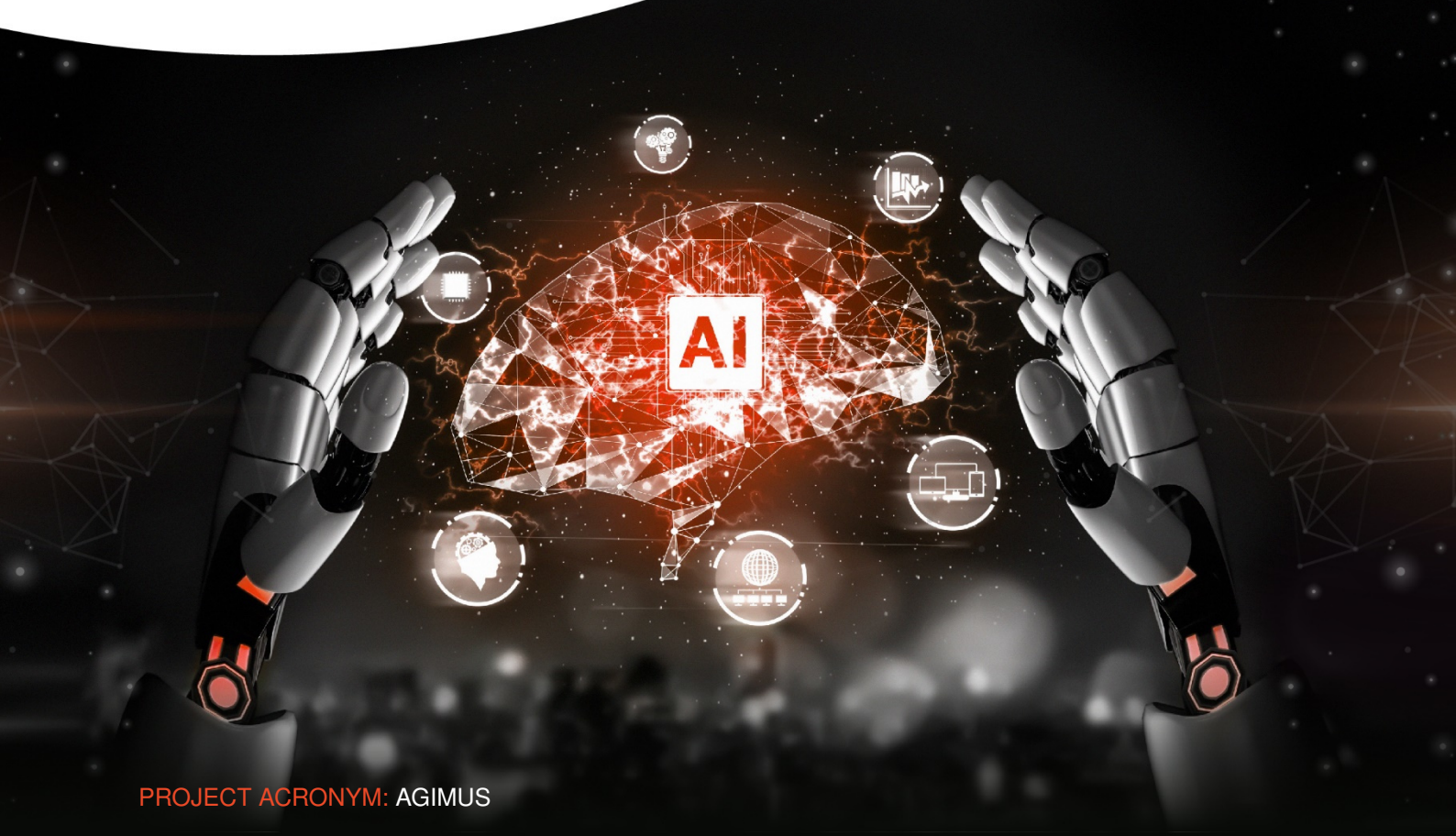
AGIMUS

INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

D4.4 – Sensor-driven MPC with visual-haptic feedback

First version



PROJECT ACRONYM: AGIMUS

PROGRAMME: Horizon Europe
GRANT AGREEMENT: No 101070165
TYPE OF ACTION: Horizon Research & Innovation Actions
START DATE: 1 October 2022
DURATION: 48 months

Funded by the European Union

Document information

Issued by:	CNRS
Issue date:	1/04/2025
Due date:	31/12/2024
Work package leader:	CTU
Start date:	1 April 2023
Dissemination level:	PUB (Public)

Document History

Version	Date	Modifications made by
1	1/4/2025	Nicolas Mansard

Authors

First name	Last name	Beneficiary
Nicolas	Mansard	CNRS

In case you want any additional information, or you want to consult with the authors of this document, please send your inquiries to: nmansard@laas.fr.

Quality reviewers

First name	Last name	Beneficiary
Vladimir	Petrik	CTU
Maximilien	Naveau	TOWARD

Disclaimer

Funded by the European Union under GA no. 101070165. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union. The granting authority cannot be held responsible for them.

Executive summary

This report was developed in the context of the AGIMUS project, funded by the Horizon Europe Framework Program for Research and Innovation of the European Union for 2021-2027. Its objective is to present the progress mainly towards the specific objective **SpO1.3: Enable high accuracy perception feedback combining visual and haptic input.** that is part of the **Objective 1: Significantly accelerate the introduction and deployment of a robotic system to a new agile production environment through advanced perception and understanding of human-centric feedback**.

Table of Contents

1 Introduction and motivation

Since the beginning of the project, we have developed real-time trajectory solvers capable of handling hard constraints (D2.3), modeling contact dynamics (D2.2), and incorporating visual feedback (D4.2), all deployed in the robot. These components have proven to be effective for a wide range of manipulation tasks. However, several key tasks targeted within the project (such as robotic machining with AIRBUS, glue dispensing with KLEEMANN and THIMM, and insertion operations and contact-guided alignment with KLEEMANN and AIRBUS) require not only accurate motion planning but also active control of the contact interface. This calls for a control strategy that can go beyond trajectory following and engage directly with the forces at play during contact.

We identified in the early stage of Agimus that Model-predictive control (MPC) stands out as a promising framework in this context, due to its ability to incorporate dynamic models, enforce constraints, and integrate sensory feedback within a unified optimization-based formulation. Nevertheless, conventional implementations of MPC remain fundamentally limited in contact-rich scenarios. These limitations stem largely from the absence of force measurements in the feedback loop: optimal policies are typically computed based only on position and velocity information, leaving force regulation and contact stability to lower-level control layers. This is also true in most reinforcement learning methods, and the propositions formulated in this document would also be largely applied in this context.

The challenge addressed in this deliverable is to enable MPC for direct haptic feedback, combining its versatility in managing constraints and multitask objectives with the precision and responsiveness required for force-driven execution. This integration is nontrivial, as advanced motion frameworks like MPC or RL often offload fine-grained contact handling to low-level controllers, sacrificing the very accuracy and adaptability that contact-rich tasks demand. Bridging this gap is critical for empowering robots to perform skilled, contact-intensive operations autonomously and robustly.

To address this, the work reported here introduces a novel model-predictive control formulation that systematically incorporates measured efforts into the control loop. The key idea is to augment the system state space with a viscoelastic model of contact forces expressed in the task space, thereby enabling the controller to reason directly over physical interaction with the environment. The proposed method is evaluated both in simulation and on hardware, and is benchmarked against (i) standard MPC formulations that neglect force feedback, and (ii) a previous method that introduced torque feedback in joint space. Results demonstrate that incorporating Cartesian force measurements significantly improves the robot ability to execute demanding contact tasks.

These contributions correspond to the following two articles or conference papers:

- Sébastien Kleff et al. “Force Feedback in Model Predictive Control: A Soft Contact Approach”. In: *HAL 04572399* (2024). Subm. IEEE TRO. URL: https://hal.science/hal-04572399/
- Armand Jordana et al. “Force feedback model-predictive control via online estimation”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024. URL:

https://hal.science/hal-04564888/

These contributions are summarized in the following sections, and the associated preprints are reported at the end of this document.

2 Background

2.1 Formulation of the optimal control problem

In AGIMUS, we typically consider the following Optimal Control Problem (OCP) transcripted as a nonlinear program (NLP):

$$\min_{\substack{x_0,\dots,x_N \\ u_0,\dots,u_{N-1}}} \quad \sum_{k=0}^{N-1} \ell_k(x_k, u_k) + \ell_N(x_N) \tag{1a}$$

$$\text{s.t.} \quad x_0 = \hat{x}, \tag{1b}$$

$$x_{k+1} = f_k(x_k, u_k) \tag{1c}$$

where $\hat{x}$ is the measured state, u the control input and f is the dynamics model, ℓ_k, ℓ_N and f_k denote the discretized costs and dynamics with sampling step Δt (typically resulting from a semi-implicit Euler integration scheme). The solution is a locally optimal control sequence $u_0^*, ..., u_{N-1}^*$ from which we only send the first element u_0^* to the robot low-level controllers. This NLP is typically solved using the solvers defined in D2.3. The step dynamics f_k typically arises from the time integration of a the differential equation:

$$\dot{x} = f(x, u)$$

.

2.2 MPC with contact

Existing MPC implementations like those developed in Agimus optimize contact forces as input variables [8, 10], control them only in a feed-forward sense [6, 13, 17, 18]. Some works have relaxed the rigid contact assumption yet mostly to enable some gradients at the contact interface [20, 7] hence creating some contact invariance (i.e. capability to decide the contact phases while deciding the movement).

A *rigid contact* is a kinematic constraint between the robot and the environment. The equations of motion of a fully-actuated robot in rigid contact with the environment can be derived from the Karush-Kuhn-Tucker (KKT) conditions of the convex optimization problem corresponding to Gauss' least constraint principle [22], as already explained in D2.1:

$$\min_{\ddot{q}} \ \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)} \tag{2a}$$

$$\text{s.t. } J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0 \tag{2b}$$

where $q, \dot{q} \in \mathbb{R}^n$ are the vectors of joint positions and velocities, $M(q) \in \mathbb{S}^n_+$ is the generalized inertia matrix, $J(q) \in \mathbb{R}^{m \times n}$ is the Jacobian of the m-dimensional contact, $\ddot{q}_f = M(q)^{-1}(\tau - h(q, \dot{q})) \in \mathbb{R}^n$ is the free acceleration, $h(q, \dot{q}) \in \mathbb{R}^n$ is the vector of centrifugal, Coriolis and gravity forces and $\tau \in \mathbb{R}^n$ is the vector of joint torques. The KKT conditions of (2), namely

$$\begin{bmatrix} M(q) & J(q)^T \\ J(q) & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ -\lambda^r \end{bmatrix} = \begin{bmatrix} \tau - h(q, \dot{q}) \\ -\dot{J}(q)\dot{q} \end{bmatrix} \tag{3}$$

reveal the generalized contact forces $\lambda^r \in \mathbb{R}^m$ as the Lagrange multipliers associated with the rigid contact constraint (2b).

In this formulation, the continuous-time dynamics constraint f is defined from the solution map of the KKT conditions (3). It corresponds a controlled dynamical system with state ${}^{cl}x \triangleq (q, \dot{q})$ and control input ${}^{cl}u \triangleq \tau$ where f maps joint positions, velocities and torques to the constrained joint accelerations $\ddot{q}$

$$f\left({}^{cl}x, {}^{cl}u\right) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)\left(\tau - h(q, \dot{q}) + J(q)^T \lambda\right) \end{bmatrix} \tag{4}$$

and λ^r is the rigid contact force (dual solution of (3)).

With this formulation, it clearly appears that **the contact force λ is an output with direct input feedthrough**, which means that the input torque affects *directly* the value of the contact force without any dynamics in between, making $lambda$ an invariant estimated from sensors non-strictly proper [5]. **In other words, the force can be written as a function of the state *and* the control input** and no force feedback can be established.

2.3 Works related to MPC with direct force feedback

In AGIMUS, we are interested in making the decision about MPC directly based on force measurements. For now, the MPC is making a decision as a joint torque command, which the low-level joint servo controllers of the robot are then tracking using joint torque measurement. Yet, the MPC itself never uses the joint torque measurements and cannot adapt its decisions based on these feedbacks. Even more, if a force sensor is attached to the end effector of the robot, the methods previously used in the project are not able to take any advantages of the measurements it provides. The goal of this deliverable is to establish new methods to consider the feedback of a force sensors directly inside the MPC, to improve the capability of the robot to accurately achieve forces driven by a physical interaction, such as machining, inserting, etc.

While it is possible in some cases to achieve output feedback control by exploiting the feedthrough as a feedforward action [9], it remains a challenge in the general case [12, 19]. In particular, the feedback interconnection of such a system with a non-linearity (e.g. a model-predictive controller) is known to create an algebraic loop of which the well-posedness is not guaranteed [1, 5]. Digital implementations of MPC usually circumvent this pathology by discarding the direct feedthrough terms as physical systems have a finite bandwidth [3].

One possible way is to augment the state by introducing delay [16], for instance by treating input as states [21, 23] or by using output predictions in the input computation [4]. The former approach was used in [2] to derive contact-aware optimal policies, and in our previous approach [15] to enable torque-feedback in MPC. In this paper, we address this challenge by proposing a new MPC formulation that introduces contact dynamics rather than adding dynamics in the joint space.

We have explored two complementary solutions. The first contribution is to introduce a soft contact model in our MPC to obtain direct feedback from force measurements. This is achieved by introducing a temporal decoupling between the MPC command and the force measurement, to the cost of a larger state space. The second contribution is to adapt the contact model through force measurements. Both approaches are complementary and can be adapted together.

3 Contribution #1 – Introducing a soft contact model in MPC to obtain direct feedback from force measurements

The main idea is to use a nonrigid contact model, where the force measurement is not a direct output of the torque command. This decoupling introduces a temporal difference between the measurement and the command, hence leading to direct force feedback. It follows the spirit of a previous paper [15] by extending it from joint torque feedback to end-effector force feedback.

3.1 Overview of the mathematics of the model

We used a visco-elastic contact dynamics to model the contact force by a linear spring-damper

$$\lambda^s(q, \dot{q}, p_c) \triangleq -K\Delta p(q, p_c) - B\dot{p}(q, \dot{q}) \tag{5}$$

where $\Delta p(q, p_c) = p(q) - p_c$ is the end-effector deflection, $p(q) \in \mathbb{R}^m$ is the end-effector Cartesian pose given by forward kinematics, $p_c \in \mathbb{R}^m$ is the contact anchor point, and $\dot{p}(q, \dot{q}) = J(q)\dot{q}$ is the end-effector Cartesian velocity given by differential forward kinematics. The matrices $K, B \in \mathbb{R}^{m\times m}$ are the stiffness and damping of the environment (diagonal and positive definite), assumed to be fixed and known. In practice, they can be arbitrary high to accurately reflects the high stiffness of the typical object of interest (e.g. contact between a machine tool and a stiff object to machine).

We define the force feedback MPC by the augmented state ${}^{ff}x \triangleq (q, \dot{q}, \lambda)$ which includes the classical state ${}^{cl}x$ and the measured contact force λ, while the control input is the joint torque ${}^{ff}u \triangleq \tau$ like in the classical formulation. Assuming $\dot{p}_c = 0$ (see paper for details) the time derivative of (5) is given by

$$\dot{\lambda}^s\left({}^{ff}x, {}^{ff}u\right) = -K\dot{p} - B\ddot{p} \tag{6}$$

where $\ddot{p} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$ is the end-effector Cartesian acceleration, and the joint acceleration $\ddot{q}$ is computed using the forward rigid-body dynamics. This reformulation leads to the full continuous-time dynamics model

$$^{ff}\dot{x} = f\left(^{ff}x, ^{ff}u\right) \tag{7}$$

where f includes the classical forward rigid-body dynamics $\ddot{q}$ *and* the visco-elastic force dynamics defined in (6)

$$f\left(^{ff}x, ^{ff}u\right) = \begin{bmatrix} \dot{q} \\ M^{-1}\left(\tau - h + J^T\lambda\right) \\ -KJ\dot{q} - B\left(JM^{-1}\left(\tau - h + J^T\lambda\right) + \dot{J}\dot{q}\right) \end{bmatrix} \tag{8}$$

where we dropped the dependencies in $q, \dot{q}$ for readibility. This model expresses the direct measure of λ (which will be the case in our experiments thanks to a F/T sensor attached to the end-effector) and uses the visco-elastic interaction model to predict its previewed evolution. Now the MPC is able to take decisions based on the predicted whole-body behaviour while being directly informed of the measured forces. In this formulation, a contact force task is formulated as an integral part of the state cost

$$\ell\left(^{ff}x, ^{ff}u\right) = \|\lambda - \bar{\lambda}\|^2 + \text{other terms} \tag{9}$$

3.2 Experimental evaluation

3.2.1 Benchmarks on a simplified polishing task

We first validate the capability of the system to track a reference force during a task where the robot slides along a contact surface while controlling the force in the normal direction to the contact. An overview of the setup is given in Fig. 1. Among other evaluations, we triggered the capabilities of our contribution in the case where the contact model is inaccurate, by tilting the plate. The accuracy of the controller in both positioning and force is plotted in Fig. 2, and compared to classical (rigid) MPC of D2.3 and our previous MPC with joint-torque feedback. These results are summarized by Table 1.

The proposed controller enables an accurate tracking of the contact force while also improving the positioning of the effector (which is expected when the contact interface is improperly modeled). Other evaluations are detailed in the companion paper [14], and featured in the video available on the paper project page[1]), where the repository of the code is also provided.

[1] https://gepettoweb.laas.fr/articles/kleff__tro_2025.html

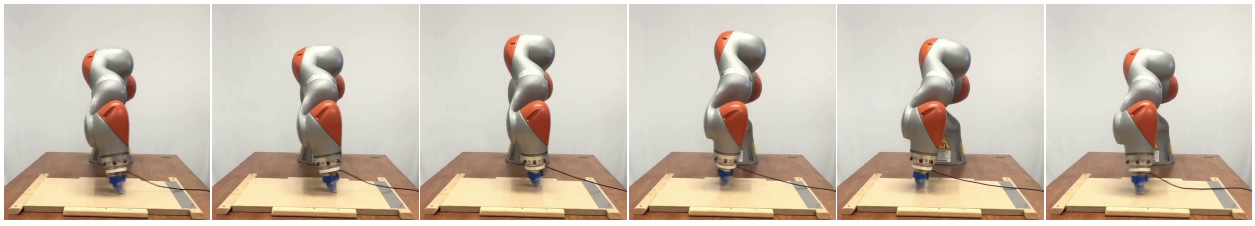
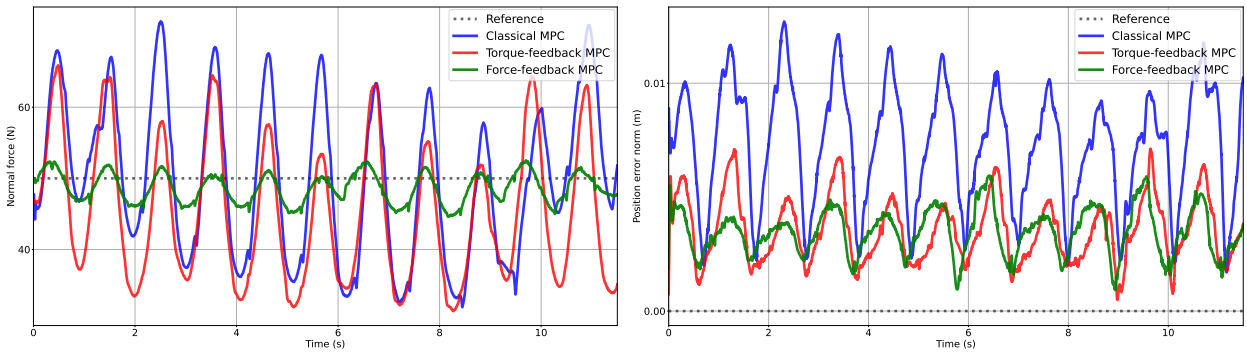


Figure 1: Snapshots of the benchmark with unkwnown table tilt: the robot has to perform circular movements on a flat surface while accuretely controlling both the 2D effector position on the surface and the normal interaction force.



(a) Normal force tracking error. **(b) Position tracking error.**

Figure 2: Tracking performance of the proposed controller with force feedback, compared to the previous controller, when the contact interface is not perfectly known (unkwnown table tilt).

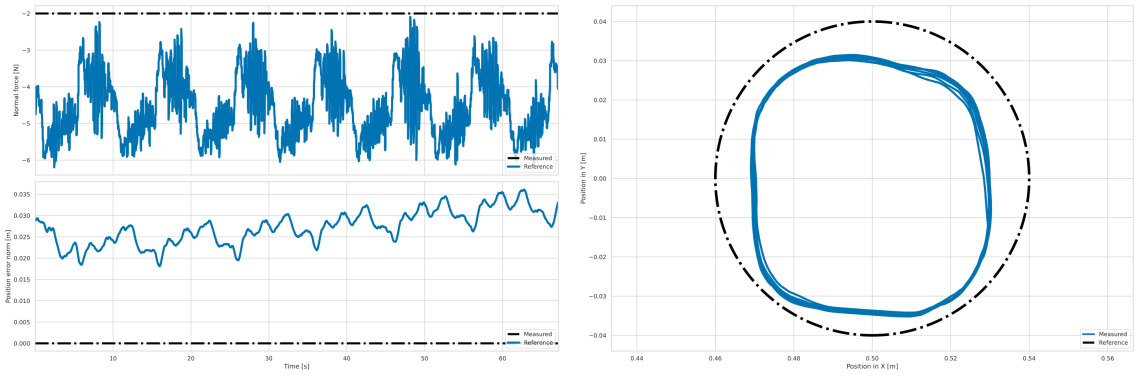
	Classical	Torque-feedback	Force-feedback
Force (N)	9.49 ± 1.59	9.67 ± 0.61	$\mathbf{2.12 \pm 0.37}$
Position (mm)	7.04 ± 0.43	3.37 ± 0.28	$\mathbf{3.34 \pm 0.11}$

Table 1: Summary of the benchmark with unkwnown table tilt: MAE of the normal force and end-effector position for the slow polishing task with imperfect table model.

Figure 3: Overview of the task with vision and haptic feedback: the robot draws a circle on a board while carefully controlling the normal force to 1N, and adapt its motion to the motion of the board.

3.2.2 Feedback combining vision and haptic

We then implemented a preliminary version of a MPC controller combining vision feedback and haptic feedback. In a first time, we only considered a QR-code tag as visual landmark, and programmed the robot to draw a circle with controlled normal force with respect to the location of the visual landmark. The robot then adapts to the movement of the plate and draws the circle in a constant position on the board. An overview of the task is given in Fig. 3. While we were expecting to mostly transfer this task on the AGIMUS-TIAGo robot, we have not spent significant time to tune the accuracy of the controller. The resulting end-effector position and pen force are plotted in Fig. 4.



(a) Normal force tracking **(b) End effector position**

Figure 4: Tracking performance of the proposed controller with force feedback, compared to the previous controller, when the contact interface is not perfectly known (board tilted).

4 Contribution #2 – Adapting the contact model through force measurements

In the first contribution, we have taken advantage of the capability to predict the future force evolutions to adequately control them. While this is effective for some context of interest, for example controlling the normal forces in an insertion task, for other contexts like controlling the friction forces during the movement along the surface, we lack of effective and computational models able to properly predict interaction changes. In this second contribution, we show that standard estimation tools (maximum likelihood optimization) together with a reformulation of the optimal control problem can provide a simple yet effective framework to achieve force-output-feedback MPC.

4.1 Formulation of the estimation problem

We consider a modification of the contact dynamics (2):

$$M(q)(\ddot{q} - \ddot{q}_f) = J^T(\lambda + \Delta\lambda) \tag{10a}$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0 \tag{10b}$$

where $\Delta\lambda$ is a perturbation that offsets the dynamics, here at the level of the contact interface (the paper proposes a more generic framework we reduced here to $\Delta\lambda$ for readibility). This offset is meant to correct the model mismatch due to inaccurate modeling of the dynamics, contact model, external disturbance, etc. The idea is to estimate the offset online, given raw measurement, given a prior on the offset, we use joint positions, velocities, accelerations, torque commands, and force measurements to update the force offset. We use maximum likelihood estimation, i.e. Kalman filtering if no constraint is imposed, and a small optimization problem otherwise, to compute the most likely $\Delta\lambda$.

The offset can be considered directly in the model used by the MPC. More precisely, we can consider that the offset will be constant over the horizon of the MPC and solve the OCP using as dynamics (3). The MPC model is then updated online at each offset estimate update. Interestingly, we also shown that the resulting MPC is in fact equivalent to modifying the force reference in the cost function, making the implementation straightforward.

4.2 Results

Compared to the previous contribution, where only the normal forces were considered, we here consider the friction and sliding effect and seek to improve the behavior of the robot (e.g. motion tracking) by using a proper force estimate. We considered three tasks in our benchmarks:

- The same polishing tasks as in the first contribution (motion in contact along a plane while controlling the 2D position and normal force);

- Tracking a force step while in a punctual contact, the reference force suddenly varies;
- Minimizing the energy in a task of tracking a varying configuration, while the force feedback is mostly used as a regularization to avoid slippage.

In particular, we have shown that the friction estimation leads to improved accuracy compared to classical predictive models of the friction. The complete analysis of the performances is available in the companion paper [11] and featured in the video available on the paper project page[2]), where the repository of the code is also provided.

5 Conclusion

This deliverable reports the findings from the research activities carried out in Task T4.3 of the AGIMUS project. We proposed two complementary solutions to enable direct force feedback within a model predictive control (MPC) framework. When a reliable model of the contact interaction is available (particularly for normal forces encountered during tooling or insertion tasks), we introduced a predictive model based on soft contact. This model allows the MPC to compute an optimal policy directly informed by force measurements. In contrast, when no predictive model of the contact interface is available (particularly for estimating future friction forces tangential to motion), we proposed to estimate the current contact forces and injecting them as a constant dynamic offset into the MPC. Both approaches are complementary and compatible.

In both scenarios, the proposed methods outperform existing state-of-the-art approaches, as demonstrated through simulation and hardware experiments.

Furthermore, the performance achieved is compatible with the requirements specified for the case studies evaluated in Deliverable D6.1. In coordination with Task T5.4, we have begun deploying these methods in the project testing zones. In parallel, we are finalizing integration on the AGIMUS TIAGo prototype, in close collaboration with Task T5.2, to mitigate deployment risks on the new hardware.

The next steps involve completing the deployment on the updated prototypes and enabling these control strategies for use in the case studies defined in Work Package 6 (WP6).

[2] https://gepettoweb.laas.fr/articles/jordana__icra2024.html

	Classical	Torque feedback	Force feedback	Force estimation
Reference	D2.3	[15]	(8),(9)	(10)
State	$(q, \dot{q})$	$(q, \dot{q}, \tau)$	$(q, \dot{q}, \lambda)$	$(q, \dot{q})$
Control	τ	$\dot{\tau}$	τ	τ
Actuation model	Perfect	Low-pass filter	Perfect	Perfect
Force	$\lambda = \lambda^r$ (output of (3))	$\lambda = \lambda^r$ (output of (3))	$\lambda = \lambda^s$ (state of (6))	$\lambda + \Delta\lambda$ (force of (10))
Contact model	Rigid	Rigid	Soft	Rigid

Table 2: Mathematical recap of the possible MPC formulation, classical (first column, without effort feedback), with joint torque feedback (second column, our previous work), with end-effector force feedback (third column, our first contribution). Both methods presented in this document are compatible, e.g. considering force-feedback in the normal direction and force estimation in tangent directions.

Reference List

[1] Ambrose A. Adegbege and William P. Heath. “Multivariable algebraic loops with complementarity constraints enforcing some KKT conditions”. In: *2014 52nd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2014* 2 (2014), pp. 1033–1039. DOI: 10.1109/ALLERTON.2014.7028568.

[2] Alp Aydinoglu et al. “Stabilization of Complementarity Systems via Contact-Aware Controllers”. In: *IEEE TRO* 38 (2021). DOI: 10.1109/TRO.2021.3120931.

[3] Alberto Bemporad, N. Lawrence Ricker, and James Gareth Owen. “Model predictive control - New tools for design and evaluation”. In: *Proceedings of the American Control Conference*. Vol. 6. 2004, pp. 5622–5627. ISBN: 0780383354. DOI: 10.1109/ACC.2004.249036.

[4] Chin Leei Cham et al. “Model predictive control with direct feedthrough with application on a MIST reactor”. In: *IFAC-PapersOnLine* 53.1 (2020), pp. 183–188. ISSN: 24058963. DOI: 10.1016/j.ifacol.2020.06.031. URL: https://doi.org/10.1016/j.ifacol.2020.06.031.

[5] Mohammed Dahleh and George Verghese. “Lectures on Dynamic Systems and Control Interconnected Systems and Feedback : Well-Posedness , Stability , and Performance”. In: *Electrical Engineering* ().

[6] Ewen Dantec, Michel Taïx, and Nicolas Mansard. “First Order Approximation of Model Predictive Control Solutions for High Frequency Feedback”. In: *IEEE RAL* 7 (2022). DOI: 10.1109/LRA.2022.3149573.

[7] Shamel Fahmi et al. “STANCE: Locomotion Adaptation over Soft Terrain”. In: *IEEE TRO* 36.2 (Apr. 2020). ISSN: 19410468. DOI: 10.1109/TRO.2019.2954670. arXiv: 1904.12306.

[8] Farbod Farshidian et al. “Real-time motion planning of legged robots: A model predictive control approach”. In: *IEEE-RAS International Conference on Humanoid Robotics (Humanoids)*. 2017. DOI: 10.1109/HUMANOIDS.2017.8246930.

[9] L. R. Fletcher. “Output feedback matrices in the presence of direct feedthrough”. In: *International Journal of Systems Science* 12.12 (1981), pp. 1493–1495. ISSN: 14645319. DOI: 10.1080/00207728108963834.

[10] Ruben Grandia et al. “Perceptive Locomotion Through Nonlinear Model-Predictive Control”. In: *IEEE Transactions on Robotics* 39.5 (2023), pp. 3402–3421. ISSN: 19410468. DOI: 10.1109/TRO.2023.3275384. arXiv: 2208.08373.

[11] Armand Jordana et al. “Force feedback model-predictive control via online estimation”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024. URL: https://hal.science/hal-04564888/.

[12] Seungeun Kang, Jihyoung Cha, and Sangho Ko. “Linear quadratic regulation and tracking using output feedback with direct feedthrough”. In: *International Journal of Aeronautical and Space Sciences* 17.4 (2016), pp. 593–603. ISSN: 20932480. DOI: 10.5139/IJASS.2016.17.4.593.

[13] Sotaro Katayama and Toshiyuki Ohtsuka. “Whole-body model predictive control with rigid contacts via online switching time optimization”. In: *IEEE International Conference on Intelligent Robots and Systems* 2022-October (2022), pp. 8858–8865. ISSN: 21530866. DOI: 10.1109/IROS47612.2022.9981790. arXiv: 2203.00997.

[14] Sébastien Kleff et al. “Force Feedback in Model Predictive Control: A Soft Contact Approach”. In: *HAL 04572399* (2024). Subm. IEEE TRO. URL: https://hal.science/hal-04572399/.

[15] Sébastien Kleff et al. “Introducing Force Feedback in Model Predictive Control”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 13379–13385. DOI: 10.1109/IROS47612.2022.9982003.

[16] J. M. Maciejowski, P. J. Goulart, and E. C. Kerrigan. *Constrained control using model predictive control*. Vol. 346. 2007, pp. 273–291. ISBN: 3540370099. DOI: 10.1007/978-3-540-37010-9_9.

[17] Carlos Mastalli et al. “Inverse-Dynamics MPC via Nullspace Resolution”. In: *IEEE Transactions on Robotics* 39.4 (2023), pp. 3222–3241. ISSN: 19410468. DOI: 10.1109/TRO.2023.3262186. arXiv: 2209.05375.

[18] Avadesh Meduri et al. “BiConMP: A Nonlinear Model Predictive Control Framework for Whole Body Motion Planning”. In: *IEEE Transactions on Robotics* 39.2 (Apr. 2023), pp. 905–922. ISSN: 1941-0468. DOI: 10.1109/TRO.2022.3228390. URL: https://arxiv.org/abs/2201.07601.

[19] Marcel Menner, Anuradha M. Annaswamy, and Alexander W. Zollitsch. “Adaptive output feedback for plants with direct feedthrough”. In: *2016 IEEE 55th Conference on Decision and Control, CDC 2016* Cdc (2016), pp. 2121–2127. DOI: 10.1109/CDC.2016.7798577.

[20] Michael Neunert et al. “Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1458–1465. ISSN: 23773766. DOI: 10.1109/LRA.2018.2800124. arXiv: 1712.02889.

[21] Andrzej W. Ordys and Andrew W. Pike. “State space generalized predictive control incorporating direct through terms”. In: *Proceedings of the IEEE Conference on Decision and Control* 4.December (1998), pp. 4740–4741. ISSN: 01912216. DOI: 10.1109/cdc.1998.762083.

[22] Firdaus E. Udwadia and Robert E. Kalaba. “A new perspective on constrained motion”. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439 (1992), pp. 407–410. ISSN: 0962-8444. DOI: 10.1098/RSPA.1992.0158.

[23] Marcelo A. Xavier, Aloisio Kawakita de Souza, and M. Scott Trimboli. “A split-future MPC algorithm for lithium-ion battery cell-level fast-charge control”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 12459–12464. ISSN: 24058963. DOI: 10.1016/j.ifacol.2020.12.1330. URL: https://doi.org/10.1016/j.ifacol.2020.12.1330.

Force Feedback in Model Predictive Control: A Soft Contact Approach

Sébastien Kleff[1,2], Armand Jordana[1], Nicolas Mansard[2,3], Ludovic Righetti[1]

***Abstract*—Model-predictive control is an appealing framework to control robots due to its ability to exploit both sensory information and model predictions. But its performance remains fundamentally limited in tasks involving contact with the environment, in part because optimal control policies do not reason over force measurements. In this article, we propose a first complete answer to this issue by introducing a novel approach that systematically includes measured efforts into the optimal control loop. We propose to augment the state-space with a visco-elastic model of the contact force in the task space. We derive a complete predictive controller with an efficient formulation whose implementation is released in open-source. We conduct extensive comparisons with two other methods: the classical model-predictive control formulation, which inherently restricts the feedback to position and velocity information, and our previous approach that enabled torque feedback in the joint space. We demonstrate through simulation studies and hardware experiments, the benefit of exploiting Cartesian force measurements in the model-predictive control framework to achieve challenging contact tasks.**

I. INTRODUCTION

A. Model-Predictive Control in contact tasks

Over the past decade, nonlinear Model-Predictive Control (MPC) has become increasingly practicable to address robot motion generation problems, mainly thanks to efficient rigid-body dynamics algorithms [1] and continuous progress in numerical optimal control, as attested by numerous hardware implementations on torque-controlled robots [2]–[7]. But its systematic deployment for tasks involving delicate contact interactions (where quantified forces are expected to achieve the task) remains to be achieved because optimal controllers rely on simplifications that hinder their capability to predict future interactions. In fact, existing MPC implementations optimize contact forces as input variables [2], [5], control them only in a feed-forward sense [3], [4], [6], [7], or relax the rigid contact assumption for control [8] or modeling [9] purposes.

The difficulty of accounting for contact phenomena in an optimal control formulation [10], [11] lead roboticists to the *rigid contact* model [12]–[14] which fundamentally limits the ability to control contact forces. This limitation can be understood from a control-theoretic perspective: under the rigid contact assumption, the contact force is an output with direct input feedthrough, which means that the input torque affects *directly* the value of the contact force without any dynamics in between, making the system non-strictly proper [15]. In other words, the force can be written as a function of the state *and* the control input.

While it is possible in some cases to achieve output feedback control by exploiting the feedthrough as a feed-forward action [16], it remains a challenge in the general case [17], [18]. In particular, the feedback interconnection of such a system with a non-linearity (e.g. a model-predictive controller) is known to create an algebraic loop of which the well-posedness is not guaranteed [15], [19]. Digital implementations of MPC usually circumvent this pathology by discarding the direct feedthrough terms as physical systems have a finite bandwidth [20]. One possible way is to augment the state by introducing delay [21], for instance by treating input as states [22], [23] or by using output predictions in the input computation [24]. The former approach was used in [25] to derive contact-aware optimal policies, and in our previous approach [26] to enable torque-feedback in MPC. In this paper, we address this challenge by proposing a new MPC formulation that introduces contact dynamics rather than adding dynamics in the joint space.

B. Contributions

The combination of MPC and force control has already been investigated in other works, either by using an explicit model of the contact force [27]–[30] or in an impedance/admittance control fashion [31]–[34]. In contrast to those approaches, we propose to plan interaction forces based on an explicit model while not requiring any other controller than the MPC. To this end, the contact force is modeled as a linear spring-damper and treated as a state inside the optimization problem. This allows to make predictions about the future evolution of the interaction and to optimize directly force-feedback torque policies. This way we can inherit the benefits of MPC (reactive planning and task conflict resolution through cost optimization) while preserving good force tracking capabilities as in direct force control.

In our preliminary work [26], we revealed the fundamental limitation of classical MPC and proposed an original formulation that allows to incorporate joint torque feedback in optimal policies. While it confirmed the benefit of exploiting joint space effort measurements in contact tasks, we found that this approach is not the only way to endow robots with force awareness. Pursuing that objective, we introduce in this article a new MPC formulation that exploits contact force measurements in Cartesian space. While it comes at the price of relaxing the rigid contact assumption, it has the advantage of using measurements directly in the task space. This benefit is confirmed through experimental comparisons between the proposed approach, classical MPC and the aforementioned torque-feedback MPC [26]. Our contributions are the following:

[1]Tandon School of Engineering, New York University, Brooklyn, NY
[2]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse
[3]Artificial and Natural Intelligence Toulouse Institute (ANITI), Toulouse
This work was in part supported by the National Science Foundation grants 1932187, 2026479, 2222815 and 2315396.

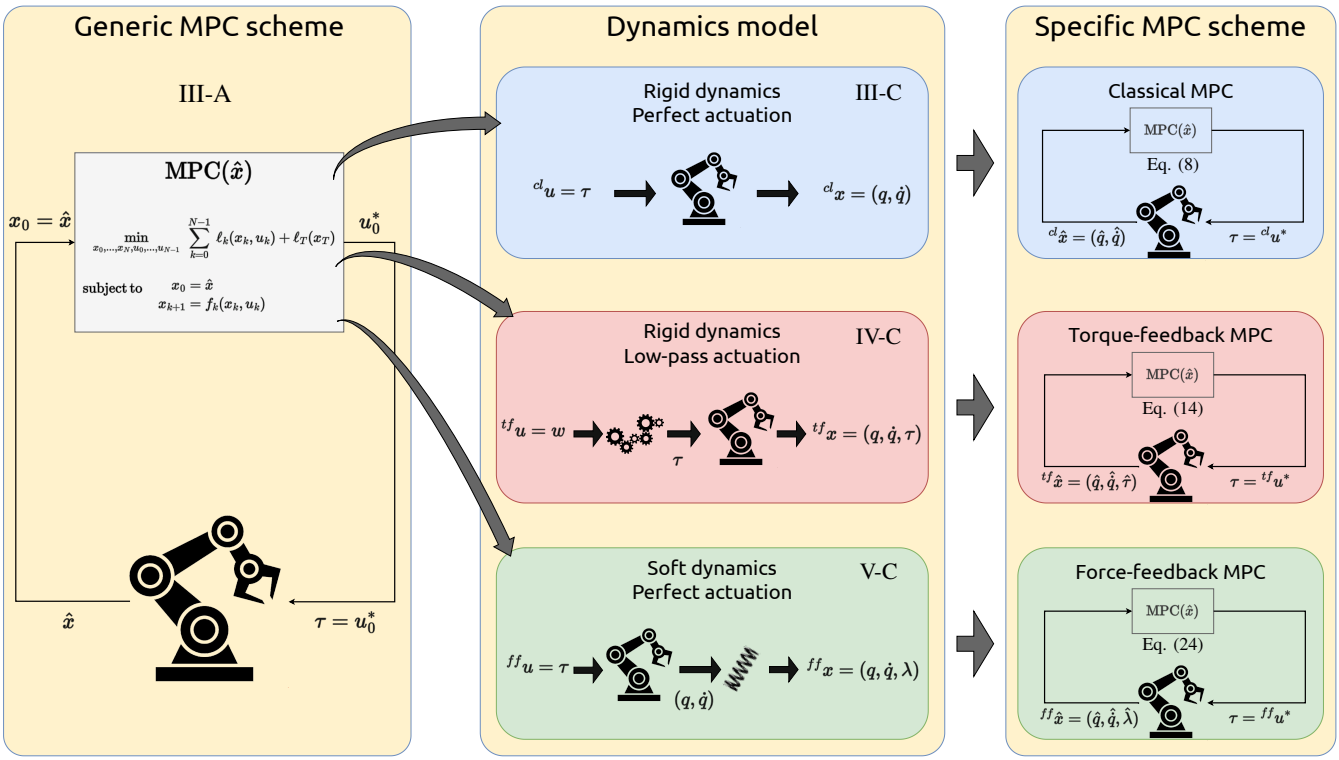


Fig. 1: Overview of the proposed MPC schemes. The generic MPC scheme (Section III-A) is agnostic to dynamics model f_k. The 3 MPC schemes presented in this paper are the result of a particular choice of state and input variables x, u: the classical MPC with position-velocity state and perfect torque actuation (Section III-C), the torque-feedback MPC scheme with position-velocity-torque state and low-pass actuation (Section IV-C) and the force-feedback MPC with position-velocity-force state and perfect torque actuation (Section V-C).

We introduce a novel MPC scheme using a visco-elastic force contact model and state augmentation (Section V). A reproduction of the experimental results of [26] on a torque-driven manipulator (iiwa) and a quantitative assessment of the force tracking capability of the torque-feedback MPC are proposed (Section VII). An experimental validation of the proposed force-feedback MPC and a comparison with classical and torque-feedback MPC are also presented. A detailed discussion on modeling assumptions and practical aspects of adding force in MPC on a real robot is offered (Section VIII). Finally, connections with existing force control literature and perspectives for MPC are established (Sections II, VIII).

II. RELATED WORK

A. Force control

The control of contact forces is a long-standing goal in robotics, as reflected by the early work of Whitney [35], and has been continuously subject to a great amount of research. This effervescence is due to the broad range of applications that necessitate controlled physical interactions, but also to the difficulty of designing reliable force control strategies [36]. Force control approaches can be broadly divided into 2 categories: indirect and direct approaches [37].

Direct force control techniques attempt to regulate the contact force directly through explicit feedback control [38], [39] based on force sensor measurements. The force control loop is generally combined with a motion controller in complementary task space directions as done in hybrid control [40], [41], or in parallel as done in parallel control [42]. The best basic force feedback control strategy has been shown - theoretically and experimentally - to be the integral controller, which enables accurate trajectory tracking [39]. But the high performance allowed by direct methods comes at the price of having to arbitrate manually the conflicts that may exist between the force and the motion tasks [43]. Besides, it inherently hides the friction phenomena [44], the exchange of mechanical work [45], and has serious theoretical pitfalls [46].

Indirect force control on the other hand attempts to regulate the dynamic relation between force and motion, a.k.a. the impedance. These techniques include notably impedance control [47], [48] and admittance control [35], [49]. The contact force is controlled *indirectly* through the specification of a desired impedance. While those techniques are relatively easy to implement and excel at realizing stable and compliant contact behaviors, their force tracking performance is inherently limited [50]: force regulation is done indirectly through impedance specification, which makes its quality depend on a priori unknown environment parameters [51], [52].

In this work, we propose a novel MPC formulation affording the explicit control of contact forces, akin direct methods.

	Classical MPC	Torque-feedback MPC	Force-feedback MPC
Section	III-C	IV-C	V-C
State	${}^{cl}x = (q, \dot{q})$	${}^{tf}x = (q, \dot{q}, \tau)$	${}^{ff}x = (q, \dot{q}, \lambda)$
Control	${}^{cl}u = \tau$	${}^{tf}u = w$	${}^{ff}u = \tau$
Actuation model	Perfect	Low-pass filter	Perfect
Force	$\lambda = \lambda^r$ (output (5b))	$\lambda = \lambda^r$ (output (10b))	$\lambda = \lambda^s$ (state (18))
Contact model	Rigid	Rigid	Soft

TABLE I: Variable definitions and notations for each MPC formulation. The ^{cl} superscript stands for the classical MPC of Section III-C, ^{tf} for the torque-feedback MPC of Section IV-C and ^{ff} for the force-feedback MPC of Section V-C. The force superscript ^{r} stands for the rigid contact model, while ^{s} denotes the soft contact force model.

B. Fundamental challenges in force control

There is a fundamental trade-off between stability and performance in force control. Colgate's analysis of force feedback [53] revealed that the contact instability observed by direct force control practitioners [38] is a special case of *coupled instability* due to a combination of the inertia scaling effect of force feedback [48] and non-collocation between sensors and actuators [54]. Stability can be enforced using a sufficient passivity condition [49], [55]–[58] that restricts the magnitude of the force feedback gain, hence the force control performance. Because some tasks may require precisely behaviors that are stable but not passive (e.g. bending, grinding), passivity can be regarded as a conservative criteria [59], although recent works suggest that it can be effectively accommodated in direct force control to allow high-bandwidth control [60]. In our work, the stability/performance trade-off is indirectly arbitrated through nonlinear optimization: the effective force feedback gain results from the minimization of a cost function.

The question of what particular impedance must be emulated to achieve a given task is very challenging. Impedance control is by nature agnostic to that question: it only provides tools to realize a *given* target impedance. Hence in practice, target behaviors are hand-designed through empirical fine-tuning to increase the task performance, which is tedious and vulnerable to uncertainties. The impedance selection problem was early recognized as a major challenge [61] to be overcome by adaptive control strategies [50], [62]–[64], environment impedance estimation [51], [65], optimization [61], [66] or learning [67], [68].

C. Toward optimization in force control

We like to think of optimal control as a way to automatize control gains synthesis, for instance by relating impedance modulation during contacts to a trade-off between disturbance rejection and measurement uncertainty [69]. This suggests that incorporating force in the optimization may result in an optimized impedance, trading off motion and force performance to achieve a higher-level objective [70]. The present work is in line with this interpretation.

As mentioned previously, the benefits of MPC and force control can be combined. This is done for instance in a direct force control fashion in [27], [28] where the contact force is treated explicitly as an output in the optimization using a linear spring model. MPC can also be used to bound the magnitude of contact forces generated by admittance control schemes in stiff environments [31]. The work in [32] extends this idea to nonlinear MPC with an emphasis on path-following. The approach in [33] proposes a task-space admittance control scheme that adds proportional-integral force term to the end-effector position reference in the cost function. Adaptive control is used in [34] to estimate impedance model parameters used in an MPC. Closer to our approach, the recent works in [29], [30] model interaction forces as linear spring models that are added directly into the state dynamics. In [29], the contact force estimated from joint torques is fed-back directly into the MPC that optimizes joint-space PD reference trajectories. In [30], a Hertz contact model is used in the trajectory optimization problem. A joint space impedance and a Cartesian admittance control allow to control forces on a soft tissue in MPC at $5Hz$. We propose instead to achieve high-frequency predictive force feedback *without* any additional stabilizing controller.

III. BACKGROUND

In this section we recall the classical MPC and the torque-feedback MPC based on actuation modeling.

A. Model-predictive control

We consider the following Optimal Control Problem (OCP)

$$\min_{u(.),x(.)} \quad \int_0^T \ell\left(x(t), u(t), t\right) \mathrm{d}t + \ell_T\left(x(T)\right) \tag{1a}$$

$$\text{s.t.} \quad x(0) = \hat{x}, \tag{1b}$$

$$\dot{x}(t) = f\left(x(t), u(t), t\right) \tag{1c}$$

where $\hat{x}$ is the measured state, u the control input and f is the dynamics model, ℓ, ℓ_T the running and terminal costs. The OCP (1) is transcripted into the following Non-Linear Program (NLP)

$$\min_{\substack{x_0,\dots,x_N \\ u_0,\dots,u_{N-1}}} \quad \sum_{k=0}^{N-1} \ell_k(x_k, u_k) + \ell_N(x_N) \tag{2a}$$

$$\text{s.t.} \quad x_0 = \hat{x}, \tag{2b}$$

$$x_{k+1} = f_k(x_k, u_k) \tag{2c}$$

where ℓ_k, ℓ_N and f_k denote the discretized costs and dynamics with sampling step Δt (typically resulting from a semi-implicit Euler integration scheme). The solution is a locally optimal control sequence $u_0^*, ..., u_{N-1}^*$ from which we only send the first element u_0^* to the robot low-level controllers. This NLP can be solved efficiently using standard nonlinear optimization such as Sequential Quadratic Programming (SQP) [71]

provided that the sparsity induced by time is exploited in the resolution [72][1].

This generic numerical optimal control formulation encompasses the various MPC controllers derived in this paper. We will propose alternative definitions of the state x and control u variables, which will lead to specific definitions of the continuous-time (1c) and discrete-time (2c) dynamics constraints, hence to distinct model-predictive controllers. The state and control variables definitions of all MPC schemes are recapitulated in Table I.

B. Rigid contact dynamics

A *rigid contact* is a kinematic constraint between the robot and the environment. The equations of motion of a fully-actuated robot in rigid contact with the environment can be derived from the Karush-Kuhn-Tucker (KKT) conditions of the convex optimization problem corresponding to Gauss' least constraint principle [73]

$$\min_{\ddot{q}} \ \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)} \tag{3a}$$

$$\text{s.t. } J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0 \tag{3b}$$

where $q, \dot{q} \in \mathbb{R}^n$ are the vectors of joint positions and velocities, $M(q) \in \mathbb{S}^n_+$ is the generalized inertia matrix, $J(q) \in \mathbb{R}^{m\times n}$ is the Jacobian of the m-dimensional contact, $\ddot{q}_f = M(q)^{-1}(\tau - h(q,\dot{q})) \in \mathbb{R}^n$ is the free acceleration, $h(q,\dot{q}) \in \mathbb{R}^n$ is the vector of centrifugal, Coriolis and gravity forces and $\tau \in \mathbb{R}^n$ is the vector of joint torques. As explained in [74], the KKT conditions of (3), namely

$$\begin{bmatrix} M(q) & J(q)^T \\ J(q) & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ -\lambda^r \end{bmatrix} = \begin{bmatrix} \tau - h(q,\dot{q}) \\ -\dot{J}(q)\dot{q} \end{bmatrix} \tag{4}$$

reveal the generalized contact forces $\lambda^r \in \mathbb{R}^m$ as the Lagrange multipliers associated with the rigid contact constraint (3b).

C. Classical MPC

In this formulation, the continuous-time dynamics constraint f is defined from the solution map of the KKT conditions (4). It corresponds a controlled dynamical system with state ${}^{cl}x \triangleq (q,\dot{q})$ and control input ${}^{cl}u \triangleq \tau$ (we omit the dependency in t for readibility)

$${}^{cl}\dot{x} = f\left({}^{cl}x, {}^{cl}u\right) \tag{5a}$$

$$\lambda = \lambda^r\left({}^{cl}x, {}^{cl}u\right) \tag{5b}$$

where f maps joint positions, velocities and torques to the constrained joint accelerations $\ddot{q}$ (primal solution of (4))

$$f\left({}^{cl}x, {}^{cl}u\right) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)\left(\tau - h(q,\dot{q}) + J(q)^T\lambda\right) \end{bmatrix} \tag{6}$$

and λ^r is the rigid contact force (dual solution of (4)). In this formulation [75], the contact force λ appears as an output of the system. Therefore, the cost function can include a force task with the following form

$$\ell\left({}^{cl}x, {}^{cl}u\right) = \|\lambda^r\left({}^{cl}x, {}^{cl}u\right) - \bar{\lambda}\|^2 + \text{other terms} \tag{7}$$

where $\bar{\lambda}$ is a reference force.

The discretized dynamics constraint f_k can be obtained using an Euler semi-implicit scheme with sampling time Δt

$$f_k\left({}^{cl}x_k, {}^{cl}u_k\right) = {}^{cl}x_k + \begin{bmatrix} \Delta t & \Delta t^2 \\ 0 & \Delta t \end{bmatrix} f\left({}^{cl}x_k, {}^{cl}u_k\right) \tag{8a}$$

$$\lambda_k = \lambda^r\left({}^{cl}x_k, {}^{cl}u_k\right) \tag{8b}$$

At each MPC cycle, the NLP (2) is initialized with the latest position-velocity state measurement ${}^{cl}\hat{x} = (\hat{q},\hat{\dot{q}})$ and solved using as a dynamics constraint the discrete dynamics derived in (8). The joint torque τ sent to the robot is then selected as the first optimal control ${}^{cl}u_0^*$ output by the NLP resolution.

IV. TORQUE-FEEDBACK MPC

We first recall the fundamental incapacity of classical MPC to do force feedback and how the formulation proposed in our previous work [26] allows to overcome this issue.

A. The limitation of classical MPC

The classical formulation of Section III-C does not allow force predictive feedback control because the contact force and control torque are algebraically coupled. While λ is used as a *prediction*, it is possible to choose a control action u. But if λ is used as a measurement (like the state x), it can be seen from (5b) (or (8b) in discrete time) that u is completely determined and cannot be chosen. The force λ appears as the output of a nonlinear system in with instantaneous transfer from the input since $\frac{\partial\lambda}{\partial u} = (JM^{-1}J^T)^{-1}JM^{-1}$ is non-zero. Attempting to control λ with some policy $u(\lambda)$ would obviously create an algebraic loop as u and λ would influence each other instantaneously.

As previously mentioned, this corresponds in control theory terms to a non-zero input-output feedthrough that makes the system non-strictly proper. One way to render such a system proper (and thus to allow force feedback) is to delay the output with respect to the system input ([21], p. 37). This approach was followed in [25] and in our previous work [26] by modeling an imperfect torque actuation. Another way is to relax the rigid contact assumption, as done for instance in [28], which also produces the needed decoupling (the delay then comes from the flexibility). We propose in this paper to follow this second track and introduce the force feedback MPC based on visco-elastic contact modeling. But first, let's recall our previously proposed MPC based on actuation dynamics, which we will use as a basis of comparison.

B. Low-pass actuation dynamics

In our previous work [26], we proposed to model the actuation dynamics as a first-order low-pass filter. The input

[1]Note that nonlinear inequality constraints on the state and control input can be easily be added to this formulation as shown in [72]

torque w is then related to the actuation torque τ according to the linear first-order differential equation

$$\dot{\tau} = \omega_c(w - \tau) \tag{9}$$

where $\omega_c > 0$ is the cut-off angular frequency in $\mathrm{rad\,s^{-1}}$, i.e. $\omega_c = 2\pi f_c$ where f_c is the ordinary cut-off frequency in Hz. We like to think of this model as an abstraction of the actuation, not an accurate model. It is simple yet generic enough to capture the linear behavior of many actuators. More complex models could be used without any changes to the rest of the approach.

Note that this approach presents similarities to the one proposed in [76] which enforces the limitation of the actuation bandwidth limitation by using a frequency-dependent cost function. In that sense, our approach resembles the special case of a low-pass shaping function. But the conceptual difference is that frequency-shaping leaves the relation between state and control unchanged during the optimization, while our controller reasons over higher dimensional dynamics, which enables to naturally derive control policies that depend on torque measurements.

C. Torque-feedback MPC formulation

The above actuation model allows to augment the classical dynamics of Section III-C. We define the state of the system as ${}^{tf}x \triangleq (q, \dot{q}, \tau)$, which includes classical state ${}^{cl}x$ *and the joint torques* τ. The control input to be computed by the MPC is defined as the unfiltered torque ${}^{tf}u \triangleq w$. The rigid contact model is however left unchanged. The full continuous-time dynamics model then reads

$$ {}^{tf}\dot{x} = f\left({}^{tf}x, {}^{tf}u\right) \tag{10a}$$

$$ \lambda = \lambda^r\left({}^{tf}x\right) \tag{10b}$$

where the dynamics constraint f now contains the low-pass filter (9):

$$f\left({}^{tf}x, {}^{tf}u\right) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)\left(\tau - h(q,\dot{q}) + J(q)^T\lambda\right) \\ \omega_c\left(w - \tau\right) \end{bmatrix} \tag{11}$$

and $\lambda^r\left({}^{tf}x\right)$ is the same rigid contact force as the one defined in the classical MPC (dual solution of the KKT system (4)). Note that in this formulation, the algebraic coupling between λ and τ still exists (we are still working under the rigid contact model assumption) but it is no longer an issue since the optimal control ${}^{tf}u = w$ can be freely computed *after* a contact force is measured. In this formulation, the cost function of the OCP can include a contact force task similarly to (7)

$$\ell\left({}^{tf}x, {}^{tf}u\right) = \|\lambda^r\left({}^{tf}x\right) - \bar{\lambda}\|^2 + \text{other terms} \tag{12}$$

with the main difference that the rigid contact force no longer depends on the system input ${}^{tf}u$, but only on the state ${}^{tf}x$.

The filter discretized under zero-order hold takes the form of an exponential moving average

$$\tau_{k+1} = \alpha\tau_k + (1 - \alpha)w_k \tag{13}$$

where $\alpha = e^{-\omega_c \Delta t}$. Note that α parameter provides an intuitive understanding of the asymptotic behavior of the actuation model: when $f_c \to \infty$, $\alpha \to 0$ and there is no filtering so w goes entirely through[2]. When $f_c \to 0$, $\alpha \to 1$ and the filtering is maximal so w is entirely blocked. However in practice, the value of f_c is upper bounded by the Nyquist frequency $\frac{f_s}{2} = \frac{1}{2\Delta t}$ to avoid aliasing phenomena. The position-velocity dynamics is then integrated using a semi-implicit Euler as in the classical MPC case, while the torque dynamics uses (13)

$$f_k\left({}^{tf}x_k, {}^{tf}u_k\right) = {}^{tf}x_k + \begin{bmatrix} \Delta t & \Delta t^2 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \frac{1-\alpha}{\omega_c} \end{bmatrix} f\left({}^{tf}x_k, {}^{tf}u_k\right) \tag{14a}$$

$$\lambda_k = \lambda^r\left({}^{tf}x_k\right) \tag{14b}$$

In practice, the torque sent to the robot at each MPC cycle is not the optimal computed *unfiltered* torque ${}^{tf}u_0^*$ because it may be too aggressive to be used when the optimizer overestimates the filtering effect of the actuation. It is safer to use instead the optimal *filtered* torque predictions: hence we send an interpolation $\tilde{\tau}$ between the initial (measured) torque $\tau_0^* = \hat{\tau}$ and predicted (optimal) *filtered* torque τ_1^*

$$\tilde{\tau} = \tau_0^* + \epsilon(\tau_1^* - \tau_0^*) \tag{15}$$

where ϵ is the ratio between the OCP sampling step Δt and the MPC re-planning step duration. The interpolation is used because the prediction τ_1^* may lie too far in the future when the OCP sampling frequency is lower than the MPC re-planning frequency (which is usually the case in practical implementations). This MPC scheme has been shown to outperform the classical MPC formulation in contact tasks thanks to its ability to re-plan based on measured joint torques [26].

V. FORCE FEEDBACK MPC

In this section we introduce the soft contact model and our new formulation of the OCP that allow direct force feedback from Cartesian space measurements.

A. Visco-elastic contact dynamics

The visco-elastic contact dynamics models the contact force by a linear spring-damper

$$\lambda^s(q, \dot{q}, p_c) \triangleq -K\Delta p(q, p_c) - B\dot{p}(q, \dot{q}) \tag{16}$$

where $\Delta p(q, p_c) = p(q) - p_c$ is the end-effector deflection, $p(q) \in \mathbb{R}^m$ is the end-effector Cartesian pose given by forward kinematics, $p_c \in \mathbb{R}^m$ is the contact anchor point, and $\dot{p}(q, \dot{q}) = J(q)\dot{q}$ is the end-effector Cartesian velocity given by differential forward kinematics. The matrices $K, B \in \mathbb{R}^{m\times m}$ are the stiffness and damping of the environment (diagonal and positive definite), assumed to be fixed and known. Notice that given joint positions, joint velocities and the anchor location, the contact force is fully determined.

[2]This corresponds to a discrete system with pure delay $\tau_{k+1} = w_k$.

B. Naive approach to visco-elastic contact

The visco-elastic model immediately leads to a first MPC with contact feedback, which we introduce now but that we will explain not to be fully satisfactory. The rigid contact force in (5) can be replaced by the visco-elastic contact force (16) which leads to the following continuous-time dynamics model with classical state ${}^{cl}x$ and input ${}^{cl}u$

$$ {}^{cl}\dot{x} = f\left({}^{cl}x, {}^{cl}u\right) \tag{17a} $$

$$ \lambda = \lambda^s\left({}^{cl}x, p_c\right) \tag{17b} $$

where f is the same as (6). The only difference between (17) and the classical formulation (5) is the output equation describing the contact force evolution: contrary to (5b), the contact force in (17b) does not depend on the control input ${}^{cl}u$. Although this reformulation clearly breaks the algebraic coupling with the control input, it still does not allow to take into account the measured force: measuring the state fully determines the contact force. In other words, (16) acts as a *measurement* model. One possibility to exploit it could use a Kalman filter to estimate the state.

Yet we would rather like to be able to inject new information into the optimizer at each measurement cycle, so that the optimal trajectories are based on both visco-elastic predictions *and* sensed efforts. For this, we propose another formulation that consists in augmenting the classical state with the contact forces.

C. Force feedback MPC formulation

We propose now to include the measured contact force directly in the system state and to use the visco-elastic contact equation (16) as a *prediction* model. The augmented state is defined as ${}^{ff}x \triangleq (q, \dot{q}, \lambda)$ which includes the classical state ${}^{cl}x$ and the measured contact force λ, while the control input is the joint torque ${}^{ff}u \triangleq \tau$ like in the classical formulation. We will assume that the anchor point is not moving, i.e. $\dot{p}_c = 0$. This corresponds to a sticking contact model, which is reasonable as long as there is no slipping. Under this assumption, the time derivative of (16) is given by

$$ \dot{\lambda}^s\left({}^{ff}x, {}^{ff}u\right) = -K\dot{p} - B\ddot{p} \tag{18} $$

where $\ddot{p} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$ is the end-effector Cartesian acceleration, and the joint acceleration $\ddot{q}$ is computed using the forward rigid-body dynamics. This reformulation leads to the full continuous-time dynamics model

$$ {}^{ff}\dot{x} = f\left({}^{ff}x, {}^{ff}u\right) \tag{19} $$

where f includes the classical forward rigid-body dynamics $\ddot{q}$ *and* the visco-elastic force dynamics defined in (18)

$$ f\left({}^{ff}x, {}^{ff}u\right) = \begin{bmatrix} \dot{q} \\ M^{-1}\left(\tau - h + J^T\lambda\right) \\ -KJ\dot{q} - B\left(JM^{-1}\left(\tau - h + J^T\lambda\right) + \dot{J}\dot{q}\right) \end{bmatrix} \tag{20} $$

where we dropped the dependencies in $q, \dot{q}$ for readibility. This model expresses the direct measure of λ (which will be the case in our experiments thanks to a F/T sensor attached to the end-effector) and uses the visco-elastic interaction model to predict its previewed evolution. Now the MPC is able to take decisions based on the predicted whole-body behaviour while being directly informed of the measured forces. In this formulation, a contact force task is formulated as an integral part of the state cost

$$ \ell\left({}^{ff}x, {}^{ff}u\right) = \|\lambda - \bar{\lambda}\|^2 + \text{other terms} \tag{21} $$

D. Model derivatives and transcription

The optimization algorithm requires the derivatives of the dynamics model (20) w.r.t. state ${}^{ff}x$ and control ${}^{ff}u$. The rigid-body dynamics part requires $\frac{\partial\ddot{q}}{\partial q}, \frac{\partial\ddot{q}}{\partial\dot{q}}, \frac{\partial\ddot{q}}{\partial\tau}$, which are classically known in closed-form [77], and $\frac{\partial\ddot{q}}{\partial\lambda}$ of which we explicit the computation

$$ \frac{\partial\ddot{q}}{\partial\lambda} = M^{-1}J^T \tag{22} $$

Note that the forward rigid-body dynamics $\ddot{q}$ is a function of $(q, \dot{q}, \tau, \lambda)$. We insist here on the fact that unlike in classical MPC case, λ is **not** a function of $q, \dot{q}$ in the above expression: as part of the state of the system ${}^{ff}x$, the variables $q, \dot{q}, \lambda$ are *simultaneously* measured and therefore independent at this stage. The visco-elastic force dynamics part requires the derivatives $\dot{\lambda}^s$ w.r.t. state ${}^{ff}x$ and control ${}^{ff}u$. They are computed using the chain rule with end-effector velocity and acceleration

$$ \frac{\partial\dot{\lambda}^s}{\partial q} = -K\frac{\partial\dot{p}}{\partial q} - B\left(\frac{\partial\ddot{p}}{\partial q} + \frac{\partial\ddot{p}}{\partial\ddot{q}}\frac{\partial\ddot{q}}{\partial q}\right) \tag{23a} $$

$$ \frac{\partial\dot{\lambda}^s}{\partial\dot{q}} = -K\frac{\partial\dot{p}}{\partial\dot{q}} - B\left(\frac{\partial\ddot{p}}{\partial\dot{q}} + \frac{\partial\ddot{p}}{\partial\ddot{q}}\frac{\partial\ddot{q}}{\partial\dot{q}}\right) \tag{23b} $$

$$ \frac{\partial\dot{\lambda}^s}{\partial u} = -B\frac{\partial\ddot{p}}{\partial\ddot{q}}\frac{\partial\ddot{q}}{\partial\tau} \tag{23c} $$

$$ \frac{\partial\dot{\lambda}^s}{\partial\lambda} = -B\frac{\partial\ddot{p}}{\partial\ddot{q}}\frac{\partial\ddot{q}}{\partial\lambda} \tag{23d} $$

where the terms $\frac{\partial\ddot{p}}{\partial q}, \frac{\partial\ddot{p}}{\partial\dot{q}}, \frac{\partial\ddot{p}}{\partial\ddot{q}}$ are computed using recursive algorithms [77].

The discretized dynamics model can be obtained using a semi-implicit Euler integration scheme similarly to Sections III-C, IV-C

$$ {}^{ff}x_{k+1} = {}^{ff}x_k + \begin{bmatrix} \Delta t & \Delta t^2 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} f\left({}^{ff}x_k, {}^{ff}u_k\right) \tag{24} $$

At each MPC cycle, the NLP is re-solved based on the measured state ${}^{ff}x$ and the first optimal joint torque ${}^{ff}u_0^*$ is sent to the robot.

VI. SIMULATION STUDY

In this section we compare the performances of the 3 formulations previously introduced in simulated polishing tasks. The benefit of force feedback in the position and force tracking accuracy becomes clear under imperfect actuation and contact modeling. Further, we highlight the effect of the low-level torque controller in the task performance by simulating the polishing task with and without this module. This reveals the weaknesses of both the classical MPC and the torque-feedback MPC formulations.

A. Task formulation

1) Frame conventions and sliding contact: In order to achieve the polishing task we need a $1D$ (i.e. *pure sliding*) contact model in order to allow only forces in the table normal direction and motions in the lateral directions. Moreover, this normal force is expressed in the *centered* frame convention: the origin of the reference frame coincides with the contact point but its axes are aligned with the table axes at all times. The derivatives of the forward dynamics must then be expressed in these moving coordinates. Such a model can be easily obtained from a $3D$ contact model by using projections and the derivations described in our previous work [78].

2) Cost function: The task is to apply a constant vertical force while drawing a circle on the horizontal plane. For the sake of clarity we split the cost function in 2 parts: a common part ℓ^0 which is the same in all 3 controllers and a controller-specific part which contains additional terms that are specific to the controller. The common cost ℓ^0 reads

$$\begin{aligned}\ell^0 = \; & c_1\|q-\bar{q}\|^2_{Q_1} && (25.1)\\ & + c_2\|\dot{q}\|^2_{Q_2} && (25.2)\\ & + c_3\|\tau-\tau_g(q,\lambda)\|^2_{Q_3} && (25.3)\\ & + c_4\|p(q)-\bar{p}(t)\|^2_{Q_4} && (25.4)\\ & + c_5\|\dot{p}(q,\dot{q})\|^2_{Q_5} && (25.5)\\ & + c_6\|\lambda-\bar{\lambda}(t)\|^2_{Q_6} && (25.6)\end{aligned}$$

where $(c_i, Q_i)_{i=1..6}$ are positive scalar weights and positive diagonal activation matrices, $\bar{q}$ is a static reference joint configuration, $\tau_g(q,\lambda)$ is the gravity compensation torque under external forces $g(q) - J(q)^T\lambda$, $\bar{p}(t), \bar{\lambda}(t)$ are time-varying reference of end-effector position and contact force respectively. The term (25.1) is a joint configuration regularization term, (25.2) a joint velocity regularization term, (25.3) a torque regularization (around gravity torque), (25.4) an end-effector position tracking term, (25.5) an end-effector velocity regularization term, (25.6) a contact force tracking term. Note that at this stage, we had to slightly abuse notation in the cost function definition since ℓ^0 can clearly be written in either one of the following ways

$$\ell^0\left({}^{cl}x, {}^{cl}u\right) = \ell^0\left({}^{tf}x, {}^{tf}u\right) = \ell^0\left({}^{ff}x, {}^{ff}u\right) \quad (26)$$

depending on the definition of the state and input variables. In particular, the force cost term (25.6) takes the form (7), (12) or (21) depending on the MPC formulation being considered. We can now define each controller's complete cost function based on ℓ^0. The only changes in the cost function are regularization terms on the new variables due to state augmentation. For the classical MPC, the full cost function reads

$$\ell\left({}^{cl}x, {}^{cl}u\right) = \ell^0\left({}^{cl}x, {}^{cl}u\right) \quad (27)$$

For the torque-feedback MPC, the full cost function includes an additional regularization term on the computed torque

$$\ell\left({}^{tf}x, {}^{tf}u\right) = \ell^0\left({}^{tf}x, {}^{tf}u\right) + c_w\|{}^{tf}u - \tau_g(q,\lambda)\|^2_{Q_w} \quad (28)$$

For the force-feeback MPC, the cost function includes an additional regularization term on the rate of change of the contact force

$$\ell\left({}^{ff}x, {}^{ff}u\right) = \ell^0\left({}^{ff}x, {}^{ff}u\right) + c_\lambda\|\dot{\lambda}\|^2_{Q_\lambda} \quad (29)$$

3) Task phases management: The task is divided into several phases with fixed time durations: stand still, reach the surface of the table, apply a normal force, circle motion, stop. The cost function weights and references are updated online based on these time-based switches.

B. Simulation setup

The simulation scheme presented in Figure 2 is explained here in details.

1) Software: Rigid-body dynamics computations are performed using the `Pinocchio` library [77]. The OCPs are formulated using the `Crocoddyl` optimal control library [79] and solved using our efficient SQP solver available in `mim_solvers`[3] [72]. The force-feedback and torque-feedback OCPs described in this work are implemented in the open-source library `force_feedback_mpc`[4]. The benchmarks and experiments can be reproduced using our dedicated public repository `force-feedback`[5]. We use `PyBullet` as a physics simulator ("RBDS" in Figure 2).

2) Contact, sensing and actuation modeling: We assess the performance of the controllers under a perturbed contact model and imperfect actuation and sensing. The contact surface is tilted by an angle θ unknown to the controller. The sensing model adds noise and delays to the measured joint positions and velocities output by the physics simulator

$$\hat{q}(t) = q^{\text{sim}}(t-\delta_s) + \nu_q \quad (30a)$$
$$\hat{\dot{q}}(t) = \dot{q}^{\text{sim}}(t-\delta_s) + \nu_{\dot{q}} \quad (30b)$$

while the actuation models adds an affine bias, noise, delay, dry friction and stiction to the motor torques

$$\hat{\tau}(t) = a\tau^m(t-\delta_s) + b - \mu_s\,\text{sign}(\hat{\dot{q}}(t)) - \mu_v\hat{\dot{q}}(t) + \nu_\tau \quad (31)$$

where $\hat{q}, \hat{\dot{q}}, \hat{\tau}$ are measured signals, $q^{\text{sim}}, \dot{q}^{\text{sim}}$ are computed by rigid-body dynamics simulation, τ^m is the motor torque output by the robot low-level controller, $\nu_q, \nu_{\dot{q}}, \nu_\tau$ are Gaussian random variables with 0 mean and variances $\sigma_q, \sigma_{\dot{q}}, \sigma_\tau$ respectively, δ_s is the sensing delay, (a, b) is a bias uniformly distributed over $[\underline{a}, \overline{a}] \times [\underline{b}, \overline{b}]$. The coefficients μ_s, μ_v encode respectively static and viscous friction in the joints. The contact surface stiffness and damping coefficients of `PyBullet` are set to 10^4 and 10^2 respectively, and the lateral friction coefficient to 0.5. The surface tilt angle θ is ranging from $-6°$ to $+6°$. The actuation bias (a, b) is uniformly drawn from $[0.8, 1.2] \times [-2, +2]$. The sensing delay is selected as $\delta_s = 1$ simulation cycle, the random noise parameters as $\sigma_q = \sigma_{\dot{q}} = 10^{-3}$ and σ_τ as a small percentage of the joint torque limits provided by the manufacturer. The sign in the static joint friction term is approximated by a hyperbolic

[3] https://github.com/machines-in-motion/mim_solvers
[4] https://github.com/machines-in-motion/force_feedback_mpc
[5] https://github.com/skleff1994/force-feedback

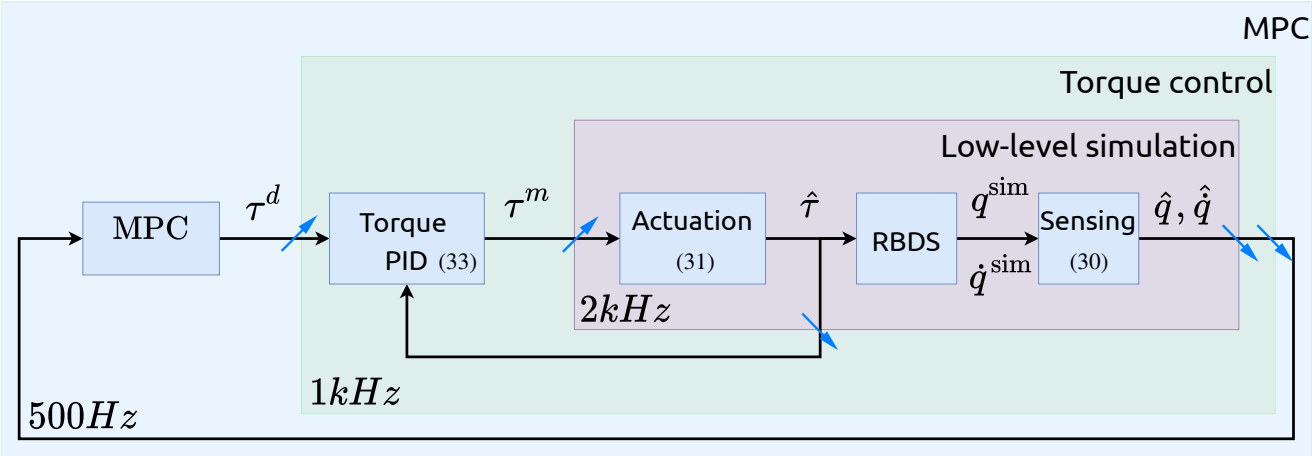


Fig. 2: Schematic of the simulation setup used in Section VI-B for the classical MPC, including low-level torque control. Gray arrows corresponds to zero-order holds (ascending) or anti-aliasing filters (descending). The torque controller is a PID controller plus feedforward (τ^d). The actuation block simulates the combined effects of motor inertia, transmission, torque sensing, etc. The RBDS block can be any rigid-body physics simulator. The sensing block simulates sensor dynamics and noise.

tangent and we use $\mu_s = 1$ and $\mu_v = 0.5$. In the case of the force-feedback MPC, the contact force is also perturbed with Gaussian noise and sensing delay

$$\hat{\lambda}(t) = \lambda^{\text{sim}}(t - \delta_s) + \nu_\lambda \tag{32}$$

with $\nu_\lambda \simeq \mathcal{N}(0, \sigma_\lambda^2)$. The low-level torque control is simulated by a torque PID controller with feedforward

$$\begin{aligned} \tau^m(t) = \; & \tau^d(t - \delta_o) && \text{(33a)} \\ & - K_P \left(\hat{\tau}(t) - \tau^d(t - \delta_o) \right) && \text{(33b)} \\ & - K_I \int_0^t \left(\hat{\tau}(s) - \tau^d(s - \delta_o) \right) \mathrm{d}s && \text{(33c)} \\ & - K_D \hat{\dot{\tau}}(t) && \text{(33d)} \end{aligned}$$

where τ^d is the desired torque output by the MPC, K_P, K_I, K_D are PID gains (manually tuned by fine trials and errors) and δ_o is the delay due to the OCP resolution set to $\delta_o = 5\,\text{ms}$.

3) MPC settings: The 3 MPC controllers run at $500Hz$. The horizon is 5 nodes of 3 ms. The force-feedback MPC has $K = 10^3$ and $B = 10^2$. The cutoff frequency of the low-pass filter model in the torque-feedback MPC is set to $f_c = 50\,\text{Hz}$. The same parameters are used on real hardware and discussed in Section VII.

C. Results

In order to compare the performances of the 3 controllers and how they are affected by the low-level torque control, we run batches of MPC simulations with and without the torque PID controller. For each simulation, an actuation model bias (a, b) and a tilt angle θ are randomly selected. The desired end-effector circle trajectory has a diameter of 14 cm and a velocity of $3\,\text{rad}\,\text{s}^{-1}$, and the force reference is 50 N. For each controller, the cost function weights were selected independently to achieve the best empirical performance. The performance is assessed using 4 metrics: the tangential position and normal force Mean Absolute Errors (MAE), namely E_P and E_λ, the maximum normal force error magnitude λ_{max} and the maximum percentage of time spent not in contact t_{break}. These quantities are computed over all tilt angles and bias parameters samples :

$$\lambda_{\text{max}} = \max_{(a,b),\theta} \|\hat{\lambda}(t)\| \tag{34}$$

$$t_{\text{break}} = \max_{(a,b),\theta} \frac{\text{\# simulation cycles where } \hat{\lambda} \leq \gamma}{\text{\# simulation cycles}} \tag{35}$$

with $\gamma = 10^{-3}\text{N}$.

1) With PID: The PID gains are set to the same values for all 3 controllers. They were selected manually to achieve the best torque tracking performance under the contact, sensing and actuation models described previously. Table III reports the performance metrics. The torque-feedback MPC and the force-feedback MPC both outperform the classical MPC in terms of tracking performance. Furthermore, the force-feedback MPC achieved lower force and position MAE and a lower λ_{max} than the torque-feedback MPC, thanks to its ability to control directly the Cartesian force.

2) Without PID: When the low-level torque PID controller is removed, the motor torque τ^m is set to the feedforward torque τ^d (with delay δ_o) without any feedback from the measured joint torque $\hat{\tau}$, so $\hat{\tau}$ can differ significantly from τ_d due to imperfect actuation, sensing and contact modeling. In this case, the force tracking performances of all controllers are degraded as shown in Table II. Among the three controllers, the torque-feedback MPC appears to be the most impacted as its force MAE becomes greater than the one of classical MPC, and it breaks contact up to more than 32 % of the time. Interestingly, the position tracking accuracy improved for the classical and torque-feedback MPC. This trade-off in favor

	Classical	Torque-feedback	Force-feedback
E_P (mm)	5.75 ± 0.06	$\mathbf{2.44 \pm 0.01}$	4.59 ± 0.01
E_λ (N)	25.52 ± 4.09	33.98 ± 4.42	$\mathbf{10.55 \pm 4.40}$
λ_{max} (N)	$\mathbf{59.36}$	105.57	110.95
t_{break} (%)	0.51	32.75	$\mathbf{0.00}$

TABLE II: MAE of the normal force and end-effector position for polishing tasks for randomized table tilting angles and actuation model parameters, without low-level torque control.

	Classical	Torque-feedback	Force-feedback
E_P (mm)	7.94 ± 1.43	3.87 ± 0.65	$\mathbf{3.07 \pm 0.01}$
E_λ (N)	15.21 ± 5.70	12.44 ± 4.16	$\mathbf{2.04 \pm 0.01}$
λ_{max} (N)	185.75	211.00	$\mathbf{111.97}$
t_{break} (%)	0.00	0.00	$\mathbf{0.00}$

TABLE III: MAE of the normal force and end-effector position for polishing tasks for randomized table tilting angles and actuation model parameters, with low-level torque control.

of position accuracy can be explained by the lower contact friction which opposes less motion of the end-effector: while the low-level torque control has the effect of generating the friction inherently required by the polishing task, its absence makes it easier to track a desired position by relaxing the force task. This analysis is also consistent with the lower maximum force observed for the torque-feedback and classical MPC: the robot without torque PID is basically "pushing" less into the table.

In order to analyze the performance degradation, Table IV reports the changes in the force and torque tracking MAE due to the removal of the low-level torque controller: we computed ΔE_λ and ΔE_τ, where Δ makes the difference of MAE between Table II and Table III. The torque tracking MAE E_τ was computed between the desired torque τ^d and the measured torque $\hat{\tau}$.

3) Discussion: In both cases (with and without torque PID), increasing force gain in the classical MPC and torque feedback did not improve the force tracking accuracy. This is a direct consequence of the inability of the rigid contact model to accept predictive feedback of measured efforts. Interestingly, the torque-feedback MPC is the most negatively impacted by the removal of the low-level torque PID, which suggests a higher correlation between the torque tracking accuracy and the task performance. This correlation is consistent with the results reported in Table IV.

In summary, our simulation study shows the superiority of the force-feedback MPC in both position and force tracking. This controller is also the least impacted by poor torque tracking accuracy. On the other hand, while the torque-feedback MPC outperforms the classical MPC under good low-level torque control, it becomes less performing when the measured torque differs significantly from the desired torque.

VII. EXPERIMENTAL VALIDATION

We report here hardware experiments on a torque-controlled manipulator that validate the proposed approach to achieve dynamic contact tasks, and its superiority over the classical MPC and the torque-feedback MPC. The accompanying video contains recordings of the various experiments presented in this section and additionally illustrates the performance and robustness of the proposed MPC formulation.

	Classical	Torque-feedback	Force-feedback
ΔE_λ (N)	10.31 ± 6.85	21.55 ± 4.95	$\mathbf{8.51 \pm 1.87}$
ΔE_τ (N m)	2.24 ± 0.08	2.20 ± 0.07	$\mathbf{1.95 \pm 0.26}$

TABLE IV: Change in the force and torque tracking MAE due to the removal of the low-level torque controller.

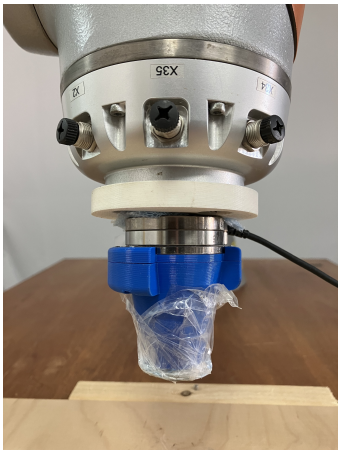


Fig. 3: Custom end-effector mount piece

A. Experimental setup

1) Software: The same software as in the simulation study (VI-B1) was used for the robot experiments.

2) Hardware: We use the torque-controlled Kuka iiwa LBR 1480. The contact surface used in all experiments is a flat wooden piece. The Cartesian wrenches are measured with an FTD-Mini-45 SI-290-10 sensor at the robot end-effector. A custom end-effector plastic piece is used to mount the FT sensor and to protect it from hard impacts with a soft foam ball taped to its tip acting as a damper (Figure 3).

3) Soft contact modeling: Let us recall that for the polishing task, the force-feedback MPC uses a unidirectional ($1D$) bilateral model, i.e. force is not enforced to be strictly positive, and the contact and stiffness parameters K, B scalars associated with the stiffness and damping in the normal direction to the table. We found that the location of the anchor point p_c is not important, as Δp doesn't affect the force dynamics (only the contact point motion does, as seen from (18)). We experimentally determined K to be around $1.3e4$ but this approximation is most likely not accurate as K jumps abruptly when the robot contacted part switches from the foam tip to the plastic mount piece (e.g. when foam is fully compressed or when the end-effector is tilting). Considering that increasing K makes the OCP poorly conditioned and prone to numerical instability[6], and that we empirically found that keeping a low K didn't affect the force tracking performance, we chose to set $K = 1000$. Note that the intuition that under-estimating K is safer than over-estimating is also followed in other works [52].

[6]It requires small integration steps, or a good but computationally-expensive integration scheme.

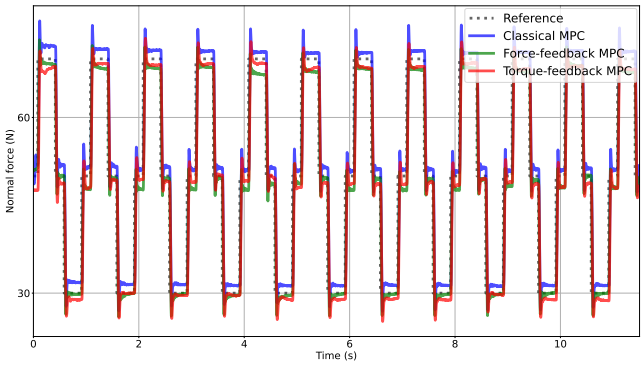


Fig. 4: Experiment 1 : Normal force square signal tracking performance for the 3 controllers.

The contact damping B should be kept high enough to prevent oscillatory predictions (e.g. higher than critical damping $2\sqrt{K}$ is a good heuristic). But we also noticed that too high B can make the force time response "overshoot" and destabilize the system. This can be understood by analogy with a first-order linear system in the singular case of a pure damper ($K = 0$). In that case, the force dynamics (18) can be written under the form

$$\dot{\lambda} + B\alpha(q)\lambda + \beta(q, \dot{q}, \tau) = 0 \tag{36}$$

which has a solution of the form $\lambda(t) = e^{-B\alpha t} + \lambda_0(\beta)$ (for fixed $q, \dot{q}, \tau$), so B acts as a time constant. We empirically determined $B = 100$ to a suitable value throughout all our experiments.

4) MPC parameter selection: There is a trade-off between model accuracy (high K) and OCP discretization step (therefore horizon length). We found also that integrating the OCP with smaller steps enables to select more finely the force trajectories which is important for stability and tracking performance. This also enables to penalize the time-derivative of the force, $\dot{\lambda}$ in the cost and to track aggressive force reference without having to decrease substantially the force tracking cost weight. Therefore we select $\Delta t = 3\,\mathrm{ms}$ and $N = 5$ nodes ($T = 15\,\mathrm{ms}$), for all 3 controllers. The importance of the horizon length and computation complexity will be further discussed in Section VIII. We allow a maximum of 8 iterations of SQP in order to achieve a re-planning frequency of 500 Hz. Note that we did not implement a real-time iteration scheme [80] as we observed in practice that letting the solver converge to the desired KKT residual tolerance (10^{-4}) [71], [72] resulted in a better performance than re-planning faster without waiting for the full convergence.

B. Experiment 1 : Force signal tracking

In this experiment, the robot must track a time-varying force reference signal as shown in Figure 4. The MAE in the normal force are 2.76 N, 2.70 N and 2.16 N for the classical, torque-feedback and force-feedback MPC respectively. The overshoot and oscillations observed in the case of the force-feedback MPC can be avoided by reducing the force tracking cost weight.

C. Experiment 2 : Polishing task with perfect model

In this experiment, the robot must exert 50 N along the vertical to the table while tracking a 14 cm-diameter circle with its end-effector in the table plane at an angular velocity of $3\,\mathrm{rad\,s^{-1}}$, as in Section VI. Figure 5 shows snapshots of the task. Figure 6 shows the normal force profiles and position errors for the 3 controllers. We use the MAE over one circle as a metric and report its mean and standard deviation over 10 circles in Table V. We can see that the torque-feedback and force-feedback MPC achieve a similar performance in this case, both clearly outperforming the classical MPC in position and force tracking. We were able to increase the position gain of the force-feedback MPC without degrading the force tracking capability. As a result, in this experiment, the force-feedback MPC performed better than the torque-feedback MPC.

	Classical	Torque-feedback	Force-feedback
Force (N)	9.31 ± 0.79	4.28 ± 0.63	$\mathbf{1.70 \pm 0.15}$
Position (mm)	6.48 ± 0.21	3.23 ± 0.12	$\mathbf{3.10 \pm 0.11}$

TABLE V: MAE of the normal force and end-effector position for the slow polishing task with perfect table model.

D. Experiment 3 : Polishing task with imperfect model

In this experiment, the task is the same as the one in Section VII-C, but the table is not perfectly horizontal anymore. It is tilted by an unknown angle, as done in the simulation study. The goal is to increase difficulty to distinguish the controllers performance. We run 3 batches of trials under increased difficulties (slow motion, then fast, then with pushes).

1) Experiment 3.1 : Slow circle: The unmodeled table tilt degrades the performances of every controllers in both position and force tracking. The torque-feedback MPC became worse than the classical MPC in force tracking as shown in Figure 7a. The force-feedback MPC still outperforms the 2 other controllers, as shown in Table VI.

	Classical	Torque-feedback	Force-feedback
Force (N)	9.49 ± 1.59	9.67 ± 0.61	$\mathbf{2.12 \pm 0.37}$
Position (mm)	7.04 ± 0.43	3.37 ± 0.28	$\mathbf{3.34 \pm 0.11}$

TABLE VI: Experiment 3.1 : MAE of the normal force and end-effector position for the slow polishing task with imperfect table model (unkwnown table tilt).

2) Experiment 3.2 : Fast circle: The task is the same except that we increase the circle velocity to $6\,\mathrm{rad\,s^{-1}}$. Surprisingly, the torque-feedback MPC struggled to perform the task This

	Classical	Torque-feedback	Force-feedback
Force (N)	9.00 ± 1.61	11.40 ± 1.45	$\mathbf{1.84 \pm 0.32}$
Position (mm)	10.85 ± 0.12	5.41 ± 0.12	$\mathbf{5.29 \pm 0.03}$

TABLE VII: Experiment 3.2 : MAE of the normal force and end-effector position for the fast polishing task with imperfect table model (unknown table tilt)

could be due to the lack of accuracy of the joint torque measurements[7]. Instead, we found better results using an

[7]We used the signal `jointTorqueMeasured` from the FRI API

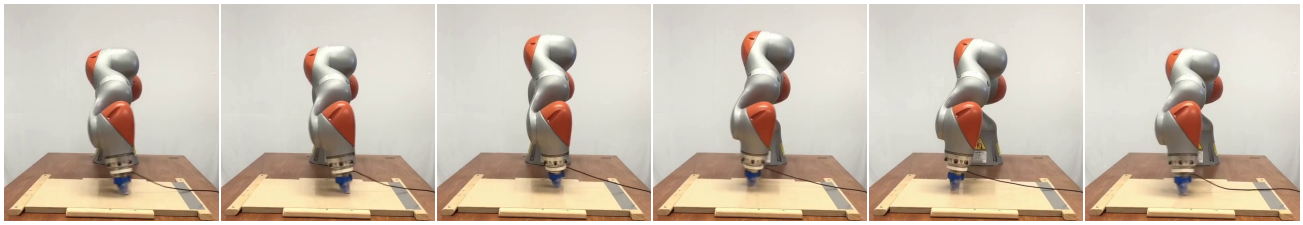
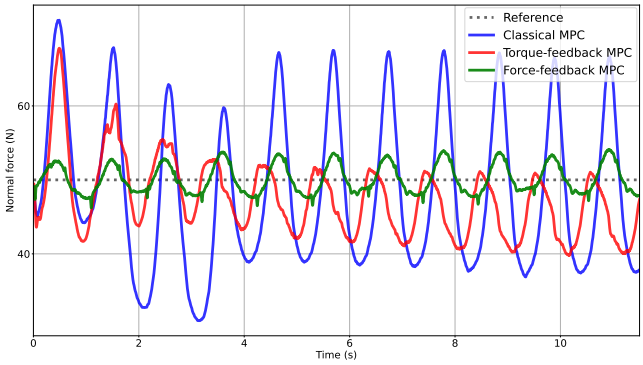
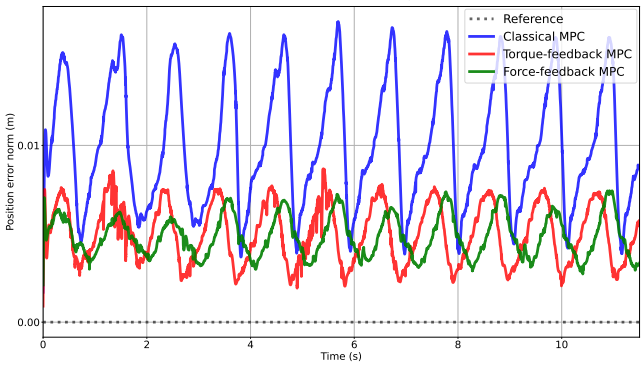


Fig. 5: Snapshots the polishing task.

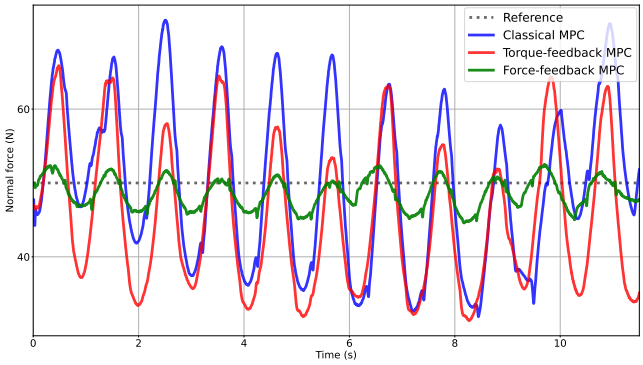


(a) Normal force tracking error.

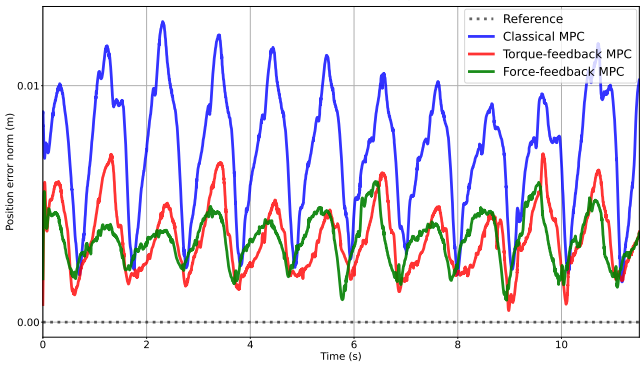


(b) Position tracking error.

Fig. 6: Experiment 2: Tracking performance of the 3 controllers on the slow polishing task (perfect table model)



(a) Normal force tracking error.



(b) Position tracking error.

Fig. 7: Experiment 3.1: Tracking performance of the 3 controllers on the slow polishing task (unknown table tilt).

estimate of the measured torques based on the external torque estimation from the KUKA and our inverse dynamics model

$$\hat{\tau} = \text{RNEA}(q, \dot{q}, \ddot{q}) - \tau_{ext} \tag{37}$$

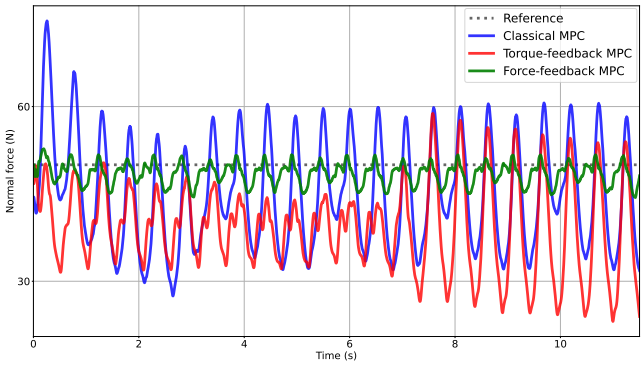
where τ_{ext} is given by `getExternalTorques` in the FRI API, RNEA is computed by `Pinocchio`, $\ddot{q}$ is estimated by finite differences from $\dot{q}$ and filtered with a 2^{nd}-order Butterworth. We verified that τ_{ext} matches our FT sensor measurements $J^T\hat{\lambda}$, which suggests that our sensor model is accurate. The force tracking performance of the torque-feedback MPC is still lower than that of the classical MPC, while its position tracking performance is higher. The force feedback MPC still outperforms both controllers and the difference in performance appears clearly in force plots of Figure 8. The overall performances are reported in Table VII.

3) Experiment 3.3 : Fast circle with external pushes: In this experiment, we run the fast polishing task ($6\,\text{rad}\,\text{s}^{-1}$) on the horizontal table (no tilt) but external pushes are applied by a human operator as shown in the attached video. The position and force performance plots in Figure 9 illustrate qualitatively the superior robustness of the force-feedback MPC in face of unknown disturbances: it is able to maintain both good force and position tracking performances throughout the whole task, while the tracking performances of the classical MPC and torque-feedback MPC are highly affected by the pushes.

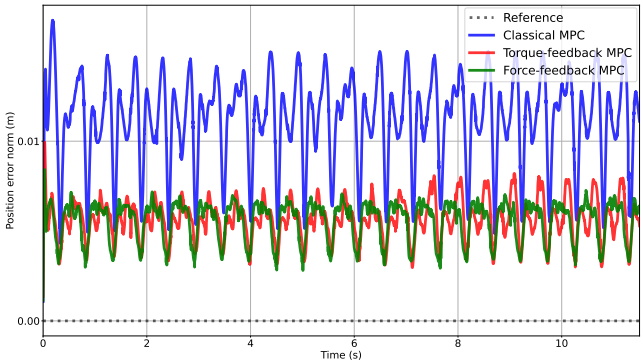
VIII. DISCUSSION

A. Computational considerations

1) Resolution complexity: Augmenting the classical MPC state with additional variables naturally comes with an increased computational cost. Assuming a fully-actuated robot

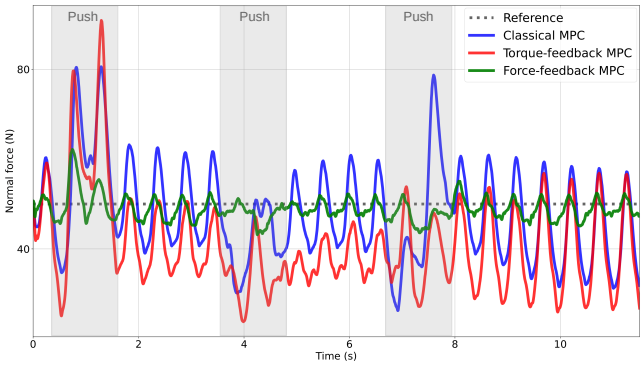


(a) Normal force tracking error.

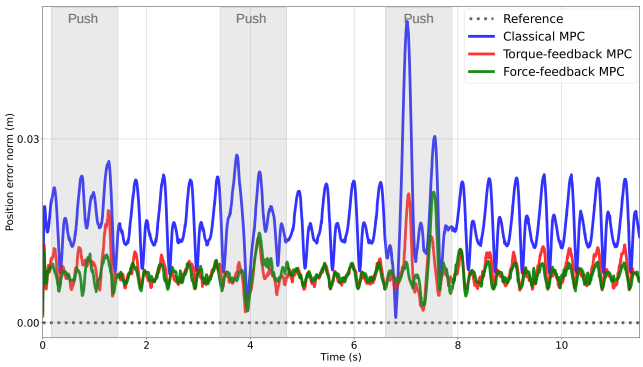


(b) Position tracking error.

Fig. 8: Experiment 3.2: Tracking performance of the 3 controllers on the fast polishing task (unknown table tilt).



(a) Normal force tracking error.



(b) Position tracking error.

Fig. 9: Experiment 3.3: Tracking performance of the 3 controllers on the fast polishing task (perfect table model) with external pushes from a human operator.

with n degrees-of-freedom, the classical MPC state contains $2n$ variables, so the NLP resolution is $O\left((2n)^3\right)$. For the torque-feedback MPC, the state includes n joint torques so the complexity becomes $O\left((3n)^3\right)$, and the force-feedback MPC state includes an m-dimensional contact force so the complexity becomes $O\left((2n+m)^3\right)$. In our polishing experiments, we have $m = 1$ so the increase in complexity is moderate. For higher dimensional contact models (e.g. a full $6D$ wrench, i.e. $m = 6$) the increase of complexity would be much more appreciable. Nevertheless we can see that in the case of a robot with many degrees-of-freedom ($n >> m$) such as a humanoid, the complexity increases much more for the torque-feedback MPC than for the force-feedback MPC. This indicates that the proposed approach is more efficient than the torque-feedback MPC in the case of high-dimensional robots.

2) Importance of the horizon: The NLP resolution complexity scales linearly with the MPC horizon N, which imposes a trade-off between horizon length and re-planning frequency. In our experiments, we found that keeping a small number of nodes ($N = 5$) to allow a fast re-planning frequency ($500\,\text{Hz}$) allowed to increase the force task weight of the force-feedback MPC and was thereby beneficial to the overall control performance. One possible strategy to increase the re-planning frequency *without* shortening the horizon is to reduce the number of nodes in the horizon while increasing the integration step Δt. However given the semi-implicit integration scheme (24) used in our implementation, we found empirically that a good compromise between force tracking performance and stability was to keep a small integration step ($\Delta t = 15\,\text{ms}$) while under-estimating the contact stiffness ($K = 10^3$). This trade-off between integration stability and model accuracy could be alleviated by using more advanced integrators that allow larger integration steps, such as exponential integration [81].

Despite these choices, keeping a horizon is important i.e. 5 nodes ($T = 15\,\text{ms}$) is *always* better than 1 node ($T = 3\,\text{ms}$). Indeed, collapsing the horizon of the classical MPC to 1 node (i.e. $T = 5\,\text{ms}$) in the polishing task leads to a substantial degradation in the position tracking performance (and a negligible improvement in force tracking). This poor trade-off in favor of the force task is not surprising since the end-effector trajectory tracking task requires more planning than the "instantaneous" force task. The same horizon reduction destabilized the force-feedback MPC. Decreasing its force tracking weight led to a similar trade-off at the expense of the motion task. Considering that a singular horizon MPC ($N = 1$) is essentially equivalent to task-space inverse dynamics [75], this indicates that a short-sighted MPC is still better performing than instantaneous control schemes thanks to its *planning* capability. This also suggests that further increasing the horizon would lead to better motion and force performances. Here we have experimented rather simple tasks where the planning capability was not fundamental (e.g. no

collision avoidance), so this observation is likely to become more true for more complex tasks.

B. Modeling assumptions

1) Contact model: The stiffness and damping K, B parameters in (16) were determined empirically under stability and performance considerations, however they could be estimated as proposed in [52] or in an indirect adaptive control fashion [50]. Besides, the location of the anchor point is assumed to be perfectly known. Although it does not affect the visco-elastic dynamics (only its velocity $\dot{p}$ appears in the soft contact force evolution equation (18)), measuring or estimating the anchor point location would be useful to improve the management of contact transitions. Note finally that the pure sliding contact model ($m = 1$) used in the polishing task could be improved by taking into the account the lateral forces using e.g. feedback linearization or a Coulomb friction model [82], or by adding directly a visco-elastic model in the MPC.

2) Contact switch: Furthermore, the proposed optimal control formulation assumes that the contact switching time is perfectly known (i.e. the time-varying cost function and robot dynamics are switched online at a pre-defined time instant). This implies in practice tedious fine-tuning to stabilize the contact transition. We could instead consider optimizing the contact switching times or make the optimization time-invariant.

C. Hardware limitations

1) Torque ripples due to gearing: Nonlinearities in the transmission generates ripples in torques measurements, with a frequency proportional to velocity. This phenomena is described and thoroughly studied in [83]. These vibrations impacted negatively the performance force-feedback MPC: they reflect in the FT sensor measurements under the form of $30 - 40Hz$ oscillations with an amplitude that varies with the force and velocity of the polishing. These oscillations can be partly filtered out with a 2^{nd}-order Butterworth filter, but it still limits the magnitude of the force weight. This phenomena is therefore a direct limitation on the performance of the proposed approach, and is expected to be improved with better hardware.

2) Mimatch between joint torque sensors and FT sensor: As previously mentioned, the joint torque measurements read from the FRI API do not align with the force sensor measurements (37). This could explain why the torque-feedback MPC did not work at first when using the measurements provided by the KUKA torque sensors. This would also corroborate the observation made in simulation that poor torque tracking impacts torque-feedback MPC the most, although further analysis would be necessary to confirm this intuition. We used the estimate (37) to carry out the experiments, which enabled to outperform performance of the classical in position tracking. But despite this improvement, the force tracking is still worse for the torque-feedback MPC than the classical MPC.

No proper explanation has been established by our experimental setup, whether it could come from neglecting the acceleration or from a significantly different robot (inertial) model internally used by KUKA algorithms.

IX. CONCLUSION

In this article, we proposed a novel MPC formulation that allows to reactively plan motions *and* contact forces. This approach relies on a reformulation of the optimal control problem: a state augmentation with a linear visco-elastic contact model allows to break the algebraic coupling that exist in the rigid contact model between joint torques and contact forces. The proposed approach was shown to outperform the classical MPC formulation, and the previously proposed torque-feedback MPC formulation that models torque actuation as a linear low-pass filter. In particular, we showed through a simulation study that our MPC scheme was less sensitive to the quality of the low-level torque control, and through hardware experiments that it enables a high motion *and* force tracking accuracy in challenging, dynamic contact tasks. In conclusion our experiments demonstrate that using sensors that are co-located with the task is beneficial to performance, and that force control and MPC can complement each other.

Our approach provides a first complete answer to the problem of achieving good control performance in contact tasks with MPC, and addresses the challenge faced by force control to achieve high-bandwidth force control without having to explicitly plan impedance profiles. Future work includes improving the contact modeling, adding constraints, and apply the present methodology to floating-based robots and locomotion problems.

REFERENCES

[1] R. Featherstone, *Rigid Body Dynamics Algorithms*, 2008.

[2] F. Farshidian, E. Jelavic *et al.*, "Real-time motion planning of legged robots: A model predictive control approach," in *IEEE-RAS International Conference on Humanoid Robotics (Humanoids)*, 2017.

[3] E. Dantec, M. Taïx, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE RAL*, vol. 7, 2022.

[4] S. Katayama and T. Ohtsuka, "Whole-body model predictive control with rigid contacts via online switching time optimization," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2022-October, pp. 8858–8865, 2022.

[5] R. Grandia, F. Jenelten *et al.*, "Perceptive Locomotion Through Nonlinear Model-Predictive Control," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023.

[6] C. Mastalli, S. P. Chhatoi *et al.*, "Inverse-Dynamics MPC via Nullspace Resolution," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3222–3241, 2023.

[7] A. Meduri, P. Shah *et al.*, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Transactions on Robotics*, vol. 39, no. 2, p. 905–922, Apr. 2023. [Online]. Available: https://arxiv.org/abs/2201.07601

[8] M. Neunert, M. Stauble *et al.*, "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[9] S. Fahmi, M. Focchi *et al.*, "STANCE: Locomotion Adaptation over Soft Terrain," *IEEE TRO*, vol. 36, no. 2, apr 2020.

[10] I. Kao, K. Lynch, and J. W. Burdick, *Contact Modeling and Manipulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 647–669. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_28

[11] E. Corral, R. G. Moreno *et al.*, "Nonlinear phenomena of contact in multibody systems dynamics: a review," *Nonlinear Dynamics*, mar 2021. [Online]. Available: http://link.springer.com/10.1007/s11071-021-06344-z

[12] D. E. Stewart, *Rigid-body dynamics with friction and impact*, 2000, vol. 42, no. 1.

[13] F. Pfeiffer and C. Glocker, *Multibody dynamics with unilateral contacts*, ser. Wiley Series in Nonlinear Science. Hoboken, NJ: Wiley, 2008. [Online]. Available: https://cds.cern.ch/record/1609914

[14] Q. L. Lidec, W. Jallet *et al.*, "Contact Models in Robotics: a Comparative Analysis," 2023. [Online]. Available: http://arxiv.org/abs/2304.06372

[15] M. Dahleh and G. Verghese, "Lectures on Dynamic Systems and Control Interconnected Systems and Feedback : Well-Posedness , Stability , and Performance," *Electrical Engineering*.

[16] L. R. Fletcher, "Output feedback matrices in the presence of direct feedthrough," *International Journal of Systems Science*, vol. 12, no. 12, pp. 1493–1495, 1981.

[17] S. Kang, J. Cha, and S. Ko, "Linear quadratic regulation and tracking using output feedback with direct feedthrough," *International Journal of Aeronautical and Space Sciences*, vol. 17, no. 4, pp. 593–603, 2016.

[18] M. Menner, A. M. Annaswamy, and A. W. Zollitsch, "Adaptive output feedback for plants with direct feedthrough," *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, no. Cdc, pp. 2121–2127, 2016.

[19] A. A. Adegbege and W. P. Heath, "Multivariable algebraic loops with complementarity constraints enforcing some KKT conditions," *2014 52nd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2014*, vol. 2, pp. 1033–1039, 2014.

[20] A. Bemporad, N. Lawrence Ricker, and J. G. Owen, "Model predictive control - New tools for design and evaluation," in *Proceedings of the American Control Conference*, vol. 6, 2004, pp. 5622–5627.

[21] J. M. Maciejowski, P. J. Goulart, and E. C. Kerrigan, *Constrained control using model predictive control*, 2007, vol. 346.

[22] A. W. Ordys and A. W. Pike, "State space generalized predictive control incorporating direct through terms," *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, no. December, pp. 4740–4741, 1998.

[23] M. A. Xavier, A. K. de Souza, and M. S. Trimboli, "A split-future MPC algorithm for lithium-ion battery cell-level fast-charge control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 12 459–12 464, 2020. [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.1330

[24] C. L. Cham, A. H. Tan *et al.*, "Model predictive control with direct feedthrough with application on a MIST reactor," *IFAC-PapersOnLine*, vol. 53, no. 1, pp. 183–188, 2020. [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.06.031

[25] A. Aydinoglu, P. Sieg *et al.*, "Stabilization of complementarity systems via contact-aware controllers," *IEEE TRO*, vol. 38, 2021.

[26] S. Kleff, E. Dantec *et al.*, "Introducing force feedback in model predictive control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 379–13 385.

[27] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," *Autonomous Robots*, vol. 40, no. 3, pp. 537–560, 2016.

[28] J. Matschek, J. Bethge *et al.*, "Force Feedback and Path Following using Predictive Control: Concept and Application to a Lightweight Robot," *IFAC-PapersOnLine*, vol. 50, 2017.

[29] T. Gold, A. Völz, and K. Graichen, "Model Predictive Interaction Control for Robotic Manipulation Tasks," vol. 39, no. 1, pp. 386–391, 2023.

[30] L. Wijayarathne, Z. Zhou *et al.*, "Real-Time Deformable-Contact-Aware Model Predictive Control for Force-Modulated Manipulation," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3549–3566, 2023.

[31] A. Wahrburg and K. Listmann, "MPC-based admittance control for robotic manipulators," *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, pp. 7548–7554, dec 2016.

[32] K. J. Kazim, J. Bethge *et al.*, "Combined Predictive Path Following and Admittance Control," *Proceedings of the American Control Conference*, vol. 2018-June, pp. 3153–3158, aug 2018.

[33] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, oct 2020.

[34] M. V. Minniti, R. Grandia *et al.*, "Model Predictive Robot-Environment Interaction Control for Mobile Manipulation Tasks," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, no. Icra, pp. 1651–1657, 2021.

[35] D. E. Whitney, "Force feedback control of manipulator fine motions," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 99, no. 2, pp. 91–97, 1977.

[36] R. P. Paul, "PROBLEMS AND RESEARCH ISSUES ASSOCIATED WITH THE HYBRID CONTROL OF FORCE AND DISPLACEMENT." pp. 1966–1971, 1987.

[37] L. Villani and J. De Schutter, *Force Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 161–185. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_8

[38] D. E. Whitney, "Historical Perspective and State of the Art in Robot Force Control," *The International Journal of Robotics Research*, vol. 6, no. 1, pp. 3–14, 1987. [Online]. Available: https://doi.org/10.1177/027836498700600101

[39] R. Volpe and P. Khosla, "The equivalence of second order impedance control and proportional gain explicit force control: Theory and experiments," *Lecture Notes in Control and Information Sciences*, vol. 190, pp. 1–24, 1993.

[40] M. T. Mason, "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[41] M. Raibert and J. J. Craig, "Hybrid Position / Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. June 1981, pp. 126–133, 1981.

[42] S. Chiaverini and L. Sciavicco, "The Parallel Approach to Force/Position Control of Robotic Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.

[43] B. Siciliano, "Parallel Force / Position Control of Robot Manipulators State-of-Art Modeling," *Robotics Research*, no. March, pp. 78–89, 1996.

[44] T. Yoshikawa, "Force control of robot manipulators," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, no. April, pp. 220–226, 2000.

[45] N. Hogan, "Contact and Physical Interaction," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 1–25, 2022.

[46] J. Duffy, "The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces," pp. 139–144, 1990.

[47] N. Hogan, "Impedance Control Part1-3," *J. Dyn. Sys., Meas., Control.*, vol. 107, pp. 1–24, 1985.

[48] Neville Hogan, "Stable Execution of Contact Tasks using Impedance Control," *Tetrahedron Letters*, vol. 28, no. 44, pp. 5241–5244, 1987.

[49] W. S. Newman, "Stability and performance limits of interaction controllers," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 114, no. 4, pp. 563–570, 1992.

[50] H. Seraji and R. Colbaugh, "Force tracking in impedance control," *International Journal of Robotics Research*, vol. 16, no. 1, pp. 97–117, 1997.

[51] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control for robot manipulators with an unknown environment: Theory, simulation, and experiment," *International Journal of Robotics Research*, vol. 20, no. 9, pp. 765–774, 2001.

[52] D. Erickson, M. Weber, and I. Sharf, "Contact Stiffness and Damping Estimation for Robotic Systems," *International Journal of Robotics Research*, vol. 22, no. 1, pp. 41–57, 2003.

[53] J. E. Colgate, "The Control of Dynamically Interacting Systems," Ph.D. dissertation, Massachusetts Institute of Technology (MIT), 1988.

[54] S. Eppinger and W. Seering, "Understanding bandwidth limitations in robot force control," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 904–909.

[55] E. D. Fasse and N. Hogan, "Control of Physical Contact and Dynamic Interaction," in *Robotics Research*, 1996, pp. 28–38.

[56] M. Dohring and W. Newman, "The passivity of natural admittance control implementations," *IEEE ICRA*, 2003.

[57] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *IJRR*, vol. 26, 2007.

[58] M. Focchi, G. A. Medrano-Cerda *et al.*, "Robot impedance control and passivity analysis with inner torque and velocity feedback loops," *Control Theory and Technology*, vol. 14, 2014.

[59] A. Ajoudani, A. M. Zanchettin *et al.*, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018. [Online]. Available: https://doi.org/10.1007/s10514-017-9677-2

[60] Ribin Balachandran, Mikael Jorda *et al.*, "Passivity-based Stability in Explicit Force Control of Robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7989050

[61] S. Hogan, Neville and Cotter, "Impedance Control: An Approach To Manipulation: Part III - Applications," *Robotics research and advanced applications*, vol. 107, no. 1982, pp. 121–128, 1985.

[62] R. Colbaugh, H. Seraji, and K. Glass, "Adaptive impedance control of redundant manipulators," in *29th IEEE Conference on Decision and Control*, 1990, pp. 2661–2666 vol.5.

[63] H. Seraji, "ADAPTIVE ADMITTANCE CONTROL: An Approach to Explicit Force Control," *Jet Propulsion*, pp. 2705–2712, 1994.

[64] S. Singh and D. Popa, "An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 912–921, 1995.

[65] L. J. Love and W. J. Book, "Environment estimation for enhanced impedance control," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1854–1859, 1995.
[66] M. Matinfar and K. Hashtrudi-Zaad, "Optimization-based robot impedance controller design," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, pp. 1246–1251, 2004.
[67] J. Buchli, F. Stulp *et al.*, "Learning variable impedance control," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.
[68] M. Bogdanovic, M. Khadiv, and L. Righetti, "Learning variable impedance control for contact sensitive tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6129–6136, 2020.
[69] B. Hammoud, M. Khadiv, and L. Righetti, "Impedance optimization for uncertain contact interactions through risk sensitive optimal control," *IEEE RAL*, vol. 6, 2021.
[70] N. Hogan, "Impedance Control: An Approach to Manipulation: Part II—Implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 03 1985. [Online]. Available: https://doi.org/10.1115/1.3140713
[71] J. Nocedal and S. J. Wright, *Numerical optimization*, 2006.
[72] A. Jordana, S. Kleff *et al.*, "Stagewise Implementations of Sequential Quadratic Programming for Model-Predictive Control," Tech. Rep. [Online]. Available: https://laas.hal.science/hal-04330251
[73] F. E. Udwadia and R. E. Kalaba, "A new perspective on constrained motion," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, pp. 407–410, 1992.
[74] R. Budhiraja, J. Carpentier *et al.*, "Differential dynamic programming for multi-phase rigid contact dynamics," in *IEEE Humanoids*, 2018.
[75] S. Kleff, A. Meduri *et al.*, "High-frequency nonlinear model predictive control of a manipulator," in *IEEE ICRA*, 2021.
[76] R. Grandia, F. Farshidian *et al.*, "Frequency-aware model predictive control," *IEEE RAL*, vol. 4, 2019.
[77] J. Carpentier, G. Saurel *et al.*, "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE/SICE SII*, 2019.
[78] S. Kleff, J. Carpentier *et al.*, "On the Derivation of the Contact Dynamics in Arbitrary Frames (Application to Polishing with Talos)," aug 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03758989https://hal.archives-ouvertes.fr/hal-03758989/document
[79] C. Mastalli, R. Budhiraja *et al.*, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE ICRA*, 2020.
[80] M. Diehl, H. Bock *et al.*, *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 65–93. [Online]. Available: https://doi.org/10.1007/978-3-540-36119-0_4
[81] B. Hammoud, L. Olivieri *et al.*, "Exponential integration for efficient and accurate multibody simulation with stiff viscoelastic contacts," *Multibody System Dynamics*, vol. 54, no. 4, pp. 443–460, apr 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11044-022-09818-z
[82] A. Jordana, S. Kleff *et al.*, "Force feedback model-predictive control via online estimation," in *IEEE ICRA*, 2024 (In press).
[83] V. Chawda and G. Niemeyer, "Toward torque control of a KUKA LBR IIWA for physical human-robot interaction," *IEEE IROS*, 2017.

Force Feedback Model-Predictive Control via Online Estimation

Armand Jordana*,[1], Sébastien Kleff*,[1], Justin Carpentier[2], Nicolas Mansard[3,4], Ludovic Righetti[1]

***Abstract*— Nonlinear model-predictive control has recently shown its practicability in robotics. However it remains limited in contact interaction tasks due to its inability to leverage sensed efforts. In this work, we propose a novel model-predictive control approach that incorporates direct feedback from force sensors while circumventing explicit modeling of the contact force evolution. Our approach is based on the online estimation of the discrepancy between the force predicted by the dynamics model and force measurements, combined with high-frequency nonlinear model-predictive control. We report an experimental validation on a torque-controlled manipulator in challenging tasks for which accurate force tracking is necessary. We show that a simple reformulation of the optimal control problem combined with standard estimation tools enables to achieve state-of-the-art performance in force control while preserving the benefits of model-predictive control, thereby outperforming traditional force control techniques. This work paves the way toward a more systematic integration of force sensors in model predictive control.**

I. Introduction

A. Motivation

Many tasks require accurate control of contact forces exerted on the environment: polishing, grinding, grasping, etc. This skill, trivial to humans, remains beyond most robot's abilities despite continuous progress in robotics research over the past decades. While Model Predictive Control (MPC) affords the online synthesis of complex motions, it remains fundamentally limited in its ability to control physical interaction. As a matter of fact, although force sensors have been used since the early days of robotics [1], they remain notably absent from modern control techniques relying on model-based optimization.

This is partly because predicting the evolution of contact forces is challenging in general and involves sophisticated models [2] that are too specific or impractical for real-time applications. Hence, the contact models used in practice for optimization-based control are kept simple for algorithmic convenience [3]. However, these simplifications hinder the ability to derive meaningful control policies in contact with explicit force feedback. To this day, the predictive feedback control of contact forces remains an open problem.

In this work, we address this issue and show that standard estimation tools [4] together with a reformulation of the optimal control problem can provide a simple yet effective framework to achieve force-output-feedback MPC.

* Equal contribution - first authors listed in alphabetical order.

[1] Machines in Motion Laboratory, New York University, USA firstname.lastname@nyu.edu

[2] Inria, Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France.

[3]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse

[4]Artificial and Natural Intelligence Toulouse Institute (ANITI), Toulouse

B. Related work

Force control techniques are classically divided into direct force control and indirect force control [5]. A full introduction is out of the scope, so we only provide here a brief overview and refer the reader to the concise introductory review on active compliant control proposed in [6].

Direct methods attempt to regulate the force explicitly using measurement feedback, typically in an integral controller - which is historically considered the best basic strategy for force tracking [7]. It can be combined with motion feedback in complementary task directions [8], or in parallel [9]. While the use of explicit force feedback enables high accuracy tracking, the artificial decoupling of force and motion tasks hides potential conflicts [10], [11] or phenomena such as contact friction [12] and exchange of mechanical work [13].

On the other hand, indirect methods, such as impedance control [14] or admittance control [15], [16], aim at regulating the dynamic relationship between force and motion. While this allows to generate stable and compliant contact interactions, such techniques are mainly limited by their force tracking capability: since the force is controlled indirectly through motion regulation, the tracking performance depends on a priori unknown environment parameters [17]–[20].

More recently, MPC has shown its ability to accommodate conflicting objectives through constrained nonlinear optimization [21]. Much research has focused on introducing MPC into direct [22], [23] and indirect [24]–[28] force control methods, mainly motivated by its ability to satisfy constraints. In contrast to [22], [23], [26], [28], [29], the proposed approach does not require a contact force *dynamics* model, which greatly simplifies the optimization. Unlike [24], [25], [27], we use a force sensor to achieve explicit force tracking rather than impedance/admittance regulation.

Estimation can also be used to improve performance in force tasks. In [30], external forces are estimated with a centroidal model. In [31], a state-dependent force correction model is adapted online. Closer to our work, [32] proposed an active Kalman observer in MPC to reject unmodeled disturbances at the input level, which can be viewed as a form of model-reference (direct) adaptive control. However, those lines of work do not consider the full dynamics model.

C. Contributions

In this paper, we propose a novel MPC formulation that allows to exploit direct feedback from force sensors. We show that simple contact models and standard estimation tools allow to incorporate force feedback in MPC and to achieve state-of-the-art performance. We claim that force feedback in MPC is not as challenging as it seems and that it solves

many issues: it circumvents tedious modeling of complex phenomena (contact, friction, etc.), boosts performance of classical MPC in contact tasks, and does not conflict with optimization contrary to traditional force control methods.

We propose to use force measurements to estimate online the mismatch between the robot's dynamics model and measurements. This mismatch is used to correct directly the predictive model or the control objective. This idea resembles that of indirect adaptive control [33], where a model of the plant is identified online to adapt the controller's parameters. Our approach allows high-quality force tracking accuracy in challenging interaction tasks. Our main contributions are:

- a new framework affording direct force feedback control inside nonlinear MPC based on online estimation and feedback linearization
- a systematic comparative experimental study of our force feedback MPC against traditional techniques.

In particular, we demonstrate that the proposed approach outperforms integral control: it benefits from the same force tracking capability *without* impeding the benefits of MPC. In particular, in contrast to integral control, our approach maintains or improves the MPC running cost performance. It has also the advantage of being conceptually simple and cheap to implement with existing tools and software.

II. Background

In this section, we recall the classical MPC formulation for torque-controlled robots under rigid contacts, and point out its inherent inability to provide force-feedback policies.

A. Classical model-predictive control

MPC solves online the Optimal Control Problem (OCP)

$$\min_{x(.),u(.)} \int_0^T \ell\left(x(t),u(t),t\right)dt + \ell_T\left(x(T)\right) \qquad (1)$$
$$\text{s.t. } \dot{x}(t) = f(x(t),u(t))$$

where $x(0) = x^m$ is the initial (measured) state, f the dynamics model, and ℓ, ℓ_T the running and terminal costs. Note that hard constraints on the state and control can be added, as soft penalties or hard constraints - which may be more challenging for real-time applications. This OCP is transcripted into a non-linear program, i.e. the cost and dynamics are discretized using an Euler discretization scheme. This program is solved online at each control cycle. For the remainder, and without limitation, we assume that the robot is fully actuated with n joints, the state vector $x = (q, \dot{q}) \in \mathbb{R}^{2n}$ includes the joint positions and velocities and the control vector $u = \tau \in \mathbb{R}^n$ includes the joint torques.

B. Rigid contact model

In optimization-based control, it is convenient to assume that contacts between the robot and the environment are *rigid*, i.e., pure kinematic constraints that can be resolved at the dynamics level. The dynamics of a robot in contact is given by the following constrained dynamical system corresponding to the KKT conditions of Gauss' principle of least constraint [34]

$$\begin{bmatrix} M(q) & J^T(q) \\ J(q) & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ -F \end{bmatrix} = \begin{bmatrix} \tau - b(q,\dot{q}) \\ -\alpha_0(q,\dot{q}) \end{bmatrix} \qquad (2)$$

where $M(q) \in \mathbb{R}^{n\times n}$ is the generalized inertia matrix, $J(q) \in \mathbb{R}^{n_c \times n}$ the contact Jacobian, $F \in \mathbb{R}^{n_c}$ the contact force, $b(q,\dot{q}) \in \mathbb{R}^n$ the nonlinear effects of Coriolis, centrifugal and gravity forces, and $\alpha_0(q,\dot{q}) \in \mathbb{R}^{n_c}$ the contact acceleration drift. For clarity, the dynamics f in (1), is in fact the solution map of system (2), i.e. $f : (q,\dot{q},\tau) \mapsto (\ddot{q}, F)$. The dependencies in $q, \dot{q}$ will be dropped in the remainder.

C. The challenge of force feedback

While the rigid contact model conveniently fits the MPC framework, it inherently prevents force feedback. The contact force F corresponds to the Lagrange multiplier of the contact constraint, namely $J\ddot{q} + \alpha_0 = 0$ (second row of the system (2)) [35]. As such, it cannot be controlled in a feedback sense: once $x = (q,\dot{q})$ and F are measured, $u = \tau$ is already completely determined by (2). Hence, u cannot be optimized as a function of F without creating an algebraic loop. This issue is a typical pathology from control systems with non-zero input-output feedthrough and can be broken by introducing delay [36]. This point was discussed and addressed in our previous work [29], where actuation was modeled as a low-pass filter, and the joint torques were treated as part of an augmented state. In contrast, we propose in this paper to break this coupling thanks to the online estimation without augmenting the state of the MPC.

III. Method

This section presents a new approach using estimation to leverage force sensor feedback in MPC. It includes an estimator, a reformulation of the MPC problem to include force feedback in the MPC model, and a feedback-linearizing compensation term for unmodeled force directions.

A. Estimation

As explained previously, it is unclear how to achieve force feedback under the rigid contact assumption without introducing delays or more complex contact models. We show here that estimation is a simple way to circumvent this issue by keeping the rigid contact assumption and correcting the model. Indeed, due to numerous model inaccuracies, the force F predicted by (2) rarely matches the force measurement. Hence a natural idea is to keep track of this mismatch by estimating online the offset between the model and the measurement with standard Kalman filtering [4].

The idea of estimating an offset error to improve the closed-loop performance of the controller is standard in estimation (e.g., [30]). We show that a disturbance Δ in the dynamics can incorporate rich force sensor feedback information in the MPC. We consider a model of the form:

$$M\ddot{q} + b = \tau + J^T F + \mathcal{M}(\Delta), \qquad (3a)$$
$$J\ddot{q} = -\alpha_0. \qquad (3b)$$

Here, $\mathcal{M}$ models how Δ offsets the dynamics. While the mismatch can be modeled in many ways, we assume that $\mathcal{M}$ is linear. Specifically, we consider two different models:

- Torque offset (in joint space) : $\mathcal{M}(\Delta\tau) = \Delta\tau$
- Force offset (in task space) : $\mathcal{M}(\Delta F) = J^T \Delta F$

This offset is meant to correct the model mismatch due to inaccurate modeling of, e.g., the dynamics, contact model, external disturbance, etc. The idea is to estimate the offset online, given raw measurement. More precisely, given a prior on the offset $\widehat{\Delta}$, we use joint positions, velocities, accelerations, torque commands, and force measurements to update the force offset. We assume perfect joint position and velocity measurements, and Gaussian measurement noise:

$$\Delta = \widehat{\Delta} + w, \qquad w \sim \mathcal{N}(0, P), \tag{4a}$$
$$\ddot{q}^m = \ddot{q} + v, \qquad v \sim \mathcal{N}(0, Q), \tag{4b}$$
$$F^m = F + \eta, \qquad \eta \sim \mathcal{N}(0, R), \tag{4c}$$

where F^m is the force measurement and $\ddot{q}^m$ the acceleration measurement. P, Q and R are positive-definite covariance matrices. As it is traditionally done in Kalman filtering, each disturbance distribution is considered to be Gaussian, which allows to solve the Maximum Likelihood Estimation (MLE) problem [4]. Here, the MLE aims at finding the parameters $\Delta, \ddot{q}, F$ that maximize the probability density function given the observed measurement and prior force offset:

$$\max_{\Delta, \ddot{q}, F} \quad p(\Delta, \ddot{q}, F \mid \widehat{\Delta}, \ddot{q}^m, F^m) \tag{5}$$
$$\text{subject to constraint (3a)}$$

Applying the negative logarithm and leveraging the normal distribution assumption, the problem is equivalent to:

$$\min_{\Delta, \ddot{q}, F} \|\Delta - \widehat{\Delta}\|^2_{P^{-1}} + \|\ddot{q} - \ddot{q}^m\|^2_{Q^{-1}} + \|F - F^m\|^2_{R^{-1}}$$
$$\text{subject to constraint (3a)} \tag{6}$$

where $\|w\|^2_{P^{-1}} = w^T P^{-1} w$. If $\mathcal{M}(\Delta)$ is linear, Problem (6) becomes an equality QP and can be solved very efficiently with off-the-shelf solvers. This, in turn, allows high-frequency online estimation, e.g., 5 kHz for a 7 DoF robot. As in a Kalman filter, the obtained estimate Δ is used as a prior at the next time step.

Note that other constraints can be considered in the QP, such as inequalities on estimated quantities (e.g. force offset).

Remark 1. *If additional inequality constraints are unnecessary, one may solve the problem using a Kalman filter [4]. More specifically, one can use Recursive Least Squares (RLS) [37] with the transition equation, $\Delta = \widehat{\Delta} + w$ along with the observation equation*

$$\begin{bmatrix} \ddot{q}^m \\ F^m \end{bmatrix} = \begin{bmatrix} -M & J^T \\ J & 0 \end{bmatrix}^{-1} \begin{bmatrix} b - \tau - \mathcal{M}(\Delta) \\ -\alpha_0 \end{bmatrix} + \begin{bmatrix} v \\ \eta \end{bmatrix}, \tag{7}$$

in order to estimate Δ online. Note that if $\mathcal{M}$ is linear, this observation model is linear, and one can use the RLS equations to derive an update rule on Δ.

B. Force feedback in the MPC via estimation

Once estimated, the force offset must be considered by the controller. This will break the coupling between forces and torques discussed in Section II-C by adding a delay between the measurement and the corrective term ΔF.

1) Naive inclusion as a corrective control: A naive approach is to add a feedforward term to the optimal torque given by the MPC, τ_{MPC}, to compensate the estimated offset:

$$\tau = \tau_{\text{MPC}} - \mathcal{M}(\Delta). \tag{8}$$

Although this work focuses on MPC, this method is agnostic to the nature of the controller.

2) Inclusion in the predictive model: Alternatively, the offset can be considered directly in the model used by the MPC. More precisely, we can consider that the offset will be constant over the horizon of the MPC and solve the OCP using as dynamics Eq. (3a) (instead of Eq. (2)). The MPC model is then updated online at each offset estimate update.

Remark 2. *Interestingly, when $\mathcal{M}(\Delta F) = J^T \Delta F$, updating the predictive model is in fact equivalent to modifying the force reference in the cost function. More specifically, the modified dynamics can be written in the following way:*

$$\begin{bmatrix} \ddot{q} \\ F \end{bmatrix} = \begin{bmatrix} -M & J^T \\ J & 0 \end{bmatrix}^{-1} \begin{bmatrix} b - \tau \\ -\alpha_0 \end{bmatrix} - \begin{bmatrix} 0 \\ \Delta F \end{bmatrix}. \tag{9}$$

Therefore, the force offset only biases the predicted forces and does not affect the acceleration. This means that this force offset has no impact on the predicted trajectory. The offset will only impact terms of the cost function that include the predicted force. Given a cost of the form $\ell(x, u, F(x, u, \Delta F))$, we can simply consider $\ell(x, u, F(x, u) - \Delta F))$, and discard ΔF from the prediction model. This greatly simplifies the implementation and gives more interpretation to the method. Interestingly, if the cost function does not depend on the force, the force offset will not impact the solution of the OCP.

C. Direct compensation of unmodeled force directions

The above formulation assumes that force can only be exerted in the n_c constrained dimensions. However, in reality, forces can exist in the other $6 - n_c$ directions and may interfere with the task if not taken into account (e.g. friction during a polishing task if only the normal force is modeled).

Following [1], instead of using an explicit 6D force model to compute a feed-forward compensation term, we propose to use the force measurements directly. This is in fact a form of Feedback Linearization (FL) as emphasized in [38]. Concretely, we add to the optimal torque given by the MPC the following compensation FL term

$$\tau = \tau_{\text{MPC}} - J^T_{\text{6D}} S F^m_{\text{6D}}, \tag{10}$$

where $J_{\text{6D}} \in \mathbb{R}^{n \times 6}$ and $F^m_{\text{6D}} \in \mathbb{R}^6$ are the full 6D Jacobian and measured force, and the selection matrix $S : \mathbb{R}^6 \to \mathbb{R}^6$ nullifies the n_c constrained dimensions. In the experiment section, we will show that this simple FL term will lead to

competitive performances with more established yet more complex friction models such as the Coulomb model.

From a control perspective, it could seem unsafe at first glance to use measured forces in the control torque because the robot would always maintain itself in a disturbed state, which would create divergence of the force (e.g., pushing harder). But this would happen only if unmodeled forces are unbounded (i.e. motion is actually constrained by the environment). If the unmodeled forces are bounded, the disturbance would simply generate motion in their directions. For instance, if the normal force on a plane is stably controlled, the lateral forces are bounded by it through the friction cone. In that case, a disturbance increasing the lateral forces would simply make the robot slip. So this FL term is a safe compensation term to use in practical situations.

Remark 3. *The FL compensation term in Eq.* (10)*, could instead be added directly inside the MPC model, assuming that it remains constant over the whole horizon.*

IV. Experimental study

In this section, we evaluate the performance of the proposed approach through a comparative experimental study on a torque-controlled manipulator. First, we show the major advantage in tracking performance of using explicit force feedback over classical MPC. This benefit is twofold: force feedback enables to effectively cancel friction, and it corrects the model mismatch thanks to online estimation. Second, we demonstrate the benefit of encoding the model mismatch in the task space (ΔF) rather than in the joint space ($\Delta \tau$). Finally, we show how the proposed approach outperforms the most established force control strategy (integral control) by demonstrating that its force tracking performance is identical, but that it additionally aligns with the MPC objectives.

A. Experimental setup

All experiments were performed on the torque-controlled KUKA LBR iiwa R82014. We used an ATI F/T Sensor Mini40 mounted at the tip of the arm on a custom end-effector mount piece. A short MPC horizon (4 nodes of $6ms$) allowed to run the MPC and the estimator synchronously at $1\,\mathrm{kHz}$. The estimation QP problem (6) is solved using ProxQP [39], the OCP (1) is transcripted using Crocoddyl [40], and rigid-body dynamics are computed using Pinocchio [41]. Our code is publicly available[1]. Moreover, the accompanying video illustrates the robustness of the proposed approach to external disturbances.

B. Tasks formulation

1) Polishing task: A constant normal force is exerted on a horizontal plane (e_x, e_y) while tracking a circular end-effector trajectory. The MPC includes a 1D rigid contact force model ($n_c = 1$) so that the constraint (3b) prevents motions in the normal direction e_z, and ignores tangential forces in the (e_x, e_y) directions. The cost function is

$$\begin{aligned}\ell(x,u,t) = \; & w_1\|x(t)-\bar{x}(t)\|^2_{Q_1} + w_2\|u(t)-\bar{u}(t)\|^2_{Q_2} \\ & + w_3\|p^{\text{ee}}(t)-\bar{p}^{\text{ee}}(t)\|^2_{Q_3} + w_4\|F(t)-\bar{F}(t)\|^2_{Q_4} \\ & + w_5\|v^{\text{ee}}(t)\|^2_{Q_5} + w_6\|\log_3\left(\bar{R}^{\text{ee}}(t)^T R^{\text{ee}}(t)\right)\|^2_{Q_6}\end{aligned}$$

where $(w_i, Q_i)_{i=1..6}$ are positive scalar weights and positive diagonal activation matrices, $\bar{x}(t) = (\bar{q}(t), 0)$ is a reference configuration, $p^{\text{ee}}(t), F(t), R^{\text{ee}}(t)$ are the position of the end-effector, contact force and end-effector orientation respectively, $, \bar{p}^{\text{ee}}(t), \bar{F}(t), \bar{R}^{\text{ee}}(t)$ are their respective references, $v^{\text{ee}}(t)$ is the end-effector velocity, $\bar{u}(t) = g(q(t)) - J^T F(t)$ is the gravity compensation torque under external forces, $\log_3 : SO(3) \mapsto so(3)$ is the logarithm map on rotations. The circular trajectory $\bar{p}^{\text{ee}}(t)$ has a diameter of $14\,\mathrm{cm}$ and a speed of $3\,\mathrm{rad\,s^{-1}}$, unless otherwise stated. The reference normal force is constant $\bar{F} = 50\,\mathrm{N}$.

2) Force step tracking task: A 3D contact force ($n_c = 3$) step signal is tracked. Hence the motion of the end-effector is constrained in normal and tangential directions. The cost function has the same form as the polishing cost function (11), with the only differences that $F(t), \bar{F}(t)$ are 3D, the reference end-effector position $p^{\text{ee}}(t)$ is now constant and the force reference is defined as $\bar{F}(t) = \left(\bar{F}_x(t), \bar{F}_y(t), \bar{F}_z(t)\right))$ where $\bar{F}_x(t)$ is a step signal from $-10\,\mathrm{N}$ to $10\,\mathrm{N}$, $\bar{F}_y(t) = 0\,\mathrm{N}$ and $\bar{F}_z(t) = 100\,\mathrm{N}$ are constant.

3) Energy minimization: A sinusoidal joint position trajectory is tracked while maintaining a fixed 3D contact with the horizontal plane and minimizing $\|\tau\|^2$. The cost function is similar to the polishing (11), except that the reference configuration $\bar{q}(t)$ is no longer constant, no end-effector cost is used ($w_3 = w_5 = 0$), the control regularization term is turned into an energy term ($\bar{u}(t) = 0$). The reference joint trajectory is a sine on the A3 joint with an amplitude of $0.2\,\mathrm{rad}$ and a frequency of $2\,\mathrm{Hz}$. Here the force objective acts as a regularization term to avoid slipping and large forces (i.e. $w_4 \ll w_1, w_2, w_6$) and the reference is $\bar{F}(t) = (0, 0, 50)$.

C. Friction model vs direct measurement feedback (FL)

We evaluate the effect of force feedback as a direct compensation of the contact friction (Section III-C). We compare its performance on the polishing task against the classical MPC (i.e., without compensation) and the well-known Coulomb's friction model

$$F_T = -\mu \frac{v}{\|v\|} F_N, \tag{11}$$

where $F_T \in \mathbb{R}^2$ is the tangential force, $F_N \triangleq F \in \mathbb{R}$ is the normal force, $v \in \mathbb{R}^2$ is the tangential velocity of the contact point and μ is the dynamic friction coefficient. This model is clearly discontinuous in v so in order to avoid chattering phenomena, we consider the following smooth relaxation

$$F_T = -\mu \frac{\tanh(\epsilon\|v\|)}{\sqrt{2}} \frac{v}{\|v\|} F_N, \tag{12}$$

where we used $\mu = 0.35$ and $\epsilon = 10$. Our results are reported in Table I for several polishing speeds. We can see that the Coulomb model is slightly better in fast

[1] https://github.com/machines-in-motion/force_observer

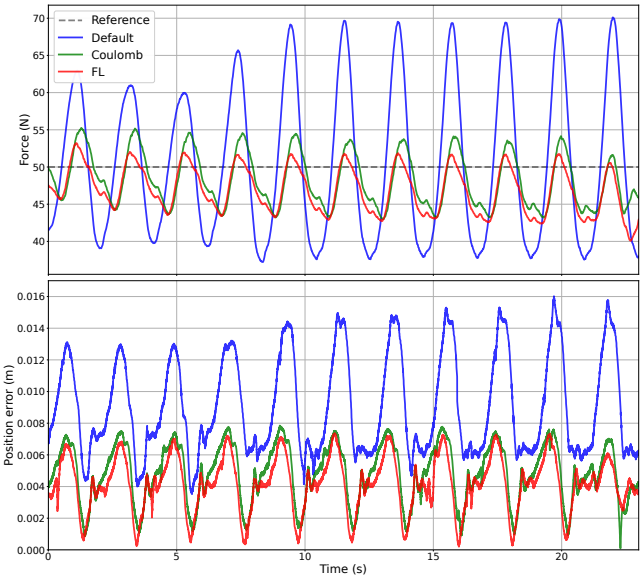


Fig. 1: Normal force trajectories of the medium-velocity polishing task. The blue curve is the classical MPC without friction compensation, the green curve is the classical MPC with the Coulomb model compensation, and the red curve is the classical MPC with FL compensation.

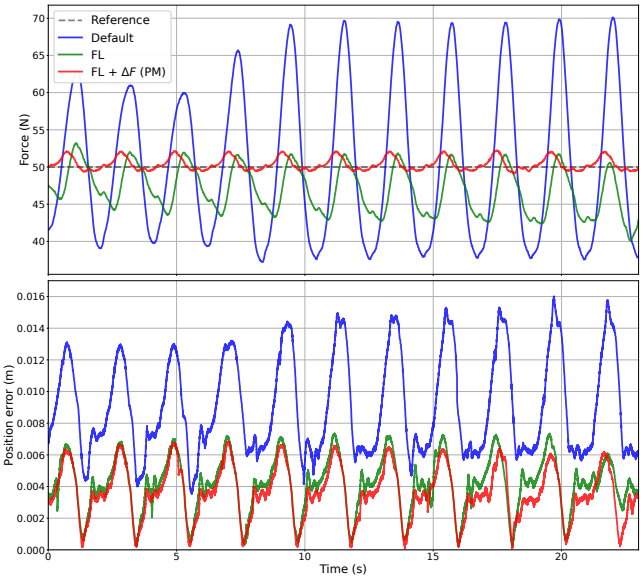


Fig. 2: Normal force (top) and end-effector position error (bottom) for the polishing task: in blue the classical MPC (II), in green the classical MPC with the FL compensation term (III-C), in red the proposed approach with FL compensation and the force offset in the predictive model (III-B.2).

		Default	FL	Coulomb
Slow	(1 rad/s)	7.67 ± 0.55	3.83 ± 0.17	4.72 ± 0.21
Medium	(3 rad/s)	9.66 ± 1.38	3.92 ± 0.56	3.99 ± 0.33
Fast	(6 rad/s)	16.42 ± 0.79	5.22 ± 0.32	4.82 ± 0.25

TABLE I: Mean-absolute error (MAE) of the normal force (in N) for the polishing task over 10 circles: classical MPC (Default), FL compensation (10) and Coulomb model (12).

	$\Delta\tau$	ΔF
Corrective control	2.01 ± 0.08	1.55 ± 0.03
Predictive model	1.95 ± 0.07	1.55 ± 0.04

TABLE II: MAE of the normal force (in N) for the polishing task: force offset ΔF vs. torque offset $\Delta\tau$, used in the control loop either in the "predictive model" way of III-B.2 or in the "corrective control" way of III-B.1.

motions but less performing in slow motions. Figure 1 shows the corresponding force trajectories for the medium-speed polishing task. Note that the FL compensation term only uses the $3D$ Jacobian as the contact torques a negligible in that task. These experiments confirm that considering the friction forces substantially increases performance w.r.t. classical MPC. Moreover, it shows that explicit force feedback from sensors can effectively be used as an FL term to directly to compensate friction effects and that it leads to a similar performance to well-established friction models.

As pointed out in Remark 3, it would be interesting to use the Coulomb model inside the MPC so that lateral forces are predicted using velocity and rigid normal force predictions, but this raises challenging issues (non-smoothness, insufficient software, breaks symmetry of KKT (2), etc.).

D. Comparison between force offset and torque offset

In this experiment, we compare the two mismatch models introduced in Section III-A, namely the torque offset $\Delta\tau$ and the force offset ΔF. Although capturing all disturbances in $\Delta\tau$ seems intuitive, experimental comparisons on the polishing task reveal a higher tracking accuracy for ΔF. For each model, we implemented the two ways of incorporating the correction into the MPC, namely

- The "corrective control" way of III-B.1: the correction is added to the optimal torque as a feedforward input
- The "predictive model" way of III-B.2: the correction is added directly to the model

Figure 2 illustrates how force feedback improves both the force tracking and the end-effector position tracking. Our results are summarized in Table II. There is a notable performance difference between ΔF and $\Delta\tau$ with a clear advantage for the force offset. Intuitively, the torque offset estimates perturbations unrelated to the contact (e.g. joint stiction) while the force offset only corrects what is necessary to improve the force tracking. There is, however, no clear difference in performance between using the estimate as a corrective control or in the predictive model. There seem to be a slight advantage for the predictive model, but the performance gap is too shallow to draw any conclusions.

E. Integral force control

Our approach is now compared to the most established direct force control approach - integral control. We were not able to find a difference in performance between using the integral term in the predictive model or as a corrective control. This question being out of the scope of this paper,

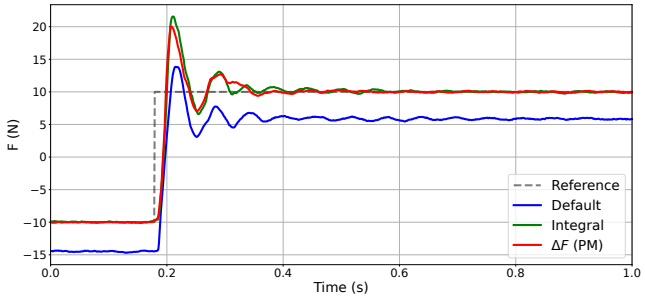


Fig. 3: Lateral force trajectories in the e_x direction for the force step tracking task: the blue curve is the classical MPC (Default), the green curve is the classical MPC with with integral control (Integral) and the red curve is the force offset estimation ΔF included in the predictive model (ΔF (PM)).

we propose to consider only the latter:

$$\tau = \tau_{\text{MPC}} - J(q)^T \left(-K_I \int_0^t \left(F(t') - \bar{F}(t') \right) dt' \right) \quad (13)$$

Note that we deliberately chose not to include a proportional and a derivative control term as Volpe et. al. [7] demonstrated both theoretically and experimentally that pure integral gain control was the best choice for accurate force tracking.

1) Polishing: We observed the same force tracking performance on the polishing task for the integral controller (1.69 ± 0.05 N) than for the proposed approach (cf. Table II, ΔF as corrective control).

2) Step experiment: We show in this experiment that the proposed approach and integral control have equivalent force tracking performances on a force step tracking task. The force trajectories are in Figure 3. We also report the average force tracking error of all the controllers in Table III.

	Avg. error
Default	1.99
ΔF (predictive model)	0.71
ΔF (corrective control)	0.60
$\Delta \tau$ (predictive model)	0.80
$\Delta \tau$ (corrective control)	0.87
Integral control	0.68

TABLE III: MAE of the normal force error for a step tracking task for different controllers: classical MPC (Default), force offset estimation (ΔF), torque offset estimation ($\Delta \tau$) and integral control. ΔF and $\Delta \tau$ are used as corrective control (III-B.1) or in the predictive model (III-B.2).

3) Energy minimization: In this experiment, we illustrate the ability of force feedback MPC to achieve contact tasks with conflicting objectives. Table IV shows how the proposed force estimation approach aligns with the MPC objectives by trading off force tracking against energy minimization: its overall cost is lower than the integral controller, which conflicts with the MPC and generates a high cost. These results also show interestingly that somehow, the torque offset estimation ($\Delta \tau$) uses less energy than the force offset estimation (ΔF), although it yields a slightly higher cost overall. This suggests that encoding the mismatch as a joint torque offset may have its own benefits, other than accurate force tracking. The accompanying video illustrates the relative importance of $w_2 \|\tau\|^2_{Q_2}$ w.r.t. the total cost.

	Avg. $\|\tau\|^2$	Total cost
Default	136 ± 21	0.44 ± 0.02
ΔF (predictive model)	139 ± 13	0.43 ± 0.01
ΔF (corrective control)	145 ± 18	0.43 ± 0.02
$\Delta \tau$ (predictive model)	131 ± 21	0.48 ± 0.01
$\Delta \tau$ (corrective control)	132 ± 22	0.51 ± 0.02
Integral control	1052 ± 29	0.82 ± 0.027

TABLE IV: Average squared torque and total cost for each controller for the energy task: classical MPC (Default), force offset estimation (ΔF), torque offset estimation ($\Delta \tau$) and integral control. ΔF and $\Delta \tau$ are used as corrective control (III-B.1) or in the predictive model (III-B.2).

V. Conclusion

In this work, we proposed a simple approach to achieve force feedback in MPC that relies on the online estimation of the mismatch between the predicted forces and the force measurements. Our experiments showed that force feedback effectively cancels friction and brings the force tracking performance to the level of the most established direct force control strategies. We also studied two variants of our approach: the estimation of a torque offset in the joint space, and the estimation of a force offset in the task space. Our experiments show that the force offset yields a more accurate force tracking while the torque offset is more generic and can enhance other criteria (e.g., energy minimization).

In conclusion, our experimental results show that current optimization-based control and estimation techniques are sufficient to incorporate force sensors in model-predictive controllers and suggest a more systematic exploitation of those modalities on real robots. In future work, it would be interesting to add the integral error as part of an augmented state in the MPC. Also, the estimation could be done over a horizon (although it has not led to any improvement so far in our trials), and the assumption of perfect joint position and velocity measurements could be relaxed, although this would turn the estimation problem into a nonlinear program. Finally, it would be interesting to extend the proposed methodology to floating base robots.

Acknowledgments

This work was supported by the National Science Foundation grants 1932187, 2026479, 2222815 and 2315396, the French government under the management of Agence Nationale de la Recherche through the NIMBLE project (ANR-22-CE33-0008) and through the "Investissements d'avenir" program (PRAIRIE ANR-19-P3IA-0001 and ANITI ANR-19-P3IA-0004) and by the European Union through the AGIMUS project (GA no.101070165). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

REFERENCES

[1] D. Whitney, "Historical perspective and state of the art in robot force control," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 262–268.

[2] E. Corral, R. Moreno, M. J. G. García, and C. Castejón, "Nonlinear phenomena of contact in multibody systems dynamics: a review," *Nonlinear Dynamics*, vol. 104, pp. 1269 – 1295, 2021.

[3] R. Featherstone, *Rigid Body Dynamics Algorithms*, 2008.

[4] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[5] L. Villani and J. De Schutter, *Force Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 161–185.

[6] M. Schumacher, J. Wojtusch, P. Beckerle, and O. von Stryk, "An introductory review of active compliant control," *Robotics and Autonomous Systems*, vol. 119, pp. 185–200, 2019.

[7] R. Volpe and P. Khosla, "Theoretical and experimental investigation of explicit force control strategies for manipulators," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1634–1650, 1993.

[8] M. Raibert and J. J. Craig, "Hybrid Position / Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. June 1981, pp. 126–133, 1981.

[9] S. Chiaverini and L. Sciavicco, "The Parallel Approach to Force/Position Control of Robotic Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.

[10] B. Siciliano, "Parallel force/position control of robot manipulators," in *Robotics Research*. Springer London, 1996, pp. 78–89.

[11] J. Duffy, "The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces," *Journal of Robotic Systems*, vol. 7, no. 2, pp. 139–144, 1990.

[12] T. Yoshikawa, "Force control of robot manipulators," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, no. April, pp. 220–226, 2000.

[13] N. Hogan, "Contact and Physical Interaction," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 1–25, 2022.

[14] ——, "Impedance Control Part1-3," *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. March 1985, pp. 1–24, 1985.

[15] D. E. Whitney, "Force feedback control of manipulator fine motions," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 99, no. 2, pp. 91–97, 1977.

[16] W. S. Newman, "Stability and performance limits of interaction controllers," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 114, no. 4, pp. 563–570, 1992.

[17] H. Seraji, "ADAPTIVE ADMITTANCE CONTROL: An Approach to Explicit Force Control," *Jet Propulsion*, pp. 2705–2712, 1994.

[18] H. Seraji and R. Colbaugh, "Force tracking in impedance control," *International Journal of Robotics Research*, vol. 16, no. 1, pp. 97–117, 1997.

[19] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control for robot manipulators with an unknown environment: Theory, simulation, and experiment," *International Journal of Robotics Research*, vol. 20, no. 9, pp. 765–774, 2001.

[20] D. Erickson, M. Weber, and I. Sharf, "Contact Stiffness and Damping Estimation for Robotic Systems," *International Journal of Robotics Research*, vol. 22, no. 1, pp. 41–57, 2003.

[21] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-Time motion planning of legged robots: A model predictive control approach," *IEEE-RAS International Conference on Humanoid Robots*, pp. 577–584, 2017.

[22] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," *Autonomous Robots*, vol. 40, no. 3, pp. 537–560, 2016.

[23] J. Matschek, J. Bethge, P. Zometa, and R. Findeisen, "Force Feedback and Path Following using Predictive Control: Concept and Application to a Lightweight Robot," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9827–9832, 2017.

[24] A. Wahrburg and K. Listmann, "MPC-based admittance control for robotic manipulators," *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, pp. 7548–7554, 2016.

[25] K. J. Kazim, J. Bethge, J. Matschek, and R. Findeisen, "Combined Predictive Path Following and Admittance Control," *Proceedings of the American Control Conference*, vol. 2018-June, pp. 3153–3158, 2018.

[26] M. Bednarczyk, H. Omran, and B. Bayle, "Model Predictive Impedance Control," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 1, pp. 4702–4708, 2020.

[27] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter, "Model predictive robot-environment interaction control for mobile manipulation tasks," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1651–1657, 2021.

[28] T. Gold, A. Völz, and K. Graichen, "Model predictive interaction control for robotic manipulation tasks," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 76–89, 2023.

[29] S. Kleff, E. Dantec, G. Saurel, N. Mansard, and L. Righetti, "Introducing force feedback in model predictive control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 379–13 385.

[30] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 556–563, 2015.

[31] W. Amanhoud, M. Khoramshahi, M. Bonnesoeur, and A. Billard, "Force Adaptation in Contact Tasks with Dynamical Systems," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 6841–6847, 2020.

[32] A. Lawitzky, A. Nicklas, D. Wollherr, and M. Buss, "Determining states of inevitable collision using reachability analysis," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4142–4147, 2014.

[33] K. J. Åström, *Adaptive Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 437–450.

[34] F. E. Udwadia and R. E. Kalaba, "A new perspective on constrained motion," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1906, pp. 407–410, 1992.

[35] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *IEEE Humanoids*, 2018.

[36] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2007.

[37] S. A. U. Islam and D. S. Bernstein, "Recursive least squares for real-time implementation [lecture notes]," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 82–85, 2019.

[38] A. Dietrich, X. Wu, K. Bussmann, M. Harder, M. Iskandar, J. Englsberger, C. Ott, and A. Albu-Schäffer, "Practical consequences of inertia shaping for interaction and tracking in robot control," *Control Engineering Practice*, vol. 114, 2021.

[39] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond," in *RSS 2022 - Robotics: Science and Systems*, 2022.

[40] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[41] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," *IEEE/SICE International Symposium on System Integration*, 2019.