

# Smoothed Distance Functions for Direct Optimal Control of Contact-Rich Systems

Christian Dietz<sup>1,2</sup>, Sebastian Albrecht<sup>1</sup>, Armin Nurkanović<sup>2</sup>, Moritz Diehl<sup>2,3</sup>

**Abstract**—This paper discusses a smoothing method for a signed distance function (SDF) between polytopes that is applicable in direct optimal control methods for contact-rich systems. It is well known that the Euclidean or similar SDF are nondifferentiable with respect to degrees of freedom for rigid bodies that do not have sufficiently smooth surfaces. We consider a growth distance for polytopes modeled as a parametric linear program and give a thorough proof that smoothing this SDF by solving perturbed optimality conditions results in well-defined functions. In previous works, exact optimality conditions of the SDF were embedded directly into an optimal control problem (OCP). In contrast, smoothing the SDF allows for external evaluation and thereby smaller instances of OCP, distributing the computational effort more equally among function evaluations and the load on numerical solvers. We discuss an implementation of the smooth SDF in the `CasADi` toolbox and compare the performance to the equivalent direct embedding into the OCP. A planar peg-in-hole task is considered for which an ensemble of trajectories that share the same control input is optimized simultaneously.

## I. INTRODUCTION

Contact-rich systems stemming from robotic manipulation or locomotion can be modeled by complementarity Lagrangian systems (CLS) with state jumps [1]. Since solution trajectories of CLS exhibit severe nonsmoothness, it is not straightforward to extend standard discretization techniques used in direct optimal control of smooth dynamics to CLS. When considering a dynamical system within an optimal control framework, the derivatives of the state trajectory with respect to the controls, referred to as sensitivities, are used to characterize an optimal solution. Consequently, one would like to ensure that the sensitivities of the discretized system converge to the correct values of the continuous-time system.

A large body of work that deals with discretization of CLS relies on implicit Euler time-stepping schemes, which integrate nonsmooth systems over fixed time-steps. While these methods are widely used, the sensitivities of the discretized system do not converge to their correct values, which can hinder convergence in optimal control applications [2, 3]. In contrast, event-based discretization schemes tailored for the use within optimal control methods allow to detect switches exactly in time and, thereby, recover correct sensitivities [4, 5] at the cost of additional complexity within the discretization. Since it is challenging to solve optimization problems that are subject to complementarity constraints,

solution determination of the above mentioned formulations is often limited to offline computations. However, some works aim at devising such schemes for the real-time control of contact-rich systems, usually by introducing additional approximations to time-stepping discretizations. These methods capture the original structure of the CLS very loosely but still can yield useful control inputs for some applications [6, 7].

All of the above introduced optimal control methods applied to contact-rich systems depend on signed distance functions (SDF) that encode information whether two objects are in contact and, if so, in which direction forces and impulses are acting. Most works that consider this topic assume that the SDF are sufficiently differentiable. At first, this limits one to objects such as spheres, ellipsoids and planes. Distances between arbitrary convex shapes can be represented by a so-called growth distance [8], which determines the smallest scaling factor by which the two shapes have to be altered to have a common intersection point. For the case of polytopic shapes, this can be formulated as a linear program and, thus, results in a simpler SDF representation compared to using a signed Euclidean distance, which can be represented by an optimization problem with a quadratic equality constraint [9]. The nonsmoothness of the SDF between polytopes embodies itself in the fact that the optimization-based growth distance has nonunique primal or dual variables for some system configurations. In [10], the optimality conditions of a Euclidean SDF were embedded directly into a time-stepping discretization and it was shown that the coupling of dynamics and distance evaluation achieves unique solutions. In [11], this was done similarly with a growth distance. In [12], we showed that embedding optimality conditions of a growth distance into a CLS readily results in a well-defined continuous-time system and then derived an event-based discretization scheme that allows exact switch detection.

Adding primal and dual variables of the optimization-based distance formulations to the optimal control problems (OCP) results in very large systems that become increasingly computationally expensive to solve for a large amount of considered objects or large optimization horizons. Therefore, another approach to deal with nonsmooth SDF is to construct a smooth approximation as this allows one to evaluate distances and corresponding derivatives externally from a numerical OCP solver since the latter only relies on an interface that provides respective values. Resulting OCP for which external distance evaluation is employed are then much smaller compared to directly embedding the optimality conditions of the distance problem. Some different approaches exist that

<sup>1</sup>Autonomous Systems and Control, Siemens AG, Germany

<sup>2</sup>Department of Microsystems Engineering (IMTEK), University of Freiburg, Germany

<sup>3</sup>Department of Mathematics, University of Freiburg, Germany  
Correspondent: dietz.christian@siemens.com

aim to derive smoothed distances between polytopic bodies. In [13], the authors approximate polytopes geometrically by a number of bounding spheres, resulting in a once continuously differentiable distance expression that, however, does not have higher-order derivatives. In [14] randomized smoothing is utilized to determine derivatives of nonsmooth polytopic distances. While randomized smoothing is simple in application, resulting derivatives are subject to noise and lowering it by averaging a large amount of evaluations can be computationally expensive. Work closely related to this paper is given by [15, 16]. In [15] the authors first propose an efficient implementation of an interior-point method tailored to growth distance problems. They then apply the implicit function theorem (IFT) to determine derivatives of the distance problem, neglecting that the assumptions of the IFT fail to hold in nondifferentiable configurations. In [16] this issue is acknowledged, and it is stated that solving a perturbed Karush-Kuhn-Tucker (KKT) system results in smoothed gradients. However, it is not proven that this indeed results in well-defined derivatives and no application in direct optimal control is given.

#### A. Contribution

We give a proof that the smoothed growth distance between polytopes obtained from perturbed KKT conditions is well-defined,  $C^\infty$ , and derivatives of arbitrary order are obtained by repetitive application of the IFT. Furthermore, we propose an approximation to the gradient of this smooth SDF that can be utilized as contact normal in direct optimal control, which results in cheaper numerical evaluation compared to the exact gradient. We detail on an implementation of the SDF and corresponding derivatives in the `CasADi` toolbox [17] and compare the performance of the implementation to the equivalent direct embedding of SDF optimality conditions on a planar peg-in-hole problem.

#### B. Structure of the paper

The remainder of the paper is structured as follows. In Section II the well-posedness of the smooth SDF and corresponding derivatives are discussed. A discrete-time OCP for contact-rich systems used for numerical evaluation is introduced in Section III. In Section IV, we discuss implementation aspects of the smooth SDF and in Section V we conduct numerical experiments.

#### C. Notation

To stack vectors, we compactly write  $(x_1, \dots, x_m) = [x_1^\top, \dots, x_m^\top]^\top$  for  $x_1 \in \mathbb{R}^{n_1}, \dots, x_m \in \mathbb{R}^{n_m}$ . Furthermore,  $\mathbf{0} = (0, \dots, 0)$  and  $\mathbf{1} = (1, \dots, 1)$  are the vectors with all zeros or all ones, respectively, where the corresponding dimension becomes clear from the context.

## II. SMOOTHED DISTANCE FUNCTIONS FOR POLYTOPIC BODIES

#### A. Smoothed Scaling Distance Problem

Consider two planar polytopes given through a halfspace representation

$$\mathcal{P}_1 = \{p \in \mathbb{R}^2 \mid G_1 p \leq h_1\}, \quad \mathcal{P}_2 = \{p \in \mathbb{R}^2 \mid G_2 p \leq h_2\},$$

where  $G_1 \in \mathbb{R}^{n_{1,h} \times 2}$ ,  $h_1 \in \mathbb{R}^{n_{1,h}}$ ,  $G_2 \in \mathbb{R}^{n_{2,h} \times 2}$ ,  $h_2 \in \mathbb{R}^{n_{2,h}}$ . We make the following assumptions on the polytopes that are without loss of practical applicability when modelling robotic manipulation and locomotion problems.

**Assumption 1.** *Let the halfspace representations for  $\mathcal{P}_1$  and  $\mathcal{P}_2$  be constructed such that the rows of  $G_1$  and  $G_2$  are vectors with norm unity. Additionally, let the halfspace representations be nonredundant, i.e., no inequality can be removed without altering the shape of the polytopes.*

**Assumption 2.** *Let  $\mathcal{P}_1$  and  $\mathcal{P}_2$  be bounded convex polytopes that have nonempty interior and let the origin be contained in the interior. Furthermore, let the corresponding halfspace representations be constructed such that for given mass distributions, the center of mass (CoM) of  $\mathcal{P}_1$  and  $\mathcal{P}_2$  coincides with the origin.*

We associate degrees of freedom (DoF) with the bodies, namely  $\rho_i \in \mathbb{R}^2$  for the position of their CoM and  $\xi_i \in \mathbb{R}$  for their orientation, both with respect to a world coordinate frame. All DoF of the bodies are captured in the system configuration variable  $q = (\rho_1, \xi_1, \rho_2, \xi_2) \in \mathbb{R}^6$ . We further consider a linear scale factor  $\alpha \geq 0$  and obtain transformed polytopes as

$$\begin{aligned} \mathcal{B}_1(q, \alpha) &= \{p \in \mathbb{R}^2 \mid G_1 R(\xi_1)^\top (p - \rho_1) \leq \alpha h_1\}, \\ \mathcal{B}_2(q, \alpha) &= \{p \in \mathbb{R}^2 \mid G_2 R(\xi_2)^\top (p - \rho_2) \leq \alpha h_2\}, \end{aligned}$$

where  $R(\cdot) \in \mathbb{R}^{2 \times 2}$  is the two-dimensional rotation matrix.

The growth distance introduced in [8] now determines the smallest scaling factor for a fixed  $q$  such that  $\mathcal{B}_1(q, \alpha)$  and  $\mathcal{B}_2(q, \alpha)$  intersect at a common point  $p$ . Since in this growth distance an optimal scale factor of  $\alpha = 1$  indicates that the unscaled polytopes touch in a common point,  $\alpha - 1$  corresponds to a signed distance. The growth SDF is expressed as a parametric linear program given by

$$\begin{aligned} \Phi_0(q) &= \min_{p, \alpha} \quad \alpha - 1 \\ \text{s.t.} \quad & G_1 R(\xi_1)^\top (p - \rho_1) \leq \alpha h_1, \\ & G_2 R(\xi_2)^\top (p - \rho_2) \leq \alpha h_2. \end{aligned}$$

We denote this compactly as

$$\Phi_0(q) = \min_z \quad c^\top z - 1 \tag{1a}$$

$$\text{s.t.} \quad A(q)z \leq b(q), \tag{1b}$$

with

$$A(q) = \begin{pmatrix} G_1 R(\xi_1)^\top & -h_1 \\ G_2 R(\xi_2)^\top & -h_2 \end{pmatrix}, \quad b(q) = \begin{pmatrix} G_1 R(\xi_1)^\top \rho_1 \\ G_2 R(\xi_2)^\top \rho_2 \end{pmatrix},$$

as well as  $c = (0, 0, 1)$  and  $z = (p, \alpha)$ . The perturbed KKT conditions of (1) are given by

$$\begin{aligned} A(q)^\top \lambda &= -c, \\ y &= b(q) - A(q)z, \\ \Lambda y &= \mathbf{1}\tau, \\ \lambda &> \mathbf{0}, y > \mathbf{0}, \end{aligned} \tag{2}$$

with homotopy parameter  $\tau > 0$ . Vector  $y \in \mathbb{R}^{n_g}$  denotes the slacks for the inequality constraints (1b),  $\lambda \in \mathbb{R}^{n_g}$  denotes the Lagrange multipliers and  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{n_g})$  where  $n_g = n_{1,h} + n_{2,h}$ . For  $\tau = 0$  and nonstrict inequality constraints in (2) one recovers the exact KKT conditions. We now denote the equality conditions of (2) compactly by

$$F_\tau(\gamma; q) = \begin{pmatrix} c + A(q)^\top \lambda \\ y - b(q) + A(q)z \\ \Lambda y - \mathbf{1}\tau \end{pmatrix} = \mathbf{0},$$

where  $\gamma = (z, \lambda, y) \in \mathbb{R}^{n_\gamma}$ ,  $n_\gamma = 3 + 2n_g$ . In the following, we will show that for any given  $q$  and  $\tau$  such a  $\gamma$  solving (2) is unique.

**Lemma 1.** *Let Assumptions 1 and 2 hold. Then the inequality constraint matrix  $A(q)$  has full rank for all  $q \in \mathbb{R}^6$ .*

This can be proven constructively and is postponed to Appendix VII-A in favor of better readability. Given Lemma 1, we can prove the following proposition which is the main result that ensures well-posedness of the here considered smooth SDF.

**Proposition 1.** *Let Assumptions 1 and 2 hold. Then the central path  $\gamma_\tau(q) = (z_\tau(q), \lambda_\tau(q), y_\tau(q))$  of the distance problem (1), i.e., the solution of the perturbed KKT system (2) for any  $\tau > 0$  and  $q \in \mathbb{R}^6$ , exists and is unique.*

*Proof:* For clearer notation in this proof, we assume that  $\tau$  and  $q$  are fixed and dependencies on these terms are dropped. We consider the barrier problem associated with the perturbed KKT system (2) given by

$$\min_z f_\tau(z) \quad (3)$$

with

$$f_\tau(z) = c^\top z - \tau \sum_{i=1}^m \log(b_i - a_i^\top z), \quad (4)$$

where  $a_i \in \mathbb{R}^3$  denote the rows of  $A$  and  $m = n_{1,h} + n_{2,h}$  is the number of halfspaces used to construct  $\mathcal{P}_1$  and  $\mathcal{P}_2$ . The corresponding Hessian is obtained as

$$\nabla^2 f_\tau(z) = \sum_{i=1}^m \frac{\tau}{(b_i - a_i^\top z)^2} a_i a_i^\top.$$

Thus, the Hessian is positive definite if and only if  $A$  has full rank, which is ensured by Lemma 1. Consequently,  $f_\tau(z)$  is strictly convex and admits at most one minimum.

Since we assumed  $\mathcal{P}_1$  to be bounded, the sets

$$\{z \mid Az \leq b, \alpha \leq \alpha_0\} \quad (5)$$

for given  $\alpha_0 > 0$  are also bounded, as (5) implies that  $z$  is element of a scaled version of  $\mathcal{P}_1$ . Now, if  $\|z\| \rightarrow \infty$  with  $\alpha \rightarrow \infty$  then  $\|f_\tau(x)\| \rightarrow \infty$  as  $c^\top z = \alpha$  approaches infinity faster than any negative logarithm term in (4) can approach negative infinity. Thus, for every  $\gamma_0 \in \mathbb{R}$  there exists an  $\alpha_0 > 0$  such that

$$\{z \mid f_\tau(z) \leq \gamma_0\} \subset \{z \mid Az \leq b, \alpha \leq \alpha_0\}, \quad (6)$$

i.e., the level sets of the barrier function are bounded. Furthermore, they are closed as  $f_\tau(z)$  is continuous. Thus, existence of the minimizer of (3) follows from Weierstrass' Theorem, see, e.g., [18, Proposition A.8].

The sufficient and necessary optimality condition for the minimizer of (3) is given by

$$\nabla f_\tau(z) = c + \sum_{i=1}^m \frac{\tau}{b_i - a_i^\top z} a_i = \mathbf{0}. \quad (7)$$

By defining  $y_i = b_i - a_i^\top z$  and  $\lambda_i = \frac{\tau}{b_i - a_i^\top z}$ , we determined a solution to the perturbed KKT conditions (2). Assuming two distinct solutions  $(z_1, \lambda_1, y_1)$  and  $(z_2, \lambda_2, y_2)$  for the perturbed KKT conditions (2), with  $\|\lambda_1 - \lambda_2\| > 0$  or  $\|y_1 - y_2\| > 0$ , implies  $\|z_1 - z_2\| > 0$ , which contradicts the proof that (4) has a unique minimizer. Consequently, the perturbed KKT system (2) has a unique solution. ■

Proposition 1 implies that the distance function defined by

$$\Phi_\tau(q) = \{\alpha - 1 \mid F_\tau(\gamma; q) = \mathbf{0}, \lambda > \mathbf{0}, y > \mathbf{0}\}, \quad (8)$$

is well-defined for  $\tau > 0$ . In particular, it provides an upper bound for the exact SDF, i.e.,  $\Phi_0(q) < \Phi_\tau(q)$ . This is a direct consequence resulting of application of an interior-point method to a linear program. As optimal solutions of linear programs with a nonzero objective function always lie on the boundary of the feasible set, while solutions of the perturbed KKT conditions (2) lie in the interior of the feasible set, the corresponding objective value of the latter has to be larger than the optimal objective value of the linear program. In nonsmooth dynamics, the feasible configuration space is characterized by all configurations with nonnegative SDF [1]. Due to the upper bound for the exact SDF it holds

$$\{q \in \mathbb{R}^{n_q} \mid \Phi_0(q) \geq 0\} \subset \{q \in \mathbb{R}^{n_q} \mid \Phi_\tau(q) \geq 0\}.$$

Thus, it is ensured that trajectories in the feasible region for the exact SDF (1) are also feasible for the smoothed version. This is of particular significance when considering assembly problems, which result in confined feasible regions, as in the example problem considered in Section V.

**Remark 1.** *The derivations above can be adapted to the case where  $\mathcal{P}_2$  is unbounded or a set with empty interior. The unbounded case follows directly as we only relied on assumptions on  $\mathcal{P}_1$  in the proof of Lemma 1 and Proposition 1. If  $\mathcal{P}_2$  has empty interior, this has to be reflected appropriately by equality conditions in (1), as else-wise the barrier function (4) is ill-defined. Everything else follows analogously. This observation is useful when modeling assembly problems, where the system state  $q$  only contains the DoF of one body and one is interested with its interaction with the environment. Then one can, e.g., model the surface of a table by a single halfspace.*

#### B. Derivatives of the Smoothed Distance Function

The uniqueness of  $\gamma_\tau(q)$  for a fixed  $q$  implies that the Jacobian of the equality conditions of the perturbed KKT conditions, i.e.,  $D_\gamma F_\tau(\gamma_\tau(q); q)$ , is invertible at a solution.

Therefore, the assumptions of the implicit function theorem are fulfilled, and we obtain the derivatives  $D_q\gamma_\tau(q)$  by

$$D_q\gamma_\tau(q) = -(D_\gamma F_\tau(\gamma_\tau(q); q))^{-1} D_q F_\tau(\gamma_\tau(q); q). \quad (9)$$

In particular, we can retrieve the gradient of the smoothed distance function from the corresponding row in the Jacobian, i.e, we have

$$\nabla_q \Phi_\tau(q) = D_q\gamma_\tau(q)^\top s,$$

where  $s = (0, 0, 1, 0, \dots, 0)^\top \in \mathbb{R}^{n_\gamma}$  is a selection vector.

When using a SDF within a contact-rich control setting, one requires a contact normal vector  $n(q) \in \mathbb{R}^{n_q}$  that determines the direction of forces and impulses. Classically, given a smooth SDF, one selects the normalized gradient of the SDF as contact normal. When optimizing such a problem with Newton methods and exact Hessian information, we thus need to provide third-order derivatives of the smooth SDF (8). These can be calculated by repetitively applying the chain rule to (9), which however results in increasingly complex expressions. To this end, we observe that directional derivatives of the exact SDF (1) can be obtained as

$$\partial_d \Phi_0(q) = \min_{z \in \bar{Z}(q)} \max_{\lambda \in \bar{\Lambda}(q)} d^\top \nabla_q g(z, q) \lambda, \quad (10)$$

for direction  $d \in \mathbb{R}^{n_q}$ , see [19]. Here,  $\bar{Z}(q)$  denotes the set of all primal optimal solutions to (1),  $\bar{\Lambda}(q)$  denotes the set of all corresponding dual optimal solutions and  $g(z, q) = A(q)z - b(q)$  captures the inequality constraints of the distance problem. Consequently, in configurations where the exact distance function admits unique primal-dual optimal variables it is classically differentiable with gradient equal to  $\nabla_q g(z, q) \lambda$ . This motivates us to consider the smooth SDF together with the contact normal given through

$$n_\tau(q) = \frac{\nabla_q g(z_\tau(q), q) \lambda_\tau(q)}{\|\nabla_q g(z_\tau(q), q) \lambda_\tau(q)\|_2}. \quad (11)$$

Evaluating Hessian information for  $n_\tau(q)$  now only requires differentiating through the right-hand side of (9) once instead of twice as needed for the exact gradient. Numerically, the Hessian evaluation for  $n_\tau(q)$  was accomplished three times faster than doing the same for the normalized gradient, which motivates introducing this approximation. In particular, since the central path converges to an optimal solution of (1), cf. [18, Section 4.1], the contact normal  $n_\tau(q)$  converges to the exact gradient of  $\Phi_0(q)$  for system configurations where the exact SDF is differentiable, justifying this approximation.

### III. ENSEMBLE CONTACT-IMPLICIT TRAJECTORY OPTIMIZATION

In this section, we introduce a discrete-time OCP with contact-rich dynamics that will be used for numerical evaluation of the considered SDF. To this end, in Section III-A continuous-time contact-rich dynamics are introduced and in Section III-B a corresponding discretization is considered. Finally, in Section III-C the discrete-time OCP is formulated and we detail on the direct embedding of perturbed optimality conditions of the SDF that can be alternatively used to external SDF evaluation.

#### A. Continuous-time contact-rich dynamics

Continuous-time dynamics of contact-rich systems with frictionless unilateral contacts combine standard Newton-Euler equations with complementarity conditions that determine active contact forces. Additionally, one needs to consider multi impact laws that encode how system velocities jump if a contact is established with negative normal velocity, choosing a multi impact law is a modeling decision [1]. The considered continuous-time dynamics are given by

$$\dot{q} = \nu, \quad (12a)$$

$$M\dot{\nu} = u + \sum_{i=1}^{n_d} n_{\tau,i}(q) \lambda_{n,i}, \quad (12b)$$

$$0 \leq \Phi_{\tau,i}(q) \perp \lambda_{n,i} \geq 0, \quad i = 1, \dots, n_d, \quad (12c)$$

$$\text{Multi impact law,} \quad (12d)$$

where  $\nu \in \mathbb{R}^{n_v}$  is the system velocity,  $M \in \mathbb{R}^{n_q \times n_q}$  is the positive definite and diagonal inertia matrix and  $u \in \mathbb{R}^{n_u}$  denotes control forces. The system state is denoted compactly by  $x = (q, \nu)$ . We consider  $n_d$  object pairs with smooth SDF  $\Phi_{\tau,i}$  (8), corresponding contact normals  $n_{\tau,i}$  (11) and contact force magnitudes  $\lambda_{n,i} \in \mathbb{R}$ .

#### B. Time-stepping for an ensemble of trajectories

As noted in the introduction, discretizing the nonsmooth dynamics (12) is not straightforward and various approaches exist. Furthermore, as nonsmooth dynamics include complementarity constraints (12c), corresponding discretized OCP problems naturally result in Mathematical Programs with Complementarity Constraints (MPCC). This class of optimization problems is challenging to solve as standard optimality conditions can not be applied. To this end, in direct optimal control one commonly utilizes a homotopy strategy and replaces complementarity constraints  $0 \leq a \perp b \geq 0$  by the relaxed conditions  $a \leq 0, b \leq 0, a^\top b \leq \sigma$ , where  $\sigma > 0$  is the homotopy parameter that is sequentially driven to zero.

High-accuracy event-based discretization schemes [4,5] for nonsmooth OCP provide numerical sensitivities that converge to the true values for  $\sigma \rightarrow 0$  through introducing additional conditions to standard Runge-Kutta discretizations of (12), which ensure that dynamic switches do not occur within an integration step. In [2,20] it is shown that time-stepping discretizations [21,22], which usually solely apply implicit Euler methods to (12), fail to provide correct sensitivities for fixed integration step-sizes  $h > 0$  if the homotopy parameter tends to zero  $\sigma \rightarrow 0$ . On the other hand, if the homotopy parameter is in the magnitude of the step-size  $\sigma = o(h)$ , numerical sensitivities still provide correct information. Discrete-time OCP obtained by utilizing time-stepping discretizations are less complex compared to ones obtained by using accurate event-based discretizations. Therefore, it is reasonable to solve a discrete-time OCP parameterized by time-stepping starting from a trivial initial guess and then switch to an event-based discretization during the homotopy procedure for smaller  $\sigma$ , thereby, providing a good initial guess for the latter and, thus, facilitating

convergence. We consider the problem of finding a solution to a time-stepping discretized OCP starting from a trivial initial guess to compare the performance of the external SDF evaluation as outlined in Section IV to the equivalent direct embedding of (2), which is further detailed in the following.

We consider optimal control problems that aim at providing solutions to manipulation problems which can be reliably executed on real-world systems. To accomplish this, multiple trajectories are simulated for the same control input simultaneously while randomizing initial values. Additionally, one could randomize model parameters for each trajectory. As a result, resulting controls become robust to model errors, facilitating application in real-world systems [23].

To this end, an ensemble of  $n_e \in \mathbb{N}$  trajectories is considered. Approximations of the differential and algebraic states of (12) at time  $t_k = kh$  are denoted by  $q_k^j, \nu_k^j, \lambda_{n,i,k}^j, u_k$  for  $j = 1, \dots, n_e, k = 0, \dots, N, i = 1, \dots, n_d$ , where  $h > 0$  is the step-size and  $N \in \mathbb{N}$  is the considered simulation horizon. For given initial states  $x_0^j = (q_0^j, \nu_0^j), j = 1, \dots, n_e$ , a time-stepping discretization of (12) is given by

$$q_{k+1}^j = q_k^j + h\nu_{k+1}^j, \quad (13a)$$

$$\nu_{k+1}^j = \nu_k^j + hM^{-1}(u_k + \sum_{i=1}^{n_d} n_{\tau,i}(q_{k+1}^j)\lambda_{n,i,k}^j), \quad (13b)$$

$$0 \leq \Phi_{\tau,i}(q_{k+1}^j) \perp \lambda_{n,i,k}^j \geq 0, \quad i = 1, \dots, n_d, \quad (13c)$$

$$(n_{\tau,i}(q_{k+1}^j)^\top \nu_{k+1}^j)^2 \perp \lambda_{n,i,k}^j, \quad i = 1, \dots, n_d. \quad (13d)$$

The force complementarity is considered both at position- [21] and velocity-level [22]. Some works only consider the position-level, e.g., [11]. However, we observed that if the contact normal changes rapidly across two time-steps, e.g., the contact point slides across a polytope's vertex, discarding (13d) results in unphysical behavior in which two bodies are repelled by a large contact force.

### C. Ensemble contact-implicit trajectory optimization

A discrete-time OCP based on the discretized dynamics (13) is given by

$$\begin{aligned} \min_{\mathbf{x}, \lambda_n, \mathbf{u}} \quad & \sum_{j=1}^{n_e} \sum_{k=0}^{N-1} L_r(x_k^j, u_k) + \sum_{j=1}^{n_e} L_t(x_N^j) \\ \text{s.t.} \quad & x_0^j = \bar{x}_0 + w^j, \\ & \text{Discrete-time dynamics (13),} \\ & g_p(x_k^j, u_k) \leq \mathbf{0}, \\ & \text{for } j = 1, \dots, n_e, \quad k = 0, \dots, N-1, \end{aligned} \quad (14)$$

where  $L_r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$  denotes the running cost,  $L_t : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$  the terminal cost and  $g_p : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$  are the path constraints. The initial state of each discrete-time trajectory is determined by a nominal initial state  $\bar{x}_0 \in \mathbb{R}^{n_x}$  and a random variable  $w^j \in W$ , chosen uniformly from some set  $W \subset \mathbb{R}^{n_x}$ . The optimization variables are collected by  $\mathbf{x} = (x_0^1, \dots, x_N^1, \dots, x_0^{n_e}, \dots)$ ,  $\lambda_n = (\lambda_{n,1,0}^1, \dots, \lambda_{n,1,N-1}^1, \dots, \lambda_{n,n_d,0}^{n_e}, \dots)$  and  $\mathbf{u} = (u_0, \dots, u_{N-1})$ .

Since the discrete-time OCP (14) contains through the discrete dynamics (13) the SDF  $\Phi_{\tau,i}(q_{k+1}^j)$  and corresponding contact normal  $n_{\tau,i}(q_{k+1}^j)$ , solving the OCP requires interfacing a code that allows evaluation of these terms. Alternatively, one can add the perturbed KKT conditions (2) and corresponding primal-dual variables directly to the OCP (14). This significantly increases the size of the optimization problem as the number of optimization variables and constraints introduced through the perturbed KKT conditions of the SDF usually dominates the number of optimization variables and constraints introduced through the discrete-time dynamics (13).

## IV. DISTANCE FUNCTION IMPLEMENTATION

In the following, we discuss implementation aspects of the smooth SDF (8) in the CasADi toolbox [17]. The SDF is implemented by using the `Callback` class in CasADi to wrap the solving operations and derivative evaluations. The `Callback` class provides a public API to the `FunctionInternal` class, which is the base class for functions in CasADi. By overwriting a few virtual methods, a `Callback` object can be included in an optimization problem like any other CasADi function. In particular, we provide code for the nominal function evaluation itself, for the evaluation of the corresponding Jacobian and adjoint derivatives, and for the evaluation of Hessian information. Since in contrast to native functions, calls to `Callback` functions are not parallelized by CasADi, we consider a batched version that takes all discretized system configurations at once as input and returns all corresponding distances and contact normals as defined in (11). This allows us to utilize parallel computing inside the `Callback` function. To solve the perturbed KKT conditions (2), we first solve the corresponding parametric LP (1) by using HPIPM [24], a high-performance interior-point solver, through its Python interface. This solving operation is parallelized for  $Nn_e$  system configurations by making calls in parallel through utilization of the multiprocessing library. Given fully converged solutions to the distance problem (1), we then use the `rootfinder` class of CasADi to solve the perturbed KKT conditions (2) for a fixed homotopy parameter, where the `rootfinder` performs Newton iterations with linesearch and is initialized with the solution determined by HPIPM. The calls to `rootfinder` are parallelized by using the map functionality in CasADi, which allows us to evaluate multiple calls to a function or solver in parallel.

The derivatives of the SDF and contact normal are implemented using native CasADi functions and the map functionality for parallel evaluation. As noted in [25], the components of the `Callback` distance function are called by CasADi in the following order during an optimization procedure: nominal call, Jacobian call, adjoint and Hessian call. To this end, we save the solution to the perturbed KKT system determined during the nominal call and reuse it during the calls to derivatives for their evaluation. Furthermore, we save the Jacobian and reuse it during the adjoint and Hessian call.

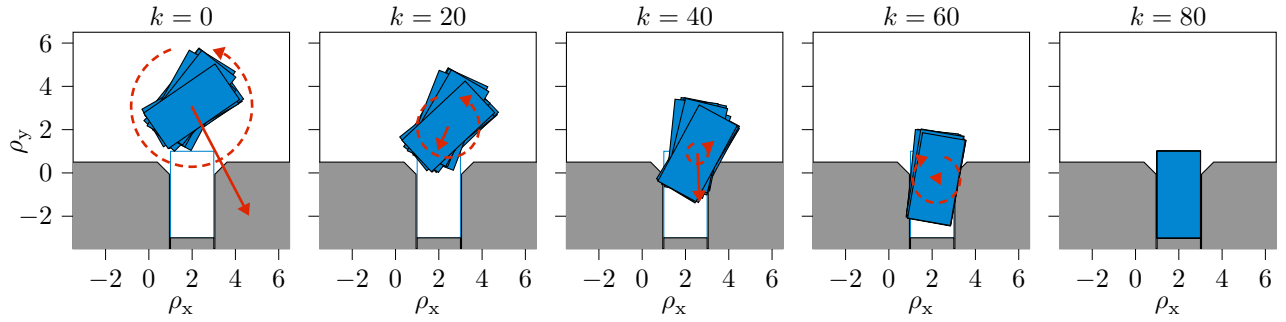


Fig. 1: Solution trajectory for the considered benchmark problem with  $N = 80$  and  $n_e = 10$ . The solid arrow represents linear control force  $(u_{k,x}, u_{k,y})$  and the dashed arrow represents angular control force  $u_{k,a}$ , such that  $u_k = (u_{k,x}, u_{k,y}, u_{k,a})$ . The arrow lengths are proportional to the control magnitudes.

TABLE I: Parameters for the numerical experiment.

|                                        |                                        |
|----------------------------------------|----------------------------------------|
| $T = 6$                                | $N, n_e$ , cf. Figure 2                |
| $h = \frac{T}{N}$                      | $n_d = 3$                              |
| $\bar{q}_0 = (2, 3.3, -\frac{\pi}{4})$ | $\bar{q}_N = (2, -1, 0)$               |
| $\bar{x}_0 = (\bar{q}_0, \mathbf{0})$  | $\bar{x}_N = (\bar{q}_N, \mathbf{0})$  |
| $L_r(x, u) = \frac{1}{n_e} \ u\ _2^2$  | $L_t(x) = 10000 \ x - \bar{x}_N\ _2^2$ |
| $g_D(x, u) = \ u\ _2^2 - 1.2^2$        | $W = [-0.3, 0.3]^3 \times \{0\}^3$     |

**Remark 2.** We note that the procedure for nominal evaluations of the SDF can be improved by using a custom adaptation of an interior-point method that stops at the pre-defined barrier parameter  $\tau$  and then solves the perturbed KKT conditions (2) up to a selected tolerance. The well-established IPOPT [26] solver provides such an option, however, utilizing it proved to be much slower than the considered combination of HPIPM and rootfinder.

## V. NUMERICAL EVALUATION

We consider a planar peg-in-hole problem, where a rectangular actuated polytope has to be steered between two octagons and one rectangle as illustrated in Figure 1. We solve a corresponding instance of the discrete-time OCP (14), for which the performance of external SDF evaluation as detailed in Section IV is compared to the performance of embedded SDF evaluation as explained in Section III-C. Computations are carried out on a Lenovo ThinkPad P15 Laptop with Intel i7-11850H CPU and 32 GB RAM. We use IPOPT [26] as numerical solver from the Python interface of CasADi.

The discrete-time OCP (14) is solved for varying horizons  $N$  and number of trajectory samples  $n_e$  as noted in Figure 2. Further parameters for the considered problem are given in Table I. The mass matrix  $M$  is constructed accordingly for a rectangle with dimensions  $1 \times 2$  and mass unity. The discrete-time OCP (14) is solved by relaxing complementarity constraints as explained in Section III-B with  $\sigma = 0.01$ , the smooth SDF (8) is considered with  $\tau = 0.01$ . For each choice of  $N$  and  $n_e$ , ten problem instances are considered, which differ through redrawing the random variables  $w^j$  from  $W$ . The values in Figure 2 are obtained by averaging the results of the ten problem instances. All solves are started from

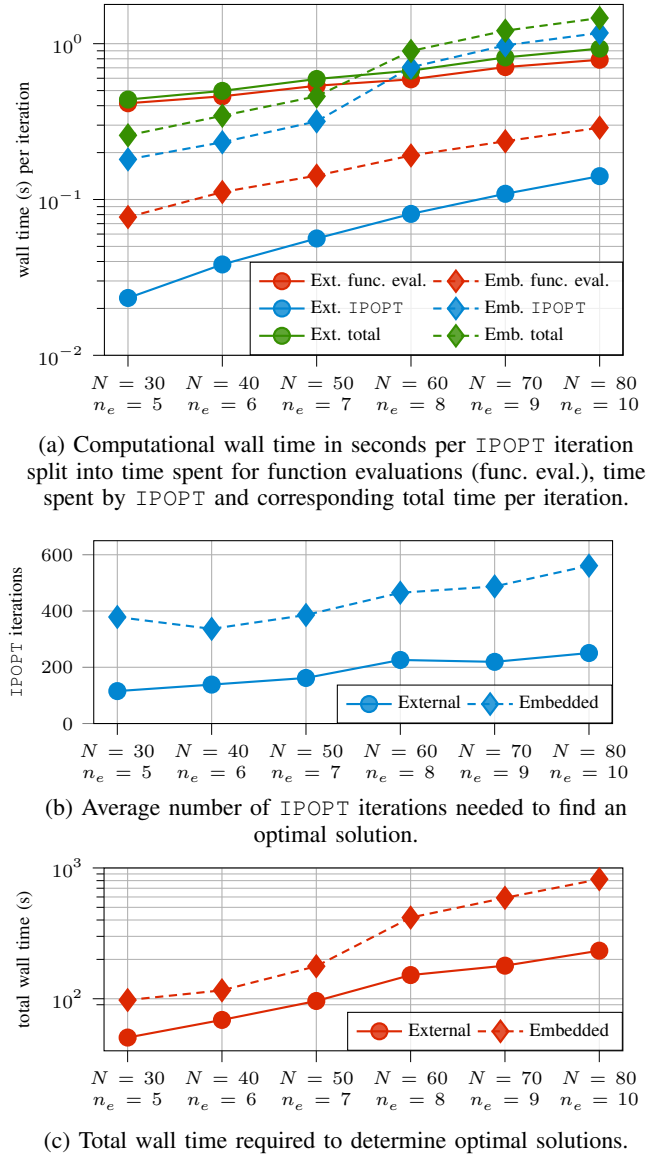


Fig. 2: Computational time and IPOPT iteration comparison for external and embedded SDF evaluation.

a trivial initial guess, which is given through all discrete-time configurations equal to the starting configuration, and all other discrete-time optimization variables equal to zero.

Figure 2 shows computation times and solver iterations for varying problem sizes. We observe that when utilizing external SDF evaluation, total computation time is dominated by time required for function evaluation. In contrast, for embedded SDF evaluation, total computation time is mainly determined by the solving operations carried out by IPOPT. Consequently, external SDF evaluation can be utilized to take computational load away from the numerical solver, this possibly facilitates tackling of complex manipulation problems in three-dimensions, as one obtains the means to distribute computational load between function evaluations and solving operations if the latter dominates the computation time. Now, in all considered problem variants, using external SDF evaluation results in faster convergence to an optimal solution compared to the embedded version as significantly fewer solver iterations are required, cf. Figure 2(b). This is reasonable as distance information for the external SDF evaluation is exact from the first solver iteration on, while distance information stemming from the embedded perturbed KKT conditions only gets precise in later iterations when the solver satisfies the equations (2) accurately. Our implementation for external SDF evaluation results in equal total computation time required per solver iteration for medium to large sized problems with more than  $\sim 400$  discretized state variables, cf. Figure 2(a). Since problems with external SDF evaluation always require significantly less solver iterations, we achieve a faster overall computation time for all considered problems, see Figure 2(c), resulting in an overall speed-up of up to factor four for the largest problem.

## VI. CONCLUSION

We proved that a smooth approximation for the signed distance between polytopes can be obtained by solving perturbed optimality conditions of an optimization-based distance formulation. We discussed how corresponding derivatives can be determined and introduced an approximation to the gradient which can be utilized as contact normal vector. The applicability of the method was illustrated in an numerical experiment, where utilization of external distance evaluation resulted in faster computation times compared to direct embedding of optimality conditions for all considered problem instances. In future work, we aim to explore how this methodology can be utilized to solve real-world robotic manipulation problems in three dimensions.

## VII. APPENDIX

### A. Proof of Lemma 1

Let  $q$  be fixed and consider  $\tilde{G} = G_l R(\xi_l)^\top$ ,  $\tilde{h} = h_l$ , for  $l = 1$  or  $l = 2$ . The set  $\tilde{\mathcal{P}} = \{p \mid \tilde{G}p \leq \tilde{h}\}$  corresponds to a version of  $\mathcal{P}_l$  that is rotated around the origin and, thus, due to Assumption 2 is also a bounded polytope. We show that the matrix

$$\begin{pmatrix} \tilde{G} & -\tilde{h} \end{pmatrix} \in \mathbb{R}^{\tilde{n} \times 3}, \quad (15)$$

with  $\tilde{n} = n_{l,h}$  has full rank. Since by Assumption 2 the origin is contained in the interior of the polytopes, we have  $\tilde{h}_i > 0$  for  $i = 1, \dots, \tilde{n}$ . By dividing the  $i$ -th row of (15) by  $\tilde{h}_i$  for  $i = 1, \dots, \tilde{n}$ , respectively, we obtain a matrix with equivalent rank given by

$$\begin{pmatrix} \hat{G} & -\mathbf{1} \end{pmatrix} \in \mathbb{R}^{\tilde{n} \times 3}, \quad (16)$$

where  $\hat{g}_i = \frac{\tilde{g}_i}{\tilde{h}_i} \in \mathbb{R}^2$  denote the rows of  $\hat{G} \in \mathbb{R}^{\tilde{n} \times 2}$ , and  $\tilde{g}_i \in \mathbb{R}^2$  denote the rows of  $\tilde{G}$ . Thus, we have

$$\hat{G} = \begin{pmatrix} \hat{g}_1^\top \\ \vdots \\ \hat{g}_{\tilde{n}}^\top \end{pmatrix}, \quad \tilde{G} = \begin{pmatrix} \tilde{g}_1^\top \\ \vdots \\ \tilde{g}_{\tilde{n}}^\top \end{pmatrix}.$$

We can use  $\hat{G}$  to represent  $\tilde{\mathcal{P}}$  as

$$\tilde{\mathcal{P}} = \{p \mid \tilde{G}p \leq \tilde{h}\} = \{p \mid \hat{G}p \leq \mathbf{1}\}.$$

Now, let

$$i, j, k \in \{1, \dots, \tilde{n}\}, i \neq j, j \neq k, i \neq k. \quad (17)$$

Due to Assumption 1, the halfspace representations of the polytopes are nonredundant and, thus,  $\hat{g}_{1,i}$  and  $\hat{g}_{1,j}$  are linear independent. Then there exist  $\mu_1, \mu_2 \in \mathbb{R}$ , possibly different for each choice of  $i, j, k$ , such that

$$\mu_1 \hat{g}_{1,i} + \mu_2 \hat{g}_{1,j} = \hat{g}_{1,k}.$$

In the following, we show that there exists a choice of  $i, j, k$  (17) such that

$$\Theta = \mu_1 + \mu_2 - 1 \neq 0,$$

from which we can infer that (15) has full rank.

To this end, assume for all choices of  $i, j, k$  (17) it holds  $\Theta = 0$ . We fix some  $i, j$  and consider differing  $k \in \{1, \dots, \tilde{n}\} \setminus \{i, j\}$ . If  $\Theta = 0$  holds for any  $k$ , all rows of  $\hat{G}_1$  are element of a line in  $\mathbb{R}^2$  and, thus, can be represented by

$$\hat{g}_{1,i} = v + \alpha_i w, \quad \alpha_i \in \mathbb{R}, \quad i = 1, \dots, \tilde{n},$$

with  $v, w \in \mathbb{R}^2$  orthogonal to each other. Now, if  $v \neq \mathbf{0}$ , define  $d = -v$ , and if  $v = \mathbf{0}$ , choose  $d \in \mathbb{R}^2$  to be any vector orthogonal to  $w$  with positive norm. Then for all  $\kappa > 0$  it holds

$$\kappa \hat{G} d \leq \mathbf{1},$$

which contradicts the assumption that  $\tilde{\mathcal{P}}$  is bounded.

Therefore, consider a choice of  $i, j, k$  (17) such that  $\Theta \neq 0$ . Then we can apply a elementary matrix operation to the submatrix of (16) consisting of the  $i$ -th,  $j$ -th and  $k$ -th row to obtain a matrix that clearly has full rank, i.e., we have

$$\text{rank} \begin{pmatrix} \hat{g}_{1,i}^\top & -1 \\ \hat{g}_{1,j}^\top & -1 \\ \hat{g}_{1,k}^\top & -1 \end{pmatrix} = \text{rank} \begin{pmatrix} \hat{g}_{1,i}^\top & -1 \\ \hat{g}_{1,j}^\top & -1 \\ \mathbf{0}^\top & \Theta \end{pmatrix}.$$

## ACKNOWLEDGMENTS

This research was supported by BMWK via 20D2123B, 03EI4057A and 03EN3054B, by DFG via Research Unit FOR 2401, project 424107692 and 525018088, and by the EU via ELO-X 953348. We thank Jonathan Frey and Anton Pozharskiy for fruitful discussions on implementation aspects of the signed distance function.

## REFERENCES

- [1] B. Brogliato, *Nonsmooth Mechanics*. Springer, 2016.
- [2] D. Stewart and M. Anitescu, “Optimal control of systems with discontinuous differential equations,” *Numerische Mathematik*, vol. 114, pp. 653–695, 2010.
- [3] Y. D. Zhong, J. Han, and G. O. Brikis, “Differentiable physics simulations with contacts: Do they have correct gradients w.r.t. position, velocity and control?,” in *ICML AI for Science Workshop*, 2022.
- [4] A. Nurkanović, M. Sperl, S. Albrecht, and M. Diehl, “Finite elements with switch detection for direct optimal control of nonsmooth systems,” *Numerische Mathematik*, vol. 156, pp. 1115–1162, 2024.
- [5] A. Nurkanović, J. Frey, A. Pozharskiy, and M. Diehl, “FESD-J: Finite elements with switch detection for numerical optimal control of rigid bodies with impacts and Coulomb friction,” *Nonlinear Analysis: Hybrid Systems*, vol. 52, p. 101460, 2024.
- [6] A. Aydinoglu, A. Wei, W.-C. Huang, and M. Posa, “Consensus complementarity control for multicontact MPC,” *IEEE Transactions on Robotics*, vol. 40, pp. 3879–3896, 2024.
- [7] V. Kurtz, A. Castro, A. Özgün Önel, and H. Lin, “Inverse dynamics trajectory optimization for contact-implicit model predictive control,” arXiv preprint arXiv:2309.01813, 2023.
- [8] C. J. Ong and E. Gilbert, “Growth distances: new measures for object separation and penetration,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 888–903, 1996.
- [9] X. Zhang, A. Liniger, and F. Borrelli, “Optimization-based collision avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 972–983, 2021.
- [10] J. Xie and N. Chakraborty, “Rigid body dynamic simulation with line and surface contact,” in *IEEE Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pp. 9–15, 2016.
- [11] S. Le Cleac’h, M. Schwager, Z. Manchester, V. Sindhvani, P. Florence, and S. Singh, “Single-level differentiable contact simulation,” *IEEE Robotics and Automation Letters*, pp. 4012–4019, 2023.
- [12] C. Dietz, A. Nurkanović, S. Albrecht, and M. Diehl, “High accuracy numerical optimal control for rigid bodies with patch contacts through equivalent contact points – Extended Version,” arXiv preprint arXiv:2403.13931, 2024.
- [13] A. Escande, S. Miossec, M. Benallegue, and A. Kheddar, “A strictly convex hull for computing proximity distances with continuous gradients,” *IEEE Transactions on Robotics*, vol. 30, pp. 666–678, 2014.
- [14] L. Montaut, Q. L. Lidec, A. Bambade, V. Petrik, J. Sivic, and J. Carpentier, “Differentiable collision detection: a randomized smoothing approach,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3240–3246, 2023.
- [15] K. Tracy, T. A. Howell, and Z. Manchester, “Differentiable collision detection for a set of convex primitives,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3663–3670, 2023.
- [16] K. Tracy and Z. Manchester, “On the differentiability of the primal-dual interior-point method,” arXiv preprint arXiv:2406.11749, 2024.
- [17] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.
- [18] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016.
- [19] W. Hogan, “Directional derivatives for extremal-value functions with applications to the completely convex case,” *Operations Research*, vol. 21, pp. 188–209, 1973.
- [20] A. Nurkanović, S. Albrecht, and M. Diehl, “Limits of MPCC formulations in direct optimal control with nonsmooth differential equations,” in *European Control Conference (ECC)*, pp. 2015–2020, 2020.
- [21] D. E. Stewart and J. C. Trinkle, “An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction,” *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 2673–2691, 1996.
- [22] M. Anitescu and F. A. Potra, “Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems,” *Nonlinear Dynamics*, vol. 14, pp. 231–247, 1997.
- [23] I. Mordatch, K. Lowrey, and E. Todorov, “Ensemble-CIO: Full-body dynamic motion planning that transfers to physical humanoids,” in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 5307–5314, 2015.
- [24] G. Frison and M. Diehl, “HPIPM: a high-performance quadratic programming framework for model predictive control,” *IFAC-PapersOnLine*, vol. 53, pp. 6563–6569, 2020.
- [25] J. Frey, J. D. Schutter, and M. Diehl, “Fast integrators with sensitivity propagation for use in CasADi,” in *European Control Conference (ECC)*, pp. 1–6, 2023.
- [26] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, pp. 25–57, 2006.