c(q, p) = \min_{q' \in \partial w(p)} \|q - q'\|_2 , \tag{2}$$

where $\Gamma_c(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^3 \to \mathbb{R}$. Here, q' represents configurations in the whole-body contact set, defined as $\partial w(p) = \{q' \in \mathbb{R}^n \mid \Gamma_s(q', p) = 0\}$. This set includes all joint configurations that bring the manipulator surface into contact with the point p.

B. Control Lyapunov Functions-Based QP

Control Lyapunov functions are proposed to stabilize a system with a feedback control law. Consider a nonlinear control-affine system

$$\dot{x} = f(x) + g(x)u, \tag{3}$$

where the state $x \in D \subset \mathbb{R}^n$ and the control input $u \in U \subset \mathbb{R}^m$. Here, U denotes the admissible control set, defined as $U = \{u \in \mathbb{R}^m \mid u_{\min} \leq u \leq u_{\max}\}$, and D denotes the admissible state set, given by $D = \{x \in \mathbb{R}^n \mid x_{\min} \leq x \leq x_{\max}\}$. The drift term $f : \mathbb{R}^n \to \mathbb{R}^n$ and the control influence matrix $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are both locally Lipschitz continuous.

***Definition** 1 (Class $\mathcal{K}$ Functions [21]):* A Lipschitz continuous function $\mu : [0, a) \to [0, \infty)$, where $a > 0$, is classified as a $\mathcal{K}$ function if it is strictly increasing and satisfies $\mu(0) = 0$. Additionally, a function is in class $\mathcal{K}_\infty$ if it meets the criteria of class $\mathcal{K}$, with the further properties that $a = \infty$ and $\mu(b) \to \infty$ as $b \to \infty$. When the domain of μ is extended to $[-\infty, +\infty]$, we call it an extended class $\mathcal{K}_\infty$ function.

A continuously differentiable function V is a CLF for the system (3) if it is positive definite and satisfies [21]

$$\inf_{u \in U} \left[L_f V(x) + L_g V(x) u\right] \leq -\gamma(V(x)), \tag{4}$$

where $L_f V(x) := \frac{\partial V}{\partial x} f(x)$ and $L_g V(x) := \frac{\partial V}{\partial x} g(x)$ are Lie-derivatives of $V(x)$, $\gamma(\cdot)$ belongs to class $\mathcal{K}$ function.

Since CLF constraints are affine in controls, we can define a QP and the objective function is minimizing the control efforts:

$$\begin{aligned} u^*(x) = &\underset{u \in U}{\operatorname{argmin}} \ \frac{1}{2} u^\top R u \qquad (5) \\ &\text{s.t. } L_f V(x) + L_g V(x) u \leq -\gamma(V(x)), \qquad \text{(CLF)} \end{aligned}$$

which is denoted as CLF-QP.

C. Control Barrier Functions-Based QP

***Definition** 2 (Forward Invariance):* Let $x(t)$ denote the unique solution to (3) starting from an initial state $x_0 \in \mathcal{C}$. The controller $\pi(x)$ renders the system (3) safe with respect to $\mathcal{C}$ if it ensures that $x(t)$ remains within the safe set $\mathcal{C}$ for all $t \in I(x_0)$, where $I(x_0) = [t_0, t_{\max})$ denotes the maximum interval of existence of $x(t)$.

Consider a set $\mathcal{C}$ defined as the superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} = \{x \in D \subset \mathbb{R}^n \mid h(x) \geq 0\}, \tag{6}$$

throughout this paper, we refer to $\mathcal{C}$ as a safe set. The safety constraints in (3) can be framed as enforcing the forward invariance of a set, i.e., the system stays inside the safe set. Given access to a Lipschitz continuous controller $u = \pi(x)$ such that $\dot{x} := f(x) + g(x)\pi(x)$.

A continuously differentiable function h is called a CBF if it satisfies $\frac{\partial h}{\partial x} \neq 0$ for all $x \in \partial\mathcal{C}$ and if, for the system (3), the following condition holds:

$$\sup_{u \in U} [L_f h(x) + L_g h(x) u] \geq -\alpha(h(x)), \tag{7}$$

where $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ and $L_g h(x) = \frac{\partial h(x)}{\partial x} g(x)$ are the Lie derivatives of $h(x)$ with respect to f and g, respectively. Here, $\alpha(\cdot)$ is an extended class $\mathcal{K}_\infty$ function.

The CBF constraint, being affine in the control input u, can be integrated directly into a QP alongside the CLF constraint:

$$\begin{aligned} u^*(x) = &\underset{u,\delta}{\operatorname{argmin}} \ \frac{1}{2} u^\top R u + p\delta^2 && (8)\\ &\text{s.t. } L_f V(x) + L_g V(x) u + \gamma(V(x)) \leq \delta, && \text{(CLF)}\\ &\quad L_f h(x) + L_g h(x) u \geq -\alpha(h(x)), && \text{(CBF)}\\ &\quad u \in U, \ \delta \in \mathbb{R}_+, \end{aligned}$$

where δ is a relaxation variable introduced to prioritize safety over stability requirements, allowing the optimization problem to remain feasible for an appropriately chosen weight $p > 0$.

III. Controller Design

We aim to control an n-DOF manipulator to perform whole-body collision avoidance and goal-reaching tasks in dynamic environments with K moving obstacles in the task space, where each obstacle's state follows $x_p = p_i(t) \in \mathbb{R}^3$ for $i \in \{1, \ldots, K\}$ and one moving target object given state $x_g = p_g(t) \in \mathbb{R}^3$. All joint angles of the manipulator are bounded by joint limits $q \in [q_{\min}, q_{\max}]$. This work aims to design a safety-critical controller in a configuration space where the manipulator system is linear, $u = [\dot{q_1}, \ldots, \dot{q_n}] \in U \subset \mathbb{R}^n$ and the geometry is a point that accomplishes the above tasks in real-time, with a desired control frequency of ≥ 200 Hz. In this section, we detail the development of a velocity-based controller designed to enable collision-free, whole-body reaching motions.

A. Design TVCBFs in Configuration Space

This section focuses on the design of TVCBFs for velocity-controlled manipulator systems operating in dynamic configuration space with moving obstacles. As the obstacles move, the safe set (6) for i-th dynamic obstacle becomes time-dependent and has the form [22] of :

$$\mathcal{C}_i(t) = \{q \in D \subset \mathbb{R}^n \mid h_i(q, p_i, t) \geq 0\}, \tag{9}$$

where $h_i(q, p_i, t)$ is the TVCBF for the i-th dynamic obstacle, measuring the distance between the obstacle and the manipulator's surface in dynamic environments. If an SDF is used, the corresponding SDF-TVCBF is denoted as $h_i^s(q, p_i, t)$, with a differentiable distance field following (1) $\Gamma_s(q, p_i, t)$, as illustrated in Fig.1d. In its configuration space representation (Fig.1c). If a CDF is used, the CDF-TVCBF is represented as $h_i^c(q, p_i, t)$, with the differentiable distance field $\Gamma_c(q, p_i, t)$ following (2), as shown in Fig. 1a.

Adapting the standard CBF condition (7) to TVCBFs [23], [24] setting yields:

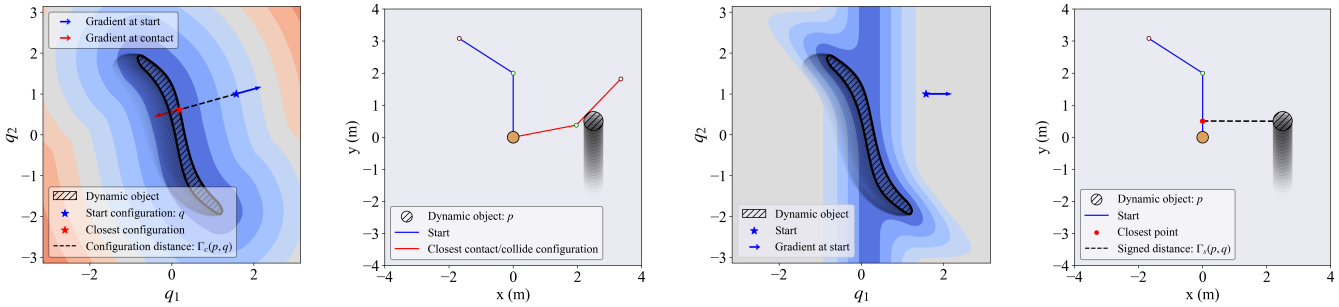
$$\begin{aligned} \sup_{u \in U} \Big[&L_f h_i(q, p_i, t) + L_g h_i(q, p_i, t) u \\ &+ \frac{\partial h_i(q, p_i, t)}{\partial p_i} \frac{\mathrm{d}p_i}{\mathrm{d}t} \Big] \geq -\alpha(h_i(q, p_i, t)), \end{aligned} \tag{10}$$

where $\frac{\partial h_i(q,p_i,t)}{\partial p_i} \frac{\mathrm{d}p_i}{\mathrm{d}t}$ accounts for the dynamics of i-th obstacle $p_i(t)$, ensuring that the robot's state remains within the safe set $\mathcal{C}_i(t)$ even as obstacles move. Given the manipulator system dynamics $\dot{q} = f_r(q) + g_r(q) u_r$ where $f_r(q) = 0$, and $g_r(q) = I$. Therefore, the first term simplifies to $L_f h_i(q, p_i, t) = \frac{\partial h_i(q,p_i,t)}{\partial q} f(q) = 0$.

The second term $L_g h_i\,(q, p_i, t)\, u = \frac{\partial h_i(q,p_i,t)}{\partial q} g(q) u$ describes how each joint influences the distance to collision with the obstacle $p_i(t)$. When $L_g h_i(q, p_i, t) = 0$, the robot cannot react to it, as the control input u vanishes in the TVCBF constraint (10). This issue arises when using an SDF $\Gamma_s(q, p_i, t)$, which fails to provide complete gradient information. Many existing methods, including those based on geometric primitives, polytopes, and learning-based models, fall into this category [1], [12], [18], [19]. For example, in Fig.1c and Fig.1d, the manipulator only has information for q_1, $L_g h_i(q, p_i, t) = [*, 0]$. If the obstacle is positioned below the manipulator's base or in a location where the closest point to the obstacle becomes the base, the information for both q_1 and q_2 disappears, deactivating the i-th TVCBF constraint. Although the TVCBF constraint may reactivate when $\frac{\partial h_i(q,p_i,t)}{\partial q} g(q)$ becomes nonzero as the obstacle moves, this delay may be too late for the manipulator to effectively avoid a dynamic obstacle.

In contrast, our second approach is built upon the CDF, ensuring that a complete gradient condition is always satisfied. Comparisons and demonstrations are provided in Section V-B, with accompanying multimedia videos. Unlike CDF-TVCBF, which remains valid only when the obstacle is within the task space, SDF-TVCBF can retain its validity even when the obstacle is outside the task space, as long as a gradient exists. This ensures that the constraint in (10) remains meaningful. A key advantage of our formulation is that $\Gamma(p, q_i, t)$ enables parallelized inference on a GPU, allowing distance computations to scale efficiently with the number of obstacles while remaining agnostic to obstacle geometries.

In this paper, we propose a solution for the third term, $\frac{\partial h_i(q,p_i,t)}{\partial p_i} \frac{\mathrm{d}p_i}{\mathrm{d}t}$, which addresses the challenge of bridging the task space and configuration space of the manipulator. Specially, $\frac{\mathrm{d}p_i}{\mathrm{d}t}$ represents the velocity v_i of the i-th obstacle in the task space, capturing its position changes over time. As the obstacle $p_i(t)$ moves in the task space, $\frac{\partial \Gamma(q,p_i,t)}{\partial p_i}$ captures how distance changes depending on the choice of the distance field. For SDF-TVCBF constraints, it



(a) CDF: Configuration Space (b) CDF: Task Space (c) SDF: Configuration Space (d) SDF: Task Space

Fig. 1: The distance and gradient information of CDF and SDF. (a) Object velocity captured in the configuration space. (b) The object is reachable in the task space. (c) The gradient is incomplete, the robot only has gradient information with respect to q_1. (d) The object velocity is directly employed in the task space.

describes when i-th obstacle moves in the task space, how the distance-to-collision in task space changes. In contrast, CDF-TVCBF constraints characterize how the distance-to-collision changes in the configuration space when the obstacle moves in the task space, an aspect investigated and applied for the first time in this paper.

For joint limits on the state variables q, we define two functions to account for violations of physical limits:

$$h_{\min}(q) = q - q_{\min}, \quad h_{\max}(q) = q_{\max} - q, \tag{11}$$

which describe the boundary constraints as CBFs:

$$L_f h_{\min}(q) + L_g h_{\min}(q)u \geq -\alpha_{\min}(h_{\min}(q)), \tag{12}$$
$$L_f h_{\max}(q) + L_g h_{\max}(q)u \geq -\alpha_{\max}(h_{\max}(q)), \tag{13}$$

where $\alpha_{\max}(\cdot)$ and $\alpha_{\min}(\cdot)$ are extended class $\mathcal{K}_\infty$ functions.

B. Design CLFs in Configuration Space

We intend to design a CLF to stabilize our system at the target object. To account for the dynamic object's motion, we define a TVCLF, which modifies the standard CLF condition (4) as follows:

$$\inf_{u\in U}\Big[L_f V(q,p_g,t) + L_g V(q,p_g,t)u + \frac{\partial V(q,p_g,t)}{\partial t}\frac{\mathrm{d}p_g}{\mathrm{d}t}\Big] \leq -\gamma(V(q,p_g,t)), \tag{14}$$

where $\frac{\partial V(q,p_g,t)}{\partial t}\frac{\mathrm{d}p_g}{\mathrm{d}t}$ captures the object's dynamics. Following the notation in (1), let $\Gamma_c(q,p_g,t)$ denote the configuration distance between the robot at joint configuration q and the object at state p_g in task space. When the object's position p_g is out of the task space, the concept of configuration distance is undefined and the configuration space for the object is empty. The dynamics of the object is $\dot{p}_g = \tilde{f}(p_g) + \tilde{g}(p_g)\tilde{u}$ where $\tilde{f}(p_g) = 0$ and $\tilde{g}(p_g) = I$. Therefore, first term $L_f V(q,p_g,t) = \frac{\partial V(q,p_g,t)}{\partial x} f(q)$ is zero. The second term in (14) follows $L_g V(q,p_g,t) = \frac{\partial V(q,p_g,t)}{\partial q} g(q) = \frac{\partial \Gamma_c(q,p_g,t)}{\partial q} g(q)$. The third term follows a similar logic to TVCBFs, $\frac{\partial V(q,p_g,t)}{\partial p_g}\frac{\mathrm{d}p_g}{\mathrm{d}t} = \frac{\partial d_c(p_g,q)}{\partial p_g}\dot{p}_g$. where $\dot{p}_g = v_g$.

C. TVCBF-TVCLF-QP Controller Synthesis

The optimization problem (8) optimizes only the control inputs and relaxation variables, allowing the direct integration of admissible control constraints into the QP formulation. The final optimization problem incorporating the TVCLF and TVCBFs constraints denoted as SDF/CDF-TVCBF-TVCLF is given as

$$u^*(q) = \underset{u,\delta}{\operatorname{argmin}} \frac{1}{2}u^\top R u + p\delta^2 \tag{15a}$$
$$\text{s.t. } L_f h_i(q,p_i,t) + L_g h_i(q,p_i,t)u + \frac{\partial h_i(q,p_i,t)}{\partial p_i}\frac{\mathrm{d}p_i}{\mathrm{d}t} \geq -\alpha(h_i(q,p_i,t)), \tag{15b}$$
$$L_f V(q,p_g,t) + L_g V(q,p_g,t)u + \frac{\partial V(q,p_g,t)}{\partial p_g}\frac{\mathrm{d}p_g}{\mathrm{d}t} + \gamma(V(q,p_g,t)) \leq \delta, \tag{15c}$$
$$L_f h_{\min}(q) + L_g h_{\min}(q)u \geq -\alpha_{\min}(h_{\min}(q)), \tag{15d}$$
$$L_f h_{\max}(q) + L_g h_{\max}(q)u \geq -\alpha_{\max}(h_{\max}(q)), \tag{15e}$$
$$i \in \{1,\dots,K\}, u \in U, \delta \in \mathbb{R}_+,$$

where $R \in \mathbb{R}^{n\times n}$ represents weights that prioritize control effort for each robotic joint, ensuring minimal effort. The parameter p controls the stabilization velocity and is particularly important in dynamic whole-body reaching tasks where exponential stabilization cannot be guaranteed, and the TVCLF is often non-monotone. Class $\mathcal{K}$ functions $\gamma(\cdot)$ and $\alpha(\cdot)$ are chosen as simple linear scalar functions for computational efficiency, i.e., $\gamma(V(q)) = k \cdot V(q)$, where k is the linear scalar. The TVCLF constraint guides the robot to generate dynamic whole-body reaching behaviors and is relaxed by δ to handle transient deviations. The TVCBF constraint ensures that the robot avoids dynamic obstacles during operation. Since CLFs and TVCLFs are both directly designed in configuration space without using the forward kinematics mapping, the singularity problem becomes a simple tracking problem and can easily solved by our QP formulation.

TABLE I: Setup of Simulation Parameters

Notation	Meaning	Value
Δt	Time step of simulation	$0.1\,\mathrm{s}$
l_1, l_2	The length of planar arm links	$2.0\,\mathrm{m}$
ϵ_{CBF}	The safety margin for collision avoidance	0.05
ϵ_{CLF}	The tolerances for whole-body reaching	$0.02\,\mathrm{rad}$
$q_{\min}$	Robot's minimum joint angles	$-\pi\,\mathrm{rad}$
$q_{\max}$	Robot's maximum joint angles	$\pi\,\mathrm{rad}$
$\dot{q}_{\min}$	Robot's joint velocities	$-2.0\,\mathrm{rad/s}$
$\dot{q}_{\max}$	Robot's maximum joint velocities	$2.0\,\mathrm{rad/s}$
$\gamma(\cdot)$	The class functions for all TVCLFs	1.0
$\alpha(\cdot)$	The class functions for all TVCBFs	1.0

IV. Numerical Simulations and Experiments

A. Implementation Details

In this section, we present the results of simulations and experiments on robotic systems to validate our approach. To ensure safety, we incorporated a safety margin ϵ_{cbf} in the TVCBF. Similarly, a destination margin ϵ_{clf} was included in the TVCLF to account for practical tolerances in whole-body reaching. The simulation parameters for the robots and the optimization controller are summarized in Table I. The qpOASES solver [25] was used to solve all QP optimization problems. We conducted a series of simulations and experiments to evaluate our approach. For additional results and details, please refer to the supplementary materials.

B. 2D Planar Arm Dynamic Collision Avoidance

The first scenario demonstrates the effectiveness of our SDF-TVCBF and CDF-TVCBF formulations that consider the velocity of obstacles for safety-critical control in dynamic environments. The planar arm's initial state is $(q_1(0), q_2(0)) = (2.5, 0.5)$ and the goal state is $q_{\text{goal}} = (-2.7, 0.5)$. A moving circular obstacle with a radius of 0.3 is introduced, initially positioned at $(2.4, -2.4)$ and moving at four different velocities $(0.5, 1.5, 2.5, 4.5)\,\mathrm{m/s}$ to evaluate the effect of the obstacles dynamics.

As shown in Fig. 2a, the CDF-TVCBF-QP controller effectively guides the robot to avoid obstacles under various velocity conditions. At a velocity of $0.5\,\mathrm{m/s}$, the planar arm moves toward the target while avoiding the obstacle when necessary. As the obstacle velocity increases to $1.5\,\mathrm{m/s}$ and $2.5\,\mathrm{m/s}$, the controller primarily adjust the joint angle q_1. At the highest velocity of $4.5\,\mathrm{m/s}$, the controller ensures safety by moving the first link backward, enabling the robot to avoid the highly dynamic obstacle. Additionally, the uniformly distributed level sets of the CDF in the configuration space confirm the existence of non-zero gradients, which is crucial for effective control. Fig. 2c further illustrates safety-aware behaviors in the trajectories of both the end-effector and the first joint.

In contrast, the SDF-TVCBF constraint exhibits different safety-aware behaviors. Unlike the CDF-TVCBF constraint, which proactively adjusts two joints to avoid the obstacle, the SDF-TVCBF keeps q_2 largely unchanged initially, relying solely on q_1 for obstacle avoidance. This occurs because the gradient information $\frac{\partial h_i^s(x,t)}{\partial q}$ does not effectively guide the motion of q_2. At the initial stage, the closest point to the obstacle lies on the first link, resulting in a zero gradient with respect to q_2. As shown in Fig. 2b, the level sets of the SDF reveal that gradients either become zero or align parallel to q_1. In more general scenarios, when the closest point shifts to the manipulator's base, the gradient becomes zero and the manipulator is not able to react effectively to the moving obstacle. From path length of end-effector trajectories shown in Fig. 2c and Fig. 2d, we can also see that complete gradient information would help converge faster.

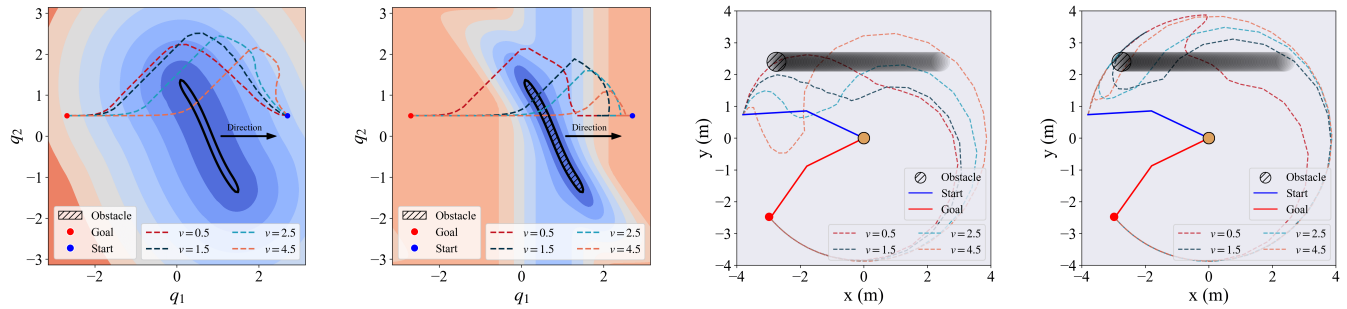
C. Planar Arm Dynamic and collision-free whole-body reaching

This section demonstrates the efficacy of the CDF-TVCBF-TVCLF-QP formulation in a dynamic reaching task. While our method is agnostic to object geometry, a circular object with a radius of $0.3\,\mathrm{m}$ is used for simplicity. The target is initially placed at $p_g = (2.2, 2.3)$ and moves downward with a velocity $v_g = (0.0, -0.8)\,\mathrm{m/s}$. Three obstacles, positioned at $p_1 = (1.9, -2.45)$, $p_2 = (2.4, 2.4)$, move with velocities $v_1 = (0.0, 1.8)\,\mathrm{m/s}$ (upward), $v_2 = (-1.5, 0.0)\,\mathrm{m/s}$ (leftward), and $v_3 = (1.5, 0.0)\,\mathrm{m/s}$ (rightward). The results are illustrated in Fig. 3. The planar arm progresses through four distinct stages to reach the target object:

Stage 1: Initial Approach ($t = 0.0\,\mathrm{s}$ to $t = 0.8\,\mathrm{s}$). During this stage, the robot moves toward the target object while avoiding the slow-moving obstacle. Joint velocities are initially negative, allowing the robot to approach the target while avoiding collisions. As shown in Fig. 3d, joint velocities $\dot{q}_1$ and $\dot{q}_2$ begin to increase toward positive values, indicating that the robot is preemptively planning to avoid future obstacles. The TVCLF decreases, signifying progress toward the target.

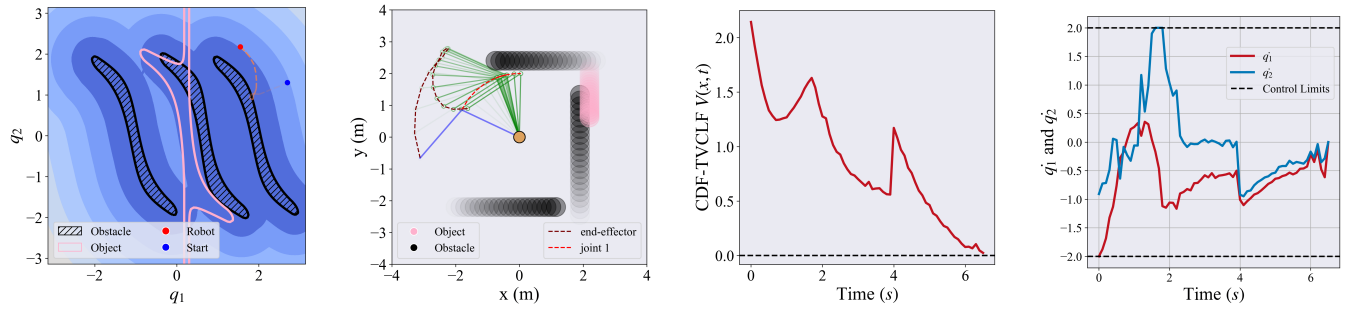
Stage 2: Safety Prioritization ($t = 0.8\,\mathrm{s}$ to $t = 1.7\,\mathrm{s}$). In this stage, the controller prioritizes safety over whole-body reaching, leading to an increase in the TVCLF as the robot temporarily moves away from the target. Joint velocity $\dot{q}_1$ becomes positive, causing the arm to adjust its configuration to avoid collisions, as shown in Fig. 3b. Meanwhile, the target object continues moving downward, increasing the TVCLF.

Stage 3: Gradual Reaching ($t = 1.7\,\mathrm{s}$ to $t = 3.9\,\mathrm{s}$). At this stage, the second obstacle overlaps with the target object, creating configurations that allow reaching but are not collision-free. The robot aims to reach the zero level set of the target object, as shown in Fig.3a. Since the first link is closer to the target, the controller attempts to reach the object using this link. As demonstrated in Fig.3c, the TVCLF decreases, indicating progress toward the target. Joint velocity $\dot{q}_1$ becomes negative, rotating the link clockwise to approach the target. However, the setup intentionally places the target object further from the robot, limiting the available time window for the first link to complete the reach. By $t = 3.9\,\mathrm{s}$, this time window closes, making it impossible for the first link to reach the target. This is reflected in a brief increase in the TVCLF (the cusp), indicating the need for a strategy adjustment.



(a) C-Space: CDF-TVCBF-QP (b) C-Space: SDF-TVCBF-QP (c) T-Space: CDF-TVCBF-QP (d) T-Space: SDF-TVCBF-QP

Fig. 2: Illustration of the CDF/SDF-based TVCBF formulation. Arrows indicate the obstacle's motion direction. In the task space, the trajectories of the end-effector and the first joint are depicted in distinct colors.



(a) C-Space (b) T-Space (c) TVCLF values (d) Control variables

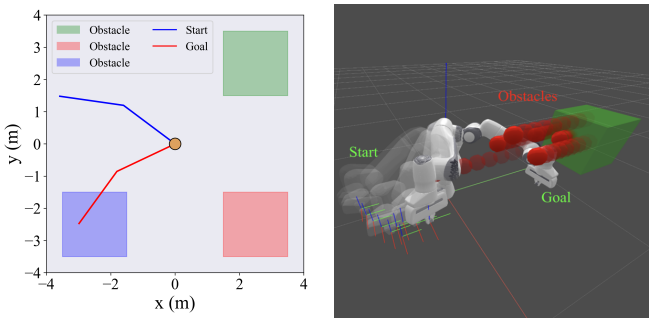
Fig. 3: The illustration of collision-free and whole-body reaching in dynamic environments. The first two figures are a snapshot at $t = 2.2\,\mathrm{s}$. The values of TVCLF and control variables are full-time horizons to show the quantitative details.

Stage 4: Switching Strategy ($t = 3.9\,\mathrm{s}$ to $t = 6.7\,\mathrm{s}$). After the first link's time window closes at $t = 3.9\,\mathrm{s}$, the CDF-TVCBF-TVCLF-QP controller switches its strategy to use the second link to reach the target. At this point, the gradient of the TVCLF with respect to q_2 becomes active, enabling the robot to adjust its configuration effectively. Joint velocity q_2 becomes negative, gradually moving the second link closer to the target object. The TVCLF decreases steadily between $t = 3.9\,\mathrm{s}$ and $t = 6.7\,\mathrm{s}$. As the robot approaches the target, joint velocities slow, and the planar arm successfully reaches the target object with the second link.

We observe that relaxing the TVCLF constraint is essential for two reasons. First, the controller must balance safety and stabilization under dynamic conditions. Second, as the target object moves, the TVCLF may increase, requiring the robot to adaptively decide which body part to use for reaching, ensuring effective and collision-free motion generation.

D. Performance Evaluation

To quantitatively evaluate our method, we compare its performance with a set of baselines. The first two baselines construct safety constraints using separation hyperplanes (SH) based on gradient information, denoted as SDF-SH-QP [12] and CDF-SH-QP [11]. The third baseline is denoted as SDF-CBF-QP proposed in [18]. This method formulates CBFs using signed distances computed by GJK and EPA algorithms. Since these baselines do not include CLF constraints, we also test simplified versions of our approach, denoted as CDF-TVCBF-QP and SDF-TVCBF-QP, with the CLF constraint removed. We use the following metrics to assess controller performance:



(a) 2D benchmark setup (b) 7D benchmark setup

Fig. 4: The 2D and 7D setup for benchmarking. Dynamic obstacles are randomly generated from the 2D and 7D box-shape region.

1) **Success Rate**: The percentage of trials where the controller successfully reaches the goal while avoiding obstacles.
2) **Time to Reach**: The time taken to reach the goal in seconds. If the controller fails, the maximum allowable time $20\,\mathrm{s}$ is recorded.

TABLE II: Performance benchmark: motion planning in highly dynamic environments.

Methods	Time to Reach (s)									Path Length (m)									Success Rate		
	S1			S2			S3			S1			S2			S3			S1	S2	S3
2D	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	-	-	-
CDF-TVCBF-QP (ours)	2.9	3.8	**3.2**	3.2	4.5	**3.8**	3.0	6.4	**4.4**	6.12	7.50	6.72	7.48	8.75	7.47	6.46	10.43	**8.25**	1.00	1.00	**1.00**
SDF-TVCBF-QP (ours)	3.5	4.8	3.9	3.4	4.5	3.9	3.1	6.4	4.8	6.22	7.71	6.64	6.30	8.03	**7.06**	6.38	12.37	8.52	1.00	1.00	**1.00**
SDF-CBF-QP ([18])	2.9	2.9	19.8	-	-	-	-	-	-	6.13	6.13	**6.13**	-	-	-	-	-	-	0.01	0.00	0.00
SDF-SH-QP ([13], [12])	3.1	4.9	8.7	3.2	4.4	13.2	3.0	6.3	9.5	6.73	9.52	7.58	6.66	8.69	7.41	6.63	10.43	8.85	0.69	0.41	0.68
CDF-SH-QP ([11])	3.1	4.7	6.9	3.1	4.4	12.6	3.0	6.3	7.2	6.54	10.04	7.48	6.82	9.63	7.62	6.68	13.62	9.60	0.79	0.45	0.82
7D	C1			C2			C3			C1			C2			C3			C1	C2	C3
CDF-TVCBF-QP (ours)	2.8	4.0	**3.1**	3.5	15.8	**12.3**	3.3	10.3	**7.1**	2.64	4.09	3.24	2.06	7.19	4.43	3.67	6.84	4.83	1.00	0.66	**0.92**
SDF-TVCBF-QP (ours)	2.8	4.3	3.3	3.4	10.5	14.0	3.7	13.0	9.6	2.78	4.35	3.51	2.29	4.80	**2.77**	4.21	10.04	5.09	1.00	0.42	0.81
SDF-CBF-QP ([18])	2.8	3.1	9.3	-	-	-	3.0	4.2	13.9	2.38	2.16	**2.63**	-	-	-	3.53	4.03	3.73	0.62	0.00	0.37
SDF-SH-QP ([13], [12])	2.5	3.4	9.7	-	-	-	2.6	6.7	16.2	2.09	3.97	2.78	-	-	-	3.75	6.44	4.62	0.63	0.00	0.24
CDF-SH-QP ([11])	2.5	2.9	9.3	-	-	-	2.5	2.8	15.8	2.07	3.49	2.68	-	-	-	3.07	3.75	**3.23**	0.59	0.00	0.24

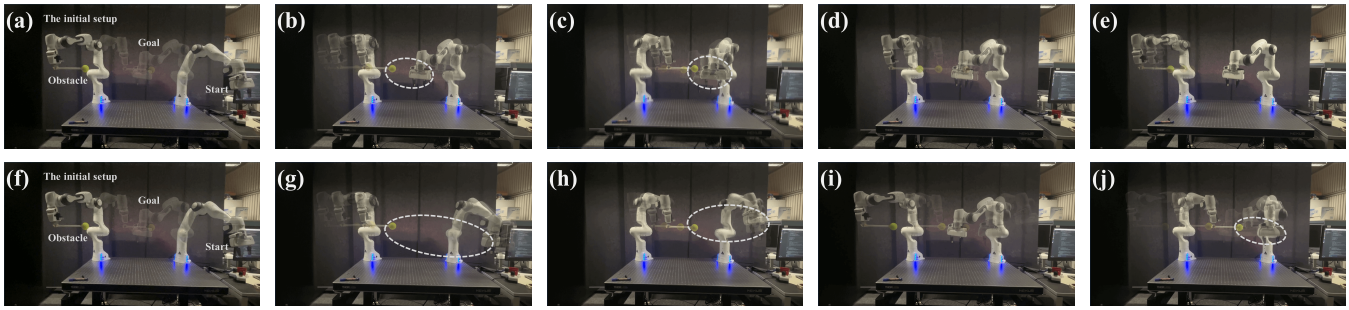


Fig. 5: CDF-TVCBF-QP: snapshots of the collision-avoidance task with different velocities of obstacles. (a) The obstacle moves slowly at $0.05\ \mathrm{m/s}$. (b) The robot reaches the goal early, as it is aware of the obstacle's dynamics. (c) Due to the slow movement of the obstacle, the robot arm stays nearby. (d)-(e) The robot returns to the goal and maintains its position as the obstacle moves far away. (f) The obstacle moves slowly at $0.15\ \mathrm{m/s}$. (g)-(h) The robot behaves in a safety-aware manner and stays farther away due to the high velocity of the obstacle. (i)-(j) The robot arm reaches the goal while prioritizing safety.

3) **Path Length**: The ℓ^2 norm of the whole-body trajectory length in meters.

We conduct experiments across three 2D scenarios (S1, S2, and S3) and three 7D scenarios (C1, C2, and C3), each comprising 100 randomly generated setups. These experiments evaluate the controller performance in dynamic environments using metrics that capture both effectiveness (success rate) and efficiency (time-to-reach and path length). In scenario S1, a single moving obstacle is randomly generated within the region $x \in [1.5, 3.5]\ \mathrm{m}$ and $y \in [1.5, 3.5]\ \mathrm{m}$. The initial and goal states of the planar arm are $q(0) = (2.5, 0.5)\,\mathrm{rad}$ and $q_g = (-2.7, 0.5)\,\mathrm{rad}$. The obstacle velocity is randomly selected within $(-4, -2)\,\mathrm{m/s}$ along the x-axis. In scenario S2, a second obstacle is added, randomly generated within $x \in [1.5, 3.0]\ \mathrm{m}$ and $y \in [-3.0, -1.5]\ \mathrm{m}$. The velocity range of the obstacle is increased, $v_1 \in [3.0, 5.0]\ \mathrm{m/s}$ along the x-axis and $v_2 \in [-5.0, -3.0]\ \mathrm{m/s}$ along the y-axis, to challenge approaches. In scenario S3, a third obstacle is introduced, randomly generated from the region $x \in [-3.0, -1.5]\ \mathrm{m}$ and $y \in [-3.0, -1.5]\ \mathrm{m}$ and its velocity is along the x-axis randomly generated between $v_3 \in [1.0, 3.0]\ \mathrm{m/s}$ The experimental setup is illustrated in Fig.4, and the results are summarized in Table II.

The results show that our approaches outperform others in both success rate and time-to-reach metrics. The observed discrepancies in path length arise because the average path length only includes successful cases, leading to simpler cases being overrepresented in SH-based methods. The SDF-CBF-QP method fails in almost all cases due to its time-invariant safe set, which does not account for dynamic obstacles. Overall, our method demonstrates superior performance in highly dynamic environments in 2D scenarios.

For the 7D scenarios, we evaluate the approaches using a 7-axis Franka robot arm, as shown in Fig. 4b. In scenario C1, two obstacles are randomly generated within $x \in [0.3, 0.6]\ \mathrm{m}$, $y \in [0.6, 0.9]\ \mathrm{m}$ and $z \in [0.5, 0.7]\ \mathrm{m}$, with velocities along the y-axis randomly selected between $-0.8\,\mathrm{m/s}$ and $-0.5\,\mathrm{m/s}$. The initial state of the robot arm is $q(0) = (-1.09, 0.35, -0.32, -1.70, 0.18, 2.05, -0.20)\,\mathrm{rad}$ and the task is to rotate the first joint to 1.09 without collisions. In scenario C2, six obstacles are randomly generated within the same region, with velocities increasing across obstacles. The first obstacle moves at a velocity range of $[-0.2, -0.1]\ \mathrm{m/s}$, the fifth at $[-0.9, -1.0]\ \mathrm{m/s}$, and the sixth covers the full range $[-0.9, -0.1]\ \mathrm{m/s}$. The robot's task is to maintain its pose at $(0.09, 0.35, -0.32, -1.70, 0.18, 2.05, -0.20)\,\mathrm{rad}$, avoid obstacles, and return to this pose. In scenario C3, the task remains the same as in C1, but the obstacle setup is identical to C2.

The results indicate that our method consistently outperforms others across metrics. However, the path length is not a primary advantage, as our method prioritizes safety, resulting in slightly longer paths for simpler tasks, such as avoiding

slow-moving obstacles. In contrast, baselines tend to take more aggressive trajectories, occasionally succeeding with shorter path lengths. While our methods account for obstacle velocities, failure can still occur due to physical limitations.

E. Real-world Experiments

To validate our approach, we conducted experiments on a 7-axis Franka robot. In our experimental setup, the right robot arm performed a goal-reaching task, moving from an initial joint configuration to a target one while accounting for a dynamic obstacle moving at a predefined speed. The left robot arm was used to simulate the moving obstacle by holding a stick with a ball, which moved at a constant velocity to ensure reproducible results. We tested obstacle velocities of 0.05 m/s and 0.15 m/s, observing distinct robot behaviors for each velocity, as shown in Fig. 5. Our approach demonstrates effective adaptive collision-avoidance behavior in response to moving obstacles. When the obstacle moves slower (top), the robot efficiently reaches its target while maintaining a safe but close proximity to the moving object. Once the obstacle moves farther away, the robot returns to the goal and stabilizes its position.

V. Conclusions

In this paper, we propose two formulations of safety-critical controllers, designed for dynamic whole-body collision avoidance and reaching tasks for robotic manipulators. The task space and configuration space of the manipulator are bridged through differentiable distance fields, SDFs and CDFs. We introduce a method to compute the configuration velocity of obstacles or targets from the task space, which is crucial for designing TVCBFs and TVCLFs constraints in the configuration space. Extensive numerical simulations, benchmarking against state-of-the-art methods, and real-world experiments demonstrate the efficacy of our approach. The gradient information in configuration space could be further explored in future work, such as for constructing convex, collision-free sets.

References

[1] M. Koptev, N. Figueroa, and A. Billard, "Reactive collision-free motion generation in joint space via dynamical systems and sampling-based mpc," *The International Journal of Robotics Research*, p. 02783649241246557, 2024.

[2] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[3] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[4] M. Spahn, B. Brito, and J. Alonso-Mora, "Coupled mobile manipulation via trajectory optimization with free space decomposition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 759–12 765.

[5] X. Chi, Z. Liu, J. Huang, F. Hong, and H. Su, "Optimization-based motion planning for autonomous parking considering dynamic obstacle: A hierarchical framework," in *2022 34th Chinese Control and Decision Conference (CCDC)*, 2022, pp. 6229–6234.

[6] T. S. Lembono and S. Calinon, "Probabilistic iterative lqr for short time horizon mpc," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 579–585.

[7] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[8] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, "Vp-sto: Via-point-based stochastic trajectory optimization for reactive robot behavior," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 125–10 131.

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

[10] S. S. Mirrazavi Salehian, N. Figueroa, and A. Billard, "A unified framework for coordinated multi-arm motion planning," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1205–1232, 2018.

[11] Y. Li, X. Chi, A. Razmjoo, and S. Calinon, "Configuration space distance fields for manipulation planning," *arXiv preprint arXiv:2406.01137*, 2024.

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

[13] Y. Li, Y. Zhang, A. Razmjoo, and S. Calinon, "Representing robot geometry as distance fields: Applications to whole-body manipulation," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2024, pp. 15 351–15 357.

[14] J. Huang, X. Chi, Z. Liu, and H. Su, "Whole-body dynamic collision avoidance with time-varying control barrier functions," in *2024 36th Chinese Control and Decision Conference (CCDC)*. IEEE, 2024, pp. 5149–5154.

[15] J. Wang, T. Zhang, Q. Zhang, C. Zeng, J. Yu, C. Xu, L. Xu, and F. Gao, "Implicit swept volume sdf: Enabling continuous collision-free trajectory generation for arbitrary shapes," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–14, 2024.

[16] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[17] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3882–3889.

[18] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 954–10 961, 2022.

[19] B. Dai, R. Khorrambakht, P. Krishnamurthy, V. Goncalves, A. Tzes, and F. Khorrami, "Safe navigation and obstacle avoidance using differentiable optimization based control barrier functions," *IEEE Robotics and Automation Letters*, 2023.

[20] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018.

[21] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, 2019, pp. 3420–3431.

[22] Y. Xian, Y. Sun, X. Luo, Y. Hu, L. Zou, D. T. M. Chan, D. Y. C. Chan, and Z. Li, "Task automated stereotactic brain biopsy robotic system with clf-cbf-based safety-critical neuronavigation," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2025.

[23] B. Dai, R. Khorrambakht, P. Krishnamurthy, and F. Khorrami, "Differentiable optimization based time-varying control barrier functions for dynamic obstacle avoidance," *arXiv preprint arXiv:2309.17226*, 2023.

[24] J. Huang, Z. Liu, J. Zeng, X. Chi, and H. Su, "Obstacle avoidance for unicycle-modelled mobile robots with time-varying control barrier functions," in *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023, pp. 1–6.

[25] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.