

# Learning-Guided Force-Feedback Model Predictive Control with Obstacle Avoidance for Robotic Deburring

Krzysztof Wojciechowski<sup>1</sup>, Ege Gursoy<sup>1</sup>, Arthur Haffemayer<sup>1</sup>, Sebastien Kleff<sup>2</sup>,  
Vincent Bonnet<sup>1,3</sup>, Florent Lamiroux<sup>1</sup> and Nicolas Mansard<sup>1,4</sup>

**Abstract**—Model Predictive Control (MPC) is widely used for torque-controlled robots, but classical formulations often neglect real-time force feedback and struggle with contact-rich industrial tasks under collision constraints. Deburring in particular requires precise tool insertion, stable force regulation, and collision-free circular motions in challenging configurations, which exceeds the capability of standard MPC pipelines. We propose a framework that integrates force-feedback MPC with diffusion-based motion priors to address these challenges. The diffusion model serves as a memory of motion strategies, providing robust initialization and adaptation across multiple task instances, while MPC ensures safe execution with explicit force tracking, torque feasibility, and collision avoidance. We validate our approach on a torque-controlled manipulator performing industrial deburring tasks. Experiments demonstrate reliable tool insertion, accurate normal force tracking, and circular deburring motions even in hard-to-reach configurations and under obstacle constraints. To our knowledge, this is the first integration of diffusion motion priors with force-feedback MPC for collision-aware, contact-rich industrial tasks. The source code can be found at [github.com/agimus-project/agimus-demos](https://github.com/agimus-project/agimus-demos).

## I. INTRODUCTION

Model Predictive Control (MPC) has long been the staple for motion generation on torque-controlled robots, with applications ranging from locomotion [1], [2], [3] to collision avoidance [4], [5]. However, most MPC formulations optimize motion alone without explicitly incorporating real-time force feedback [6], [7], [8], making force tracking purely feedforward and sensitive to modeling errors.

In contrast, the control of contact forces has been a longstanding topic in robotics [9], [10], originally developed through direct and indirect feedback methods such as hybrid force/motion control [11], impedance control [12], or admittance control [9]. These techniques offer reactivity and robustness but lack planning capabilities, which limits effectiveness in complex and contact-rich tasks.

To bridge this gap some works attempted to incorporate optimization into force control, for instance through MPC formulations based on linear spring models [13], [14], [15], [16], bounded admittance behavior [17], or adaptive

This work is supported by the European project AGIMUS (under GA no.101070165), ANITI (ANR-19-P3IA-0004), NERL (ANR-23-CE94-0004-02), AS2 (ANR-22-EXOD-0006) PEPR O2R project and by Défi Clé Robotique centrée sur l’Humain funded by Région Occitanie, France.

<sup>1</sup>LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

<sup>2</sup>Inria, AUCTUS team, Talence, France

<sup>3</sup>Image and Pervasive Access Laboratory (IPAL), CNRS-UMI, 2955, Singapore

<sup>4</sup>Artificial and Natural Intelligence Toulouse Institute, France

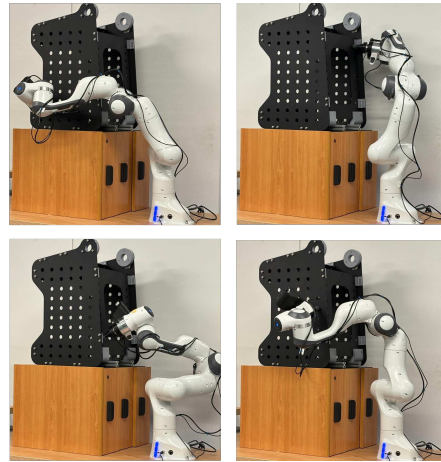


Fig. 1: Torque-controlled Franka Emika Panda robot executing the deburring task in scenario 3 (see sec. V-D). The robot inserts a custom end-effector into different holes of the workpiece and applies a constant normal force while performing circular motions, all while adapting to varying joint configurations and avoiding collisions with the environment.

impedance models [18], [19], [20]. These approaches improve motion–force coordination but often rely on simplified or low-dimensional dynamics.

Conversely, another line of work attempted to extend MPC with force feedback, either by relaxing the rigid contact model [21], [22], using spring models [23], or modifying the control structure [24], [25]. These hybrid approaches do demonstrate the importance of contact force prediction, but often face trade-offs: simplified dynamics [23], low planning rates [26], or increased state size and model approximations [27]. Force feedback is typically either ignored or estimated from joint torques. More recently, [28] proposed a unified formulation that combines force control and nonlinear MPC without these compromises.

Despite progress, running such controllers online in real time remains challenging, especially for tasks like deburring, where stable contact and precise tool motions must be maintained under uncertainty [29]. In these tasks, not only the force magnitude but also the spatiotemporal profile of force application is critical. The robot must retain and reproduce task-specific motion strategies, for instance characteristic insertion poses and trajectories for each hole to deburr. While MPC typically exploits the previous solution as a warm start, this only accelerates convergence locally and does not provide a true memory of task-specific motion strategies across different instances.

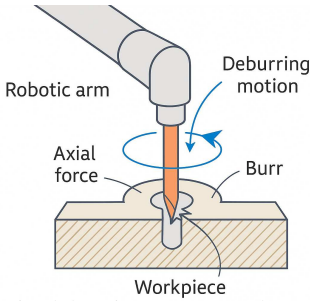


Fig. 2: Robotic deburring process: the manipulator applies a constant normal force along the tool axis while performing a circular motion inside the hole to remove burrs with limited tangential forces.

To address this limitation, generative models have recently emerged as powerful motion priors. Classical representations such as dynamic movement primitives or Gaussian mixtures could encode demonstrations [30], [31], [32], [33], but were limited in capturing the variability and multi-modality of real robot behavior. Diffusion models overcome these issues by learning rich distributions of feasible motions. Recent diffusion-based policies have shown that such models can generate diverse, feasible trajectories conditioned on goals or sensory input [34], [35], [36], making them a natural candidate for serving as a memory of motion in contact-rich planning.

In this paper, we propose a framework that combines force-feedback MPC with diffusion-based motion priors. Diffusion models provide robust initialization and adaptation for the predictive controller, while MPC ensures safe execution with force regulation and collision avoidance. We demonstrate the approach on a deburring task that requires precise tool insertion, circular force-controlled motions, and obstacle avoidance.

Our contributions are threefold. First, we formalize the deburring task and formulate the corresponding Optimal Control Problem (OCP). Second, we integrate diffusion-based motion priors into a force-feedback MPC. Finally, we experimentally validate the framework on a torque-controlled manipulator. Our experimental setup is shown in Fig. 1. To the best of our knowledge, this is the first integration of diffusion-based motion priors with force-feedback MPC for a contact-rich task such as deburring.

## II. PROBLEM STATEMENT

Deburring is the process of removing excess material to smooth the edges of a machined metal part, as shown in Fig. 2. To this end, we define it as the insertion of a precision tool into a hole, followed by the application of a constant normal force  $\lambda_{\perp}$  along the hole axis while performing a circular motion inside it to remove excess material. Tangential forces,  $\lambda_{\parallel}$ , should not be zero but only limited. In hole deburring, the tangential force component provides the cutting and abrading action. Excessive tangential force, however, can cause slip-out. The objective is to track the desired normal force and reference trajectory, while bounding tangential forces within the friction cone within a safe envelope, i.e. to respect the friction cone  $\|\lambda_{\perp}\|_2 \leq \alpha \lambda_{\parallel}$ . In our empirical observations

on dry steel deburring, a friction coefficient of  $\alpha = 0.6$  is adopted. This value is conservative, ensuring stable contact and preventing slip-out.

Reaching and deburring small holes under collision avoidance is particularly challenging in real manufacturing settings. Fig. 1 shows the realistic aeronautics-inspired workpiece used in our study. On the shop floor, it is common to reposition or re-fixturing either the workpiece or the robot base to improve manipulability for each hole or batch of holes, which is an approach that increases setup time, cost, and system complexity. A more scalable alternative is to plan from a single, fixed robot pose. However, this makes the motion-planning problem especially hard: such workpieces are highly nonconvex, with thin corridor-like features and tight clearances around fixtures, creating narrow passages where small path deviations can cause collision or loss of reachability.

Overall to perform a deburring task the following elements are required (i) tight geometric clearances around the workpiece and fixture, (ii) strict orientation constraints to align the tool axis with the hole axis, and (iii) force control that prevent slippage. These requirements must be satisfied under robot dynamics and rate limits at high control frequencies leaving little time for collision checking, Jacobian/Hessian updates, or replanning. As a result, the task stresses both planning and control: it couples geometric feasibility with contact stability and real-time computation, making it an exciting benchmark for MPC-based contact-aware motion planning.

## III. FORCE-FEEDBACK MODEL PREDICTIVE CONTROL

We recall here the force-feedback MPC approach described in [28].

### A. Optimal Control Problem

The OCP is defined as:

$$\begin{aligned} \min_{\underline{\mathbf{x}}, \underline{\mathbf{u}}} \quad & \sum_{t=0}^{T-1} \ell_t(\mathbf{x}_t, \mathbf{u}_t) + \ell_T(\mathbf{x}_T) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t), \quad 0 \leq t < T, \\ & c_t(\mathbf{x}_t, \mathbf{u}_t) \geq 0, \quad 0 \leq t < T, \\ & c_T(\mathbf{x}_T) \geq 0. \end{aligned} \quad (1)$$

where  $\underline{\mathbf{x}} = \{\mathbf{x}_0, \dots, \mathbf{x}_T\}$ ,  $\underline{\mathbf{u}} = \{\mathbf{u}_0, \dots, \mathbf{u}_{T-1}\}$  are the state and control input trajectories,  $T \in \mathbb{N}^*$  is the horizon length,  $\ell_t, f_t, c_t$  are the running cost, dynamics and constraint at stage  $t$  and  $\ell_T, c_T$  are the terminal cost and constraint.

### B. Classical MPC

For torque-controlled manipulators, a common state-control choice is the joint position–velocity pair  $(\mathbf{q}, \dot{\mathbf{q}})$  and the joint torque input  $\boldsymbol{\tau}$  [6], [7], [8]. In this case, for a fully actuated  $n$ -DoF manipulator in contact with the environment, the forward dynamics are expressed as:

$$\ddot{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = M^{-1}(\mathbf{q}) (\boldsymbol{\tau} - h(\mathbf{q}, \dot{\mathbf{q}}) + J(\mathbf{q})^{\top} \boldsymbol{\lambda}) \quad (2)$$

where  $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$  is the generalized inertia matrix,  $h(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$  denotes gravity and Coriolis terms,  $J(\mathbf{q}) \in$

$\mathbb{R}^{m \times n}$  is the contact Jacobian,  $\boldsymbol{\lambda} \in \mathbb{R}^m$  are the contact forces with  $1 \leq m \leq 3$ , and  $\ddot{\mathbf{q}}$  is the vector of joint accelerations.

This formulation, under the rigid contact assumption [37], defines the *classical* MPC setting. Contact forces can be introduced as *feedforward* terms in the dynamics, but they are not directly regulated through explicit force measurement feedback. Therefore, force objectives can be encoded in the cost function, yet tracking remains model-dependent and sensitive to uncertainties.

### C. Force-feedback MPC

In order to allow force feedback from sensors, we follow the approach described in [28] which augments the position-velocity state with the contact force  $\boldsymbol{\lambda}$ . The state and control are defined as:

$$\begin{aligned} \mathbf{x} &= (\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) \in \mathbb{R}^{2n+m} \\ \mathbf{u} &= \boldsymbol{\tau} \in \mathbb{R}^n \end{aligned} \quad (3)$$

Contact forces are modeled with a linear spring–damper law as:

$$\boldsymbol{\lambda}(\mathbf{q}, \dot{\mathbf{q}}) = -K \Delta p(\mathbf{q}, p_c) - B \dot{p}(\mathbf{q}, \dot{\mathbf{q}}). \quad (4)$$

where  $\Delta p(\mathbf{q}, p_c) = p(\mathbf{q}) - p_c$  is the end-effector deflection relative to the anchor  $p_c \in \mathbb{R}^m$ ,  $p(\mathbf{q}) \in \mathbb{R}^m$  is the forward kinematics, and  $\dot{p}(\mathbf{q}, \dot{\mathbf{q}}) = J(\mathbf{q})\dot{\mathbf{q}}$  is the Cartesian velocity. The matrices  $K, B \in \mathbb{R}^{m \times m}$  are diagonal and positive definite. Assuming no slip, differentiating (4) yields to the force dynamics:

$$\begin{aligned} \dot{\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{u}) &= -K \dot{p}(\mathbf{q}, \dot{\mathbf{q}}) - B \ddot{p}(\mathbf{x}, \mathbf{u}) \\ \ddot{p}(\mathbf{x}, \mathbf{u}) &= J(\mathbf{q})\ddot{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\tau}) + \dot{J}(\mathbf{q})\dot{\mathbf{q}} \end{aligned} \quad (5)$$

The overall system dynamics combine joint and force dynamics as:

$$f_t = \begin{bmatrix} \dot{\mathbf{q}} \\ M^{-1}(\boldsymbol{\tau} - h + J^\top \boldsymbol{\lambda}) \\ -KJ\dot{\mathbf{q}} - B(JM^{-1}(\boldsymbol{\tau} - h + J^\top \boldsymbol{\lambda}) + \dot{J}\dot{\mathbf{q}}) \end{bmatrix} \quad (6)$$

This MPC formulation allows to combine the benefits of nonlinear MPC and direct force control and has been shown to outperform both taken individually in challenging contact tasks such as surface polishing and static multicontact pushing [28]. However, tasks requiring the combined regulation of end-effector motion and tangential contact forces, such as deburring, have not been investigated yet. In particular, two contact models are considered for the deburring task: the 1D model which only predicts the normal contact force and assumes unconstrained tangential movement ( $m = 1$ ) and the 3D model which additionally predicts the tangential forces ( $m = 3$ ).

### D. Cost and constraints

The controller penalizes deviations in end-effector pose and contact force, while regularizing state and input effort. Hard constraints ensure feasibility under joint, torque, and collision limits. Throughout, the weighted norm is defined as  $\|v\|_Q^2 := v^\top Q v$  for  $Q \succeq 0$ .

*Stage and terminal costs:* The running and terminal costs are

$$\begin{aligned} \ell_t &= \|e_t\|_{Q_e}^2 + \|\mathbf{x}_t - \mathbf{x}_t^{\text{ref}}\|_{Q_x}^2 + \|\mathbf{u}_t - \boldsymbol{\tau}(\mathbf{q}_t, \boldsymbol{\lambda}_t)\|_{R_u}^2 \\ \ell_T &= \|e_T\|_{Q_{e,T}}^2 + \|\mathbf{x}_T - \mathbf{x}_T^{\text{ref}}\|_{Q_{x,T}}^2 \end{aligned} \quad (7)$$

where  $Q_e = \text{diag}(Q_p, Q_R, Q_{\boldsymbol{\lambda},\parallel}, Q_{\boldsymbol{\lambda},\perp})$  (and similarly for  $Q_{e,T}$ ) is block diagonal to weight the residuals independently. The residual vector is defined as:

$$\begin{aligned} e_t &= [e_p(t), e_R(t), e_{\boldsymbol{\lambda},\parallel}(t), e_{\boldsymbol{\lambda},\perp}(t)]^\top \\ e_p(t) &= p(\mathbf{q}_t) - p_t^{\text{ref}} \in \mathbb{R}^3 \\ e_R(t) &= \text{vec}(\log(R_t^{\text{ref}\top} R(\mathbf{q}_t))) \in \mathbb{R}^3 \\ e_{\boldsymbol{\lambda},\parallel}(t) &= \boldsymbol{\lambda}_{t,\parallel} - \boldsymbol{\lambda}_{t,\parallel}^{\text{ref}} \in \mathbb{R} \\ e_{\boldsymbol{\lambda},\perp}(t) &= \boldsymbol{\lambda}_{t,\perp} \in \mathbb{R}^2 \quad (3D \text{ model only}) \end{aligned} \quad (8)$$

where  $p(\mathbf{q}) \in \mathbb{R}^3$  and  $R(\mathbf{q}) \in \mathbb{SO}(3)$  are the forward kinematics of the end-effector position and orientation;  $p_t^{\text{ref}}$  and  $R_t^{\text{ref}}$  are their time-varying references;  $\text{vec}(\log(\cdot)) : \mathbb{SO}(3) \rightarrow \mathbb{R}^3$  denotes the orientation error as a rotation vector;  $\boldsymbol{\lambda}_t \in \mathbb{R}^m$  is the contact force;  $n \in \mathbb{R}^3$  is the unit vector along the hole axis and  $\boldsymbol{\lambda}_{t,\parallel}^{\text{ref}} \in \mathbb{R}$  is the desired normal contact force. The term  $e_{\boldsymbol{\lambda},\parallel}$  enforces tracking of the desired normal force, while  $e_{\boldsymbol{\lambda},\perp}$  penalizes tangential forces, which are defined as:

$$\begin{aligned} \boldsymbol{\lambda}_{t,\parallel} &= n^\top \boldsymbol{\lambda}_t \in \mathbb{R}, \\ \boldsymbol{\lambda}_{t,\perp} &= (I - nn^\top) \boldsymbol{\lambda}_t \in \mathbb{R}^2 \quad (3D \text{ model only}). \end{aligned} \quad (9)$$

Note that the tangential force  $\boldsymbol{\lambda}_{t,\perp}$  and residual  $e_{\boldsymbol{\lambda},\perp}(t)$  are only used with the 3D model ( $m = 3$ ). The reference  $\boldsymbol{\tau}(\mathbf{q}_t, \boldsymbol{\lambda}_t)$  is the gravity/Coriolis compensation torque, typically  $\mathbf{g}(\mathbf{q}_t) - J(\mathbf{q}_t)^\top \boldsymbol{\lambda}_t$ , assuming zero reference velocity. It was tracked thanks to a regularization term  $R_u$ .

*Task-phase weighting:* The cost terms are scheduled across phases of the task:

- 1) *Trajectory following:* follow joint trajectory ( $Q_x$  active), no force tracking ( $Q_{\boldsymbol{\lambda},\parallel} = Q_{\boldsymbol{\lambda},\perp} = 0$ ), no pose tracking ( $Q_p = Q_R = 0$ ).
- 2) *Pre-contact:* increase pose tracking ( $Q_p, Q_R$ ), reduce reliance on reference configuration.
- 3) *Insertion:* increase  $Q_{\boldsymbol{\lambda},\parallel}$  to build normal force, keep  $Q_{\boldsymbol{\lambda},\perp}$  large to suppress tangential force, maintain orientation with  $Q_R$ , and disable reference configuration tracking ( $Q_q = 0$ ).
- 4) *Deburring:* track circular  $p_t^{\text{ref}}$  in the plane orthogonal to  $n$ , enforce normal force with  $Q_{\boldsymbol{\lambda},\parallel}$ , suppress tangential forces with  $Q_{\boldsymbol{\lambda},\perp}$ , and maintain tool orientation with  $Q_R$ .

*Collision avoidance:* For each pair  $(i, j)$  of robot–environment bodies, a signed-distance margin  $\varepsilon > 0$  is imposed:

$$c_{t,i,j}(\mathbf{q}_t) := d_{ij}(\mathbf{q}_t) - \varepsilon \geq 0, \quad (10)$$

where  $d_{ij}(\mathbf{q}_t)$  is the signed distance between body  $i$  and body  $j$ . For the trajectory optimization algorithm, based on

Sequential-Quadratic Programming (SQP) [38], the following linearization is used [39]:

$$c_{t,i,j}(\mathbf{q}_t) \approx c_{t,i,j}(\bar{\mathbf{q}}_t) + \nabla_{\mathbf{q}} d_{ij}(\bar{\mathbf{q}}_t)^\top (\mathbf{q}_t - \bar{\mathbf{q}}_t) \geq 0 \quad (11)$$

#### IV. DIFFUSION PRIORS AS MEMORY OF MOTION

Collision avoidance in cluttered, non-strictly convex workspaces is hard for local optimizers, since the feasible set is highly non-convex and narrow passages are common. MPC with generic initialization often converges to poor local minima, or spends many iterations just to recover feasibility. This motivates a warm-start that already encodes task-specific strategies and collision-aware geometry.

##### A. Trajectory generation

To do so, we build a synthetic dataset of feasible deburring motions that captures the multimodality of motions around the different holes and poses. The motion data are obtained by the task-and-motion planner Humanoid Path Planner [40], by defining a simple task sequence composed of 3 steps: reach a pre-grasp location in front of the hole, apply the contact, retract to a rest location.

For each configuration we (i) solve inverse kinematics to reach the target end effector pose from many seeds, (ii) connect start and goal with a bi-directional RRT\* in configuration space to get a collision free path, (iii) refine this path with an OCP to obtain feasible and smooth trajectories. Since the dataset is inherently multimodal, with multiple approach directions and collision-avoidance strategies, we train a conditional diffusion motion to capture this diversity without mode collapse and enable controlled sampling, providing a reusable prior that encodes motion strategies.

##### B. Diffusion model for memory of motion

The model maps a noise sequence to a trajectory proposal conditioned on the initial robot state  $\mathbf{x}_0$  and the target end-effector pose. We implement this with a Diffusion Transformer (DiT) [41], following the standard diffusion architecture for trajectories [42], [34], where trajectory tokens represent joint configurations, while conditioning tokens encode  $\mathbf{x}_0$  and the target pose using learnable type and timestep embeddings. An encoder builds a memory from the conditioning tokens, a decoder cross-attends to this memory while processing trajectory tokens, and a linear head predicts the noise across all timesteps. To satisfy real-time constraints, we restrict the diffusion process to a small number of denoising steps  $D_s = 25$ , rather than the hundreds typically used in generative modeling, which keeps inference fast while preserving robustness and diversity, especially when combined with classifier-free conditioning.

##### C. Using the diffusion as prior

The diffusion output is not executed directly. Instead, the diffusion output serves as a reference for the OCP inside the MPC loop, similar to the approach in [43]. We use the state regularization term from (7), but set its reference to the diffusion prior trajectory. This effectively pulls the solution toward the prior while dynamics, force objectives,



Fig. 3: Custom end-effector mount for the Franka Emika Panda integrating a force-torque sensor and provisions for RGB-Depth camera. A spherical tool tip enables smooth circular motion within holes. (Scenario 2, Sec. V-D)

and hard constraints enforce feasibility. In practice, this gives the solver a favorable basin of attraction and accelerates convergence [31], while MPC ensures force tracking, torque limits, and collision avoidance.

## V. EXPERIMENTS

### A. Setup

A custom instrumented end-effector is attached to a Franka Emika Panda robot, as shown in Fig. 3. The end effector is provided with an ATI Mini45 force-torque sensor and an Intel RealSense D435i RGB-D camera. The tool has a spherical shape to allow smooth circular deburring-like motions. The Diffusion model is running on a computer with AMD Ryzen 9 5950X CPU with NVIDIA RTX 3060 GPU. The MPC is implemented in Python in Crocodyl [44] with CSQP solver [38] running inside of a real-time process of `ros2_control` on a computer with Intel i9-14900K @5.6 GHz. Both computers are connected in the same network and communicate through ROS2 Humble.

Similarly to [28], the controlled torques are linearly interpolated at 1 kHz, and the measured forces are low-pass filtered using a second-order Butterworth filter (cut-off frequency: 10 Hz).

### B. Task definition

The robot is set to reach one or more holes with progressively increasing difficulty (Scenarios 1–3 described below). Upon reaching a hole, the robot regulates a normal deburring force of 30 N and executed a circular (revolute) passes at angles of 7.5 deg and 15 deg about the hole axis, each lasting 5 s. When activated, the collision constraint is imposed between a capsule approximation of the forearm and a planar obstacle in front of the robot.

### C. Scenario 1: Accessible Single-Hole Deburring

This experiment serves as the method baseline for the force feedback MPC. It also evaluates 1D and 3D contact models under hard collision avoidance constraints. As explained earlier, the 1D model regulates only the normal force along the tool axis, whereas the 3D model also accounts for tangential forces. The setup, visible in Fig. 3, consists of a very accessible single hole positioned on a table in front of the robot ( $x = 0.52$  m,  $y = 0.0$  m, in robot base frame shown in Fig. 6) at the same height as the robot's base.

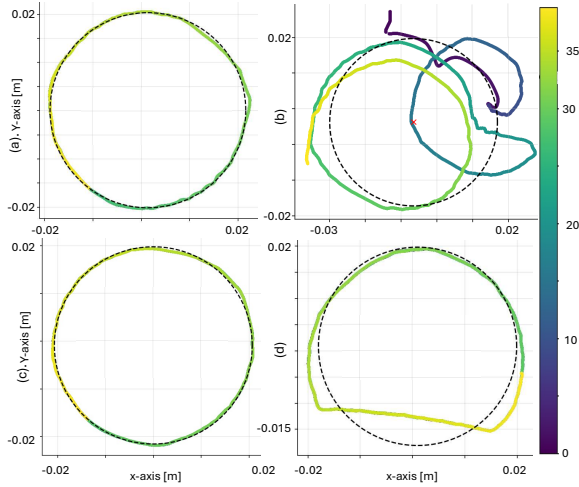


Fig. 4: Scenario 1 (Sec. V-C): Force-colored end-effector x-y trajectories for a single hole; color encodes normal force magnitude. (a) 1D contact, no obstacle avoidance; (b) 1D contact with an obstacle in front of the robot; (c) 3D contact, no collision avoidance; (d) 3D contact with collision avoidance. The obstacle shape is clearly visible in the successful example c, and prevents the robot to achieve the complete circle.

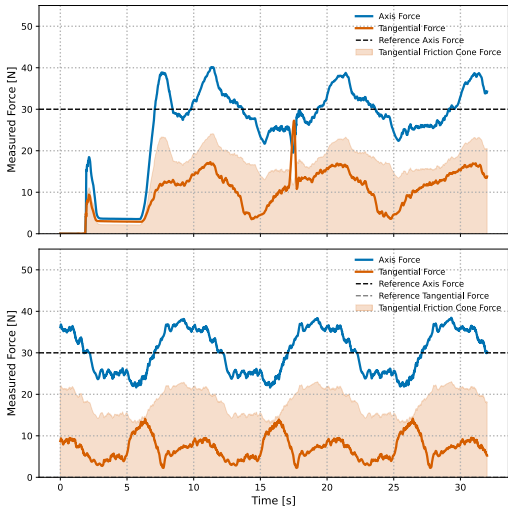


Fig. 5: Scenario 1 (Sec. V-C): Tracking of  $\lambda_{\perp}$  and  $\lambda_{\parallel}$  during deburring motion on a flat table with collision avoidance with 1D (top) and 3D contact (bottom) models. In this case when 1D contact model approaches collision,  $\lambda_{\parallel}$  exceed force friction cone ( $\mu = 0.6$ ) and break the contact while 3D contact maintains stable movement.

Identical cost weights and gains were used across conditions to ensure fair comparison. For these experiments, the MPC was executed at 200 Hz with a 300 ms horizon and 15 nodes, using 3 SQP iterations and 150 QP iterations per cycle. In the nominal setting, the robot performs the deburring task without any obstacle. As shown in Fig. 4a and 4c, when there is no collision constraint, both models were able to correctly track desired end-effector trajectories. The normal force tracking for both models in this obstacle-free setting along four repetitions of the deburring task is shown in Fig. 5a and Fig. 5b. The Root Mean Square Error (RMSE) was

TABLE I: Inverse velocity manipulability and corresponding force RMSE.

Hole	$w_f$ (-)	Force RMSE [N]
Hole 1	10.6956	11.30
Hole 2	6.4821	7.98
Hole 3	6.9166	17.20
Hole 4	19.0211	17.41
Hole 5	12.3884	15.34

5.35 N and 4.40 N, for 1D and 3D models respectively. In the nominal case both models perform equally well.

A second setup placed the robot close to a wall (i.e. close to a collision). In this setting, the nominal circular deburring motion would make the robot collide with the wall. For the 3D model, activating the collision-avoidance constraint in (10) forced the solver trade-off trajectory tracking against constraint satisfaction and deviate the end-effector path during execution (Fig. 4d). Under these conditions, the 1D contact model failed to maintain stable in-hole contact (Fig. 4b). The tool slid out of the hole as the optimizer prioritized clearance over orientation, while tangential forces were not explicitly regulated. This led to violation of the no-slip condition. This setup confirms that the explicit regulation of tangential forces is crucial to preserve contact stability.

#### D. Scenario 2: Obstacle-Constrained Multi-Hole Reach-and-Deburring Sequence

This scenario evaluates force-tracking performance in more demanding configurations involving both collision-avoidance constraints and far-reach targets (Fig. 6a). The workpiece contains holes intentionally placed near the robot's workspace boundary: the robot's maximum nominal reach is 0.85 m, whereas Holes 3 and 2 are located at radial distances of 0.92 m and 1.00 m, respectively. Additional holes are positioned to highlight the effect of a plane collision constraint defined at the base of the workpiece (Fig. 7a). The resulting five reaching postures are shown in Fig. 6; as expected, holes 3–2 induce near fully-extended arm configurations. Thus, as shown in Table I this induces a reduced force manipulability.<sup>1</sup> To improve stability in these configurations, the controller is executed at 500 Hz with a prediction horizon of 30 ms, discretized into 15 nodes. The solver uses 3 SQP iterations and 25 QP iterations per cycle. No collision constraints are applied in this task.

Across all five targets, deburring is completed without slip or collisions (success rate of 100%), indicating that the controller can jointly satisfy contact and geometric constraints even in adverse poses. However, the normal-force tracking RMSE increases as the robot operates farther from the baseline pose of Scenario 1 (Fig. 6c). This degradation is consistent with (i) poor Jacobian conditioning near workspace limits, which reduces force authority along the hole axis; (ii) proximity to joint/torque bounds that constrains corrective actions; and (iii) tighter orientation feasibility when the end-effector must remain aligned with the hole

<sup>1</sup>Force manipulability [45] is calculated as  $w_f = \frac{1}{\sqrt{(\det(JJ^T))}}$ .

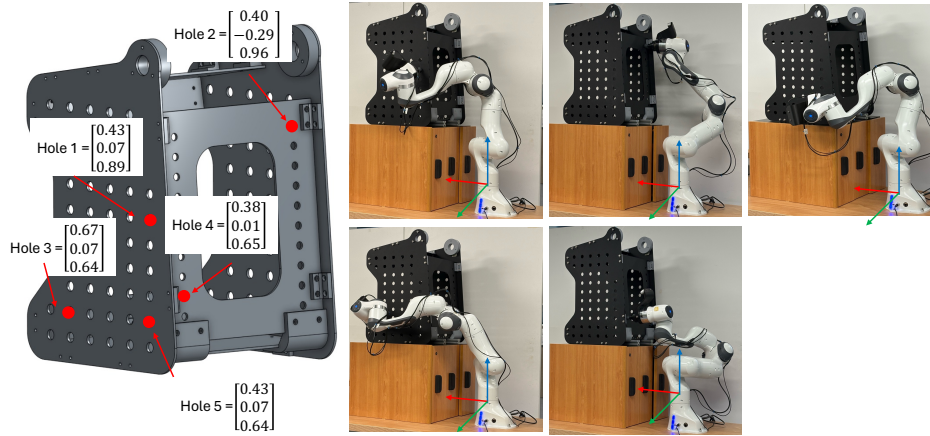


Fig. 6: Positions of the 5 holes to be deburred on the investigated workpiece and reaching postures for each hole.

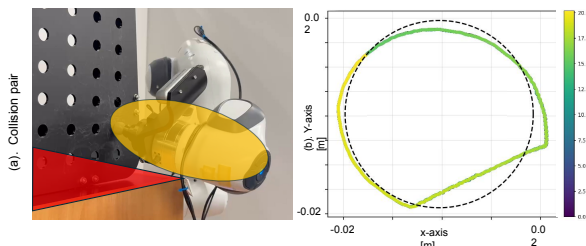


Fig. 7: Description of the collision pair between the plane (red) and ellipsoid (yellow) (a). Force-colored end-effector  $x$ - $y$  trajectories for a single hole. Color encodes  $\lambda_{\perp}$ .

while respecting the plane constraint. In practice, these factors reduce the effective bandwidth available for force regulation and increase sensitivity to small pose deviations or modeling errors. For holes adjacent to the plane constraint, the nominal circular deburring trajectory can not be executed in full, as the avoidance constraint truncated the path, which is an outcome analogous to Scenario 1 but amplified by the far-reach geometry. As shown in Fig. 7. In these cases the solver produces collision-aware path deflections that preserves contact stability while trading off perfect trajectory tracking against the maintenance of the desired normal force.

### E. Scenario 3: Long Sequence Integration

In this scenario visible in the accompanying video, it is shown a complete integration of the deburring task on the MPC level with a diffusion-guided motion around the object. Here we demonstrate the complete integration of the pipeline. An ad-hoc sequence decides the list of holes to be machined. For each hole, the diffusion model is queried to produce a reference guide used as specified in Sec. IV.

The MPC executes the motion while regulating the contact force. It compensates for model inaccuracies and object pose deviations, while simultaneously enforcing local collision avoidance. The diffusion guidance allows the robot to reach difficult locations beyond narrow passages which would be very expensive to compute at run-time with an on-line planner. Holes are reached and machined within the accuracy range observed in Scenario V-D. As shown in the video, the torque controlled compliance also tolerates physical disturbance by a human co-operator.

## VI. DISCUSSIONS

The proposed method enabled us to perform a complex industrial task by integrating capabilities that are often treated separately in robotics workflows: collision avoidance, reactive control, accurate force feedback. While we have argued that the approach is broadly applicable, its current performance relies on several assumptions and design choices that warrant discussion.

First, our approach relies on offline computations using a motion planner to train a diffusion model. In this work, we considered static scenes, so the computation cost and sample requirements are not representative of what would be needed to generalize to unknown environments, ideally by conditioning on sensor-based representations of obstacles and targets. Future work will explore parallelized motion planning on GPU [46], which could be integrated into our current workflow with minimal modifications. Moreover, exhaustive offline exploration could be alleviated by enabling fast refinement of the motion prior at run time, for instance with sample-based optimizers such as MPPI [47], which naturally provide strong exploration capabilities.

At high control frequencies, MPC cannot efficiently handle a large number of collision pairs. To reach 200  $Hz$  in our experiments, we indeed restricted the controller to a single pair. This limitation is largely mitigated by the motion prior, which provides safe, collision-free guidance, but it may become restrictive in more complex scenarios. Achieving such performance also requires careful tuning of the numerical solver (e.g., QP iterations), a know-how that could be alleviated by adopting recent advances in numerical optimization [48].

The experiments also highlight that a key limiting factor lies in the robot kinematics and link lengths. In such complex scenes, the robot struggles to reach all the holes shown in Fig. 1 while simultaneously maintaining the prescribed circular motion. Our results show that task accuracy is strongly correlated with manipulability. More generally, far-reach and obstacle-constrained conditions stress both motion planning and contact regulation, motivating future work on manipulability-aware cost shaping [49], [50] and constraint

handling to preserve force fidelity near workspace limits. One possible direction is to introduce a cost term such as  $\log(\det(J^T))$  to bias reaching postures toward configurations with higher directional force authority. However, the nonlinearity of such a term may negatively impact solver performance, which suggests investigating local convexifications or scheduling strategies to maintain a tractable QP at each MPC step.

It would also be beneficial to transfer the task to a mobile manipulator. The added mobility introduces the need for a careful sequencing of the holes to be machined, to minimize unnecessary base movements. External perception, e.g. through vision-based system, would become essential to coordinate the robot's repositioning and ensure accuracy.

## VII. CONCLUSION

We presented a solution based on model predictive control to perform an industrial-inspired deburring task. A controller enables the robot to safely navigate around complex obstacles, avoid collisions despite disturbances, and maintain precise force regulation through direct sensor feedback. Operating in torque control further improves the safety as the robot can be interrupted by a nearby worker, and the risk of damage is reduced even in the presence of undetected obstacles, which makes the approach suitable for human-robot collaboration.

## REFERENCES

- [1] R. Grandia and F. F. et al., "Feedback MPC for Torque-Controlled Legged Robots," 2019.
- [2] G. Romualdi and S. D. et al., "Online Non-linear Centroidal MPC for Humanoid Robot Locomotion with Step Adjustment," in *IEEE ICRA*, 2022.
- [3] E. Dantec and M. N. et al., "Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot," in *IEEE Humanoids*, 2022.
- [4] M. Gaertner and M. B. et al., "Collision-Free MPC for Legged Robots in Static and Dynamic Scenes," in *IEEE ICRA*, 2021.
- [5] A. Haffemayer and A. J. et al., "Collision Avoidance in Model Predictive Control using Velocity Damper," in *IEEE ICRA*, 2025.
- [6] F. Farshidian and E. J. et al., "Real-time motion planning of legged robots: A model predictive control approach," in *IEEE-RAS International Conference on Humanoid Robotics (Humanoids)*, 2017.
- [7] R. Grandia and F. J. et al., "Perceptive Locomotion Through Nonlinear Model-Predictive Control," *IEEE T-RO*, 2023.
- [8] C. Mastalli and S. P. C. et al., "Inverse-Dynamics MPC via Nullspace Resolution," *IEEE T-RO*, 2023.
- [9] D. E. Whitney, "Force Feedback Control of Manipulator Fine Motions," *Journal of Dynamic Systems Measurement and Control-Transactions of The Asme*, 1977.
- [10] L. Villani and J. D. Schutter, *Force Control*, 2008.
- [11] M. T. Mason, "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Transactions on SMC*, 1981.
- [12] N. Hogan, "Impedance Control Part1-3," *J. Dyn. Sys., Meas., Control.*, 1985.
- [13] M. D. Killpack and A. K. et al., "Model predictive control for fast reaching in clutter," *Autonomous Robots*, 2016.
- [14] J. Matschek and J. B. et al., "Force Feedback and Path Following using Predictive Control: Concept and Application to a Lightweight Robot," *IFAC-PapersOnLine*, 2017.
- [15] S. Husmann and S. S. et al., "Model Predictive Force Control in Grinding based on a Lightweight Robot," *IFAC-PapersOnLine*, 2019.
- [16] D. Müller and A. M. et al., "Model predictive force control for robots in compliant environments with guaranteed maximum force," in *ACC*, 2019.
- [17] A. Wahrburg and K. Listmann, "MPC-based admittance control for robotic manipulators," *CDC*, 2016.
- [18] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE RA-L*, 2020.
- [19] K. J. Kazim and J. B. et al., "Combined Predictive Path Following and Admittance Control," *Proceedings of the American Control Conference*, 2018.
- [20] M. V. Minniti and R. G. et al., "Model Predictive Robot-Environment Interaction Control for Mobile Manipulation Tasks," *Proceedings - IEEE ICRA*, 2021.
- [21] M. Neunert and M. S. et al., "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE RA-L*, 2018.
- [22] S. Fahmi and M. F. et al., "STANCE: Locomotion Adaptation over Soft Terrain," *IEEE TRO*, 2020.
- [23] T. Gold and A. V. et al., "Model Predictive Interaction Control for Robotic Manipulation Tasks," *IEEE T-RO*, 2023.
- [24] S. Kleff and E. D. et al., "Introducing force feedback in model predictive control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [25] A. Gazar and G. N. et al., "Jerk control of floating base systems with contact-stable parameterized force feedback," *IEEE T-RO*, 2021.
- [26] L. Wijayarathne and Z. Z. et al., "Real-Time Deformable-Contact-Aware Model Predictive Control for Force-Modulated Manipulation," *IEEE T-RO*, 2023.
- [27] M. Selvaggio and A. G. et al., "Non-prehensile object transportation via model predictive non-sliding manipulation control," *IEEE Transactions on Control Systems Technology*, 2023.
- [28] S. Kleff and A. J. et al., "Force Feedback in Model Predictive Control: A Soft Contact Approach," 2024.
- [29] M. Makulavičius and S. P. et al., "Industrial Robots in Mechanical Machining: Perspectives and Limitations," *Robotics*, 2023.
- [30] N. Jetchev and M. Toussaint, "Trajectory prediction: learning to map situations to robot trajectories," in *Proceedings - ICML*, 2009.
- [31] N. Mansard and A. D. et al., "Using a Memory of Motion to Efficiently Warm-Start a Nonlinear Predictive Controller," in *IEEE ICRA*, 2018.
- [32] T. S. Lembono and C. M. et al., "Learning How to Walk: Warm-starting Optimal Control Solver with Memory of Motion," in *IEEE ICRA*, 2020.
- [33] E. Dantec and R. B. et al., "Whole Body Model Predictive Control with a Memory of Motion: Experiments on a Torque-Controlled Talos," in *IEEE ICRA*, 2021.
- [34] M. Janner and Y. D. et al., "Planning with Diffusion for Flexible Behavior Synthesis," in *ICML*, 2022.
- [35] G. Zhou and S. S. et al., "Diffusion Model Predictive Control," 2024.
- [36] C. Chi and Z. X. et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *IJRR*, 2024.
- [37] R. Budhiraja and J. C. et al., "Differential dynamic programming for multi-phase rigid contact dynamics," in *IEEE Humanoids*, 2018.
- [38] A. Jordana and S. K. et al., "Structure-Exploiting Sequential Quadratic Programming for Model-Predictive Control," *IEEE T-RO*, 2025.
- [39] A. Haffemayer and A. J. et al., "Model predictive control under hard collision avoidance constraints for a robotic arm," in *UR*, 2024.
- [40] J. Mirabel and S. T. et al., "Hpp: A new software for constrained motion planning," in *IEEE IROS*, 2016.
- [41] W. Peebles and S. Xie, "Scalable Diffusion Models with Transformers," 2023.
- [42] J. Ho and A. J. et al., "Denosing diffusion probabilistic models," in *Proceedings - NeurIPS*, 2020.
- [43] A. Haffemayer and A. C. et al., "Collision-free model predictive control with diffusion model warm-starting," 2025.
- [44] C. Mastalli and R. B. et al., "Crocodyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," *CoRR*, 2019.
- [45] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, 1985.
- [46] J. Białkowski and S. K. et al., "Massively parallelizing the rrt and the rrt," in *IEEE IROS*, 2011.
- [47] C. H. Huang and P. J. et al., "prrtc: Gpu-parallel rrt-connect for fast, consistent, and low-cost motion planning," 2025.
- [48] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," 2020.
- [49] F. Marić and O. L. et al., "Fast manipulability maximization using continuous-time trajectory optimization," in *IEEE IROS*, 2019.
- [50] F. Kennel-Maushart and R. P. et al., "Manipulability optimization for multi-arm teleoperation," in *IEEE ICRA*, 2021.