Time-Optimal Trajectory Planning With Interaction With the Environment

Vincenzo Petrone , Enrico Ferrentino , and Pasquale Chiacchio, *Senior Member, IEEE*

Abstract—Optimal motion planning along prescribed paths can be solved with several techniques, but most of them do not take into account the wrenches exerted by the end-effector when in contact with the environment. When a dynamic model of the environment is not available, no consolidated methodology exists to consider the effect of the interaction. Regardless of the specific performance index to optimize, this article proposes a strategy to include external wrenches in the optimal planning algorithm, considering the task specifications. This procedure is instantiated for minimum-time trajectories and validated on a real robot performing an interaction task under admittance control. The results prove that the inclusion of end-effector wrenches affects the planned trajectory, in fact modifying the manipulator's dynamic capability.

Index Terms—Constrained motion planning, compliance and impedance control, optimization and optimal control.

I. INTRODUCTION

MOTION planning of robotic manipulators is the problem of determining a joint space trajectory, suitable for the robot controller to execute. The process is usually divided into two hierarchical sub-tasks. First, a geometric *path* is defined in task or joint space, considering specific task requirements, environmental obstacles, joint mechanical limits. Then, a time law is assigned, resulting in a *trajectory* respecting actuator constraints, such as maximum joint velocities and torques [1].

One common approach to time parametrization is to optimize some performance function, usually determined by the robot operational conditions. When the objective is to minimize the trajectory tracking time, the problem is usually referred to as *Time-Optimal Trajectory Planning* (TOTP) [2].

A. Motivation and Scope

Considering motion planning with the aforementioned decoupled approach, the focus of this letter is on TOTP. The process input is a joint space path, possibly derived via inverse kinematics (IK), if the task is initially defined in task space.

The importance of minimum time trajectories has been widely discussed in the literature. In manufacturing, it is connected to the maximization of the plant throughput [1]; in space, it might help optimizing the operations of assets whose usage is notably expensive [3].

In practice, TOTP can be addressed with several strategies, a concise overview of which is reported in [4], and by modeling the robotic system and its constraints in many different ways. The resolution strategy depends on the selected model and the constraints of interest. Numerical integration approaches deal with time optimality through the explicit maximization of the path velocity [5], [6]. Convex optimization techniques [1] address the problem from the same perspective, but viscous friction effects and limits on the trajectory's third-order derivatives cannot be included. Dynamic Programming (DP) can account for arbitrary forms of constraints and cost functions, but can be extremely slow [7], [8]. Recently, a novel evolutionary approach based on genetic algorithms [3] has been proposed to specifically account for singular points.

The limitation of the state-of-the-art techniques is that they assume the manipulator's end-effector not in contact with the surrounding environment. This seldom is a realistic hypothesis in nowadays scenarios, since the robot certainly has to interact with the working environment to modify it. Welding, gluing and assembly are common examples of industrial tasks requiring interaction. If external contacts are not included in planning, but they do occur during the execution, the trajectory could be infeasible, i.e. the trajectory tracking performance could degrade. The reason of poor tracking lies in the wrong assumption made on the dynamic modeling of the system. If the torques exploited to exert wrenches on the environment are not considered in the planning phase, then the ones required to produce the motion might not be available at execution.

If the task is known a priori, and a dynamic characterization of the interaction can be devised, then the resulting wrenches can be explicitly included in planning [9], through a simple extension of the parametrized dynamic model originally proposed in the seminal works [10] and [11]. In this communication, the more realistic assumption is made that the interaction wrenches cannot be precisely modeled, but lower and upper limits are available from the task specification. Dynamic programming is preferred as the underlying resolution technique due to its flexibility [7].

When it comes to trajectory execution, the problem of interacting with unknown environments can be addressed at control level with the employment of variable impedance controllers, with the environment being progressively modeled through learning [12]. This letter focuses on planning, leaving the actual success of

the execution to the specific controller employed for trajectory tracking. Yet, some details about the employed controller, used to support the experiments on the UR10 robot, are discussed.

In view of the aforementioned considerations, in this letter, a modification of the state-of-the-art TOTP algorithm is proposed. In particular, by making the assumption that no information about the environment is available, only upper and lower bounds for interaction wrenches are included at planning level, as part of the task specification. To this aim, the equations provided to the TOTP solver are extended to account for these bounds. The planned trajectory is eventually executed on a real manipulator with an admittance controller, to allow the robot to interact with a blackboard.

B. Overview

The remainder of this letter is organized as follows. Section II presents the dynamic model adopted in the planning process. Section III describes the control scheme used in the trajectory execution phase. Section IV illustrates the proposed methodology, highlighting how interaction wrenches can be included in the planning algorithm. Section V reports the experimental validation, showing how external wrenches affect TOTP. In addition, the results of executing a time-optimal planned interaction trajectory on an admittance-controlled real manipulator are presented.

II. DYNAMIC MODELING OF ROBOTIC ARMS

Consider a robotic manipulator with n Degrees Of Freedom (DOF), being assigned a task of $m \leq n$ DOF. Suppose the end-effector (EE) in contact with an (unknown) environment, such that the manipulator exerts some forces $\boldsymbol{f}_e$ and momenta $\boldsymbol{m}_e$, collected in the external wrench vector $\boldsymbol{h}_e := (\boldsymbol{f}_e, \boldsymbol{m}_e)^T$.

Said $\boldsymbol{q}$ the $n \times 1$ vector of joint variables, the Equations Of Motion (EOM) relate the manipulator's generalized torques at the actuators $\boldsymbol{\tau}$ to the joint positions $\boldsymbol{q}(t)$ and the corresponding time derivatives, namely the joint velocities $\dot{\boldsymbol{q}}(t)$ and accelerations $\ddot{\boldsymbol{q}}(t)$. The EOM can be expressed in a matrix form with the following dynamic model:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T \boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e, \quad (1)$$

where $\boldsymbol{B}(\boldsymbol{q})$ is the $n \times n$ inertia matrix; $\boldsymbol{C}(\boldsymbol{q})$ is the $n \times n \times n$ centrifugal and Coriolis effects matrix; $\boldsymbol{F}$ is the diagonal $n \times n$ friction matrix; $\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e$ are the torques needed to exert the wrenches $\boldsymbol{h}_e$ at the EE, expressed in the base frame, with $\boldsymbol{J}(\boldsymbol{q})$ the manipulator's Jacobian.

It is worth remarking that (1) considers a model for friction: since high-speed trajectories are supposed to be commanded to the manipulator, a considerable portion of actuation capabilities is exploited in compensating this component. Thus, the term cannot be ignored in the model for TOTP, otherwise non-negligible inaccuracies are introduced in the torques computation.

III. TASK-SPACE ADMITTANCE CONTROL FOR POSITION-CONTROLLED ROBOTS

The feasibility of a wide number of practical tasks relies on the control of the physical interaction between the robot and the EE,
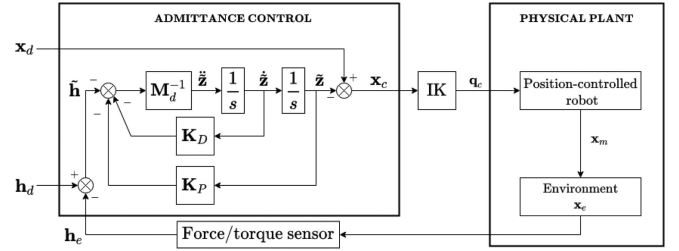

Fig. 1. Admittance control block scheme for position-controlled robots.

as pure motion control is possible only if an accurate planning of the task is available, requiring an exact characterization of the environment's geometrical and mechanical features. Since this hypothesis is unlikely to hold, interaction control has been largely studied in the last decades [13].

Interaction controllers are employed to control the manipulator motion while considering the presence of wrenches arising as an effect of the interaction. Impedance and admittance control are two popular approaches, being the latter more suitable for position-controlled robots. In fact, given a certain force error as input, it produces a corresponding displacement as output, according to a mechanical admittance. Thus, when a desired task-space trajectory $\boldsymbol{x}_d(t)$ is received, the control action modifies the position reference, producing a compliant trajectory $\boldsymbol{x}_c(t)$ to be commanded to the robot. This is of utmost importance as industrial robots typically expose a position interface [12].

Suppose that, along with a task space desired trajectory $\boldsymbol{x}_d(t)$, a reference wrench $\boldsymbol{h}_d(t)$ is given, too. Denote with $\tilde{\boldsymbol{h}} := \boldsymbol{h}_d - \boldsymbol{h}_e$ the wrench error and with $\tilde{\boldsymbol{z}} := \boldsymbol{x}_d - \boldsymbol{x}_c$ the admittance control action, namely the deviation from the desired reference to the compliant task space command. The admittance control imposes mass-spring-damper EE dynamics, described by

$$\boldsymbol{M}_d \ddot{\tilde{\boldsymbol{z}}} + \boldsymbol{K}_D \dot{\tilde{\boldsymbol{z}}} + \boldsymbol{K}_P \tilde{\boldsymbol{z}} = -\tilde{\boldsymbol{h}}, \quad (2)$$

where $\boldsymbol{M}_d$, $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$ are 6×6 diagonal matrices corresponding, respectively, to mass, damping and stiffness matrices. The coefficients in the matrices are the control parameters, to be suitably selected depending on the desired interaction dynamics.

From (2), the task space compliant frame is computed:

$$\ddot{\tilde{\boldsymbol{z}}} = \boldsymbol{M}_d^{-1}(-\tilde{\boldsymbol{h}} - \boldsymbol{K}_D \dot{\tilde{\boldsymbol{z}}} - \boldsymbol{K}_P \tilde{\boldsymbol{z}}), \quad (3a)$$

$$\dot{\tilde{\boldsymbol{z}}} = \int_t \ddot{\tilde{\boldsymbol{z}}} \, dt, \quad (3b)$$

$$\tilde{\boldsymbol{z}} = \int_t \dot{\tilde{\boldsymbol{z}}} \, dt, \quad (3c)$$

$$\boldsymbol{x}_c = \boldsymbol{x}_d - \tilde{\boldsymbol{z}}. \quad (3d)$$

Finally, the joint-space command $\boldsymbol{q}_c$ is obtained by IK from $\boldsymbol{x}_c$, and set as the reference of the inner position loop, usually provided by the robot's manufacturer. If in contact with the environment, the resulting EE pose $\boldsymbol{x}_m$ produces the wrenches $\boldsymbol{h}_e$, as a result of the environment rest position $\boldsymbol{x}_e$ and the corresponding virtual penetration $\Delta \boldsymbol{x} := \boldsymbol{x}_e - \boldsymbol{x}_m$.

Fig. 1 shows the block scheme corresponding to (2) and (3), highlighting how the wrench $\boldsymbol{h}_e$ is fed back to the control loop.

To this aim, a force/torque sensor is implied in this work, but other solutions are possible to estimate the EE wrench, even without additional sensing [14], [15].

IV. TIME OPTIMAL TRAJECTORY PLANNING WITH INTERACTION WITH THE ENVIRONMENT

In general, trajectory planning deals with assigning a time law to a pre-designed joint space path $\boldsymbol{q}(\lambda)$ or task space path $\boldsymbol{x}(\lambda)$, where λ is a curvilinear coordinate describing the path geometry as a monotonic function of time, i.e.

$$\lambda(t) : \mathbb{R} \to [0; \Lambda], \dot{\lambda} = \frac{d\lambda}{dt} > 0 \ \forall t, \tag{4}$$

with Λ being the total path length. The relationship $\lambda(t)$ is the time law to find as a result of the time parametrization process, which is usually addressed through optimization. Once the optimal time law is found, the path can be time-parametrized, obtaining the joint space trajectory $\boldsymbol{q}(\lambda(t)) = \boldsymbol{q}(t)$.

In this communication, TOTP is considered, meaning that $\lambda(t)$ is computed so as to minimize the overall duration of the task, considering robot-specific architectural limits and other application-dependent constraints. Among them, the inclusion of external interaction wrenches is considered.

A. Dynamic Model Parametrization

Suppose that a task space path $\boldsymbol{x}(\lambda)$ is available from the task specification. Suppose to plan for a non-redundant robot, thus a joint space path $\boldsymbol{q}(\lambda)$ can be easily computed via IK.

Denote with $\boldsymbol{q}'(\lambda)$ and $\boldsymbol{q}''(\lambda)$ the first- and second-order derivatives of $\boldsymbol{q}(\lambda)$ with respect to λ. The time derivatives are obtained by simply applying the chain rule:

$$\dot{\boldsymbol{q}} = \boldsymbol{q}'(\lambda)\dot{\lambda}, \tag{5}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{q}'(\lambda)\ddot{\lambda} + \boldsymbol{q}''(\lambda)\dot{\lambda}^2. \tag{6}$$

Plugging (5) and (6) into (1) yields

$$\boldsymbol{a}(\lambda)\ddot{\lambda} + \boldsymbol{b}(\lambda)\dot{\lambda}^2 + \boldsymbol{c}(\lambda)\dot{\lambda} + \boldsymbol{g}(\lambda) = \boldsymbol{\tau} - \boldsymbol{J}^T(\lambda)\boldsymbol{h}_e(\lambda), \tag{7}$$

where

$$\boldsymbol{a}(\lambda) = \boldsymbol{B}\left(\boldsymbol{q}(\lambda)\right)\boldsymbol{q}'(\lambda), \tag{8}$$

$$\boldsymbol{b}(\lambda) = \boldsymbol{B}\left(\boldsymbol{q}(\lambda)\right)\boldsymbol{q}''(\lambda) + \left(\boldsymbol{q}'(\lambda)\right)^T \boldsymbol{C}\left(\boldsymbol{q}(\lambda)\right)\boldsymbol{q}'(\lambda), \tag{9}$$

$$\boldsymbol{c}(\lambda) = \boldsymbol{F}\boldsymbol{q}'(\lambda), \tag{10}$$

$$\boldsymbol{g}(\lambda) = \boldsymbol{g}\left(\boldsymbol{q}(\lambda)\right), \tag{11}$$

$$\boldsymbol{J}(\lambda) = \boldsymbol{J}\left(\boldsymbol{q}(\lambda)\right). \tag{12}$$

As regards the wrench $\boldsymbol{h}_e$, a λ-parametrization is possible only if a dynamic characterization of the environment is available. For instance, [9] proposes a dynamic model for a manipulated object, and assumes the external wrench to result from the object's planned motion. When the interaction model is not known, there is no particular strategy to plan for external wrenches. However, they can be included with the meaning of expected maximum and minimum exerted wrenches, given the task specification.

B. Constraints Parametrization

The TOTP problem is subject to constraints, given by the manipulator structure and the specific task. It is useful to define such limits with lower and upper bounds on a constrained variable, denoted with ($\bullet$) and ($\overline{\bullet}$) respectively.

The robot inherent technological limits must be included in planning, so as to reach a feasible trajectory in terms of the actual manipulator capabilities. Typically, the joint actuators are limited in both torques and velocities. For each $j \in \{0, \dots, n\}$, the j-th joint's maximum and minimum torque and velocity limits are respectively expressed as

$$\underline{\tau}_j \leq \tau_j \leq \overline{\tau}_j, \tag{13}$$

$$\underline{\dot{q}}_j \leq \dot{q}_j \leq \overline{\dot{q}}_j. \tag{14}$$

An exact characterization of the robot-environment interaction is hard to devise. More commonly, depending on the task specification, it is possible to define lower and upper bounds on the external wrenches vector. Typically, they correspond to the maximum and minimum wrenches that the EE and the environment can tolerate, or that are required by the task to achieve a certain level of quality. These can be seen as application-dependent constraints:

$$\underline{\boldsymbol{h}_e}(\lambda) \leq \boldsymbol{h}_e \leq \overline{\boldsymbol{h}_e}(\lambda). \tag{15}$$

As the dependency on λ suggests, it is worth remarking that such limits do not need to be constant, but can be adapted along the path.

The task specifications, including the maximum stress the working environment must be subject to, are usually detailed in industrial contexts, whilst are typically not well-defined in service tasks. In such cases, the bounds could be identified by performing preliminary experiments and recording the measured wrenches, thus deciding upon the limits to enforce according to a certain degree of safety or quality of the executed task. If it is not possible to replicate the real-case scenario in which the planned trajectory has to be executed, and a physical copy of the system is not available, then at least a high-fidelity simulated environment should be adopted for that purpose.

One typical choice for the limit wrenches for common interaction tasks is when the contact only concerns one direction. For simplicity, it can be made to correspond with a specific axis of the base frame, for instance

$$\underline{\boldsymbol{h}_e}(\lambda) = \left(\underline{f}_N(\lambda), \underline{f}_{T_1}(\lambda), \underline{f}_{T_2}(\lambda), 0, 0, 0\right)^T,$$

$$\overline{\boldsymbol{h}_e}(\lambda) = \left(\overline{f}_N(\lambda), \overline{f}_{T_1}(\lambda), \overline{f}_{T_2}(\lambda), 0, 0, 0\right)^T,$$

where $f_N(\lambda)$ is the normal force along the contact direction (which is supposed to be the x axis in the base frame), and $f_{T_1}(\lambda)$ and $f_{T_2}(\lambda)$ are the tangential force components along the orthogonal motion directions, represented by the unit vector $\boldsymbol{t} = (t_1, t_2)^T$, defined in the tangential plane. The maximum and minimum tangential force components relate to the normal force according to

$$\underline{f}_{T_1}(\lambda) = \mu \underline{f}_N(\lambda)t_1(\lambda) \quad \overline{f}_{T_1}(\lambda) = \mu \overline{f}_N(\lambda)t_1(\lambda)$$

$$\underline{f}_{T_2}(\lambda) = \mu \underline{f}_N(\lambda)t_2(\lambda) \quad \overline{f}_{T_2}(\lambda) = \mu \overline{f}_N(\lambda)t_2(\lambda)$$

where $\mu > 0$ is the friction coefficient. It can be available by knowledge of the EE and environment surfaces or easily estimated from trials, as in the experimental case of Section V. Clearly, the tangential forces can, in general, behave according to a more complicated (or, as in the worst case, completely unknown) model. If a model is supposed for the tangential components, its parameters must be estimated, and the model itself must be validated during preliminary experiments. Otherwise, tangential forces can be given, as for normal forces, in terms of independent upper and lower bounds that are unrelated to the normal force ones. The methodology we propose does not restrict the way tangential forces bounds are devised.

Denote with $\gamma_j(\lambda) := \boldsymbol{J}_j^T(\lambda)\boldsymbol{h}_e(\lambda)$ the torque to apply to joint j to exert wrenches $\boldsymbol{h}_e$ on the environment in a specific configuration $\boldsymbol{q}(\lambda)$. Then, starting from (13), the following inequality holds

$$\underline{\tau_j} - \gamma_j(\lambda) \le \tau_j - \gamma_j(\lambda) \le \overline{\tau_j} - \gamma_j(\lambda). \tag{16}$$

Because of (15), γ_j can vary in the range

$$\boldsymbol{J}_j^T(\lambda)\underline{\boldsymbol{h}_e}(\lambda) \le \gamma_j(\lambda) \le \boldsymbol{J}_j^T(\lambda)\overline{\boldsymbol{h}_e}(\lambda), \tag{17}$$

hence $\tau_j - \gamma_j$ must respect the strictest limit given by (17), i.e.

$$\max \left\{ \underline{\tau_j} - \underline{\gamma_j}(\lambda), \underline{\tau_j} - \overline{\gamma_j}(\lambda) \right\} \le \tau_j - \gamma_j(\lambda)$$
$$\le \min \left\{ \overline{\tau_j} - \underline{\gamma_j}(\lambda), \overline{\tau_j} - \overline{\gamma_j}(\lambda) \right\}, \tag{18}$$

where

$$\underline{\gamma_j}(\lambda) = \boldsymbol{J}_j^T(\lambda)\underline{\boldsymbol{h}_e}(\lambda), \tag{19}$$

$$\overline{\gamma_j}(\lambda) = \boldsymbol{J}_j^T(\lambda)\overline{\boldsymbol{h}_e}(\lambda). \tag{20}$$

By taking $\underline{\tau_j}$ and $\overline{\tau_j}$ out of the min and max operators, (18) becomes

$$\underline{\tau_j} + \max \left\{ -\underline{\gamma_j}(\lambda), -\overline{\gamma_j}(\lambda) \right\} \le \tau_j - \gamma_j(\lambda)$$
$$\le \overline{\tau_j} + \min \left\{ -\underline{\gamma_j}(\lambda), -\overline{\gamma_j} \right\},$$

from which it follows that

$$\underline{\tau_j} - \min \left\{ \underline{\gamma_j}(\lambda), \overline{\gamma_j}(\lambda) \right\} \le \tau_j - \gamma_j(\lambda)$$
$$\le \overline{\tau_j} - \max \left\{ \underline{\gamma_j}(\lambda), \overline{\gamma_j}(\lambda) \right\}. \tag{21}$$

Plugging (7) into the constraint (21) yields

$$\underline{\tau_j} - \underline{\tau_h}_j(\lambda) \le a_j(\lambda)\ddot{\lambda} + b_j(\lambda)\dot{\lambda}^2 + c_j(\lambda)\dot{\lambda} + g_j(\lambda)$$
$$\le \overline{\tau_j} - \overline{\tau_h}_j(\lambda), \tag{22}$$

where

$$\underline{\tau_h}_j(\lambda) = \min \left\{ \underline{\gamma_j}(\lambda), \overline{\gamma_j}(\lambda) \right\}, \tag{23}$$

$$\overline{\tau_h}_j(\lambda) = \max \left\{ \underline{\gamma_j}(\lambda), \overline{\gamma_j}(\lambda) \right\}. \tag{24}$$

Solving inequality (22) for $\ddot{\lambda}$ and accounting for the strictest limit among all n joints, the following constraint is obtained:

$$L(\lambda, \dot{\lambda}) \le \ddot{\lambda} \le U(\lambda, \dot{\lambda}), \tag{25}$$

where

$$L(\lambda, \dot{\lambda}) = \max_{j=1,\dots,n} L_j(\lambda, \dot{\lambda}), \tag{26}$$

$$U(\lambda, \dot{\lambda}) = \min_{j=1,\dots,n} U_j(\lambda, \dot{\lambda}), \tag{27}$$

with

$$L_j(\lambda, \dot{\lambda}) = \frac{\begin{array}{c}(\underline{\tau}_j - \underline{\tau_h}_j(\lambda))\delta_j(\lambda)+ \\ + (\overline{\tau}_j - \overline{\tau_h}_j(\lambda))(1 - \delta_j(\lambda)) + \\ -b_j(\lambda)\dot{\lambda}^2 - c_j(\lambda)\dot{\lambda} - g_j(\lambda)\end{array}}{a_j(\lambda)}, \tag{28}$$

$$U_j(\lambda, \dot{\lambda}) = \frac{\begin{array}{c}(\overline{\tau}_j - \overline{\tau_h}_j(\lambda))\delta_j(\lambda)+ \\ + (\underline{\tau}_j - \underline{\tau_h}_j(\lambda))(1 - \delta_j(\lambda)) + \\ -b_j(\lambda)\dot{\lambda}^2 - c_j(\lambda)\dot{\lambda} - g_j(\lambda)\end{array}}{a_j(\lambda)} \tag{29}$$

and

$$\delta_j(\lambda) = \begin{cases} 1, & \text{if } a_j(\lambda) > 0 \\ 0, & \text{if } a_j(\lambda) < 0 \end{cases}. \tag{30}$$

While (25) bounds $\ddot{\lambda}$, (14) produces a limit on $\dot{\lambda}$, since joint velocities are a function of the pseudo-velocity, as evident from (5):

$$\max_{j=1,\dots,n} \frac{\underline{\dot{q}}_j}{q'_j(\lambda)} \le \dot{\lambda} \le \min_{j=1,\dots,n} \frac{\overline{\dot{q}}_j}{q'_j(\lambda)}. \tag{31}$$

C. Solution Using a Dynamic Programming Algorithm

The objective of the TOTP algorithm is to minimize the overall trajectory tracking time. Let $\boldsymbol{s}(t) = [\lambda(t), \dot{\lambda}(t)]^T$ be the *state* and $\ddot{\lambda}(t)$ the *control input* of the considered dynamical system. The problem of finding the minimum-time trajectory can be stated as follows:

$$\min_{\ddot{\lambda}(t) \in [L(\boldsymbol{s}), U(\boldsymbol{s})]} \int_0^{t_f} dt, \tag{32}$$

subject to (31) and the following boundary conditions:

$$\boldsymbol{s}(0) = [0, 0]^T,$$
$$\boldsymbol{s}(t_f) = [\Lambda, 0]^T, \tag{33}$$

ensuring that the robot will start and stop at rest.

In this letter, the same DP algorithm as in [7] is adopted to solve the TOTP problem above, i.e. the phase plane $(\lambda$-$\dot{\lambda})$ is discretized to get a rectangular grid of $N_\lambda \times N_{\dot{\lambda}}$ cells. The cost function uses the approximation proposed in [8] for the time discretization from one cell to another.

By employing such a framework, valid nodes for the solution are selected that satisfy conditions (25), (31) and (33) for each node in the grid. After the trajectory is planned, the parametric velocities and accelerations are turned into joint velocities and accelerations according to (5) and (6).
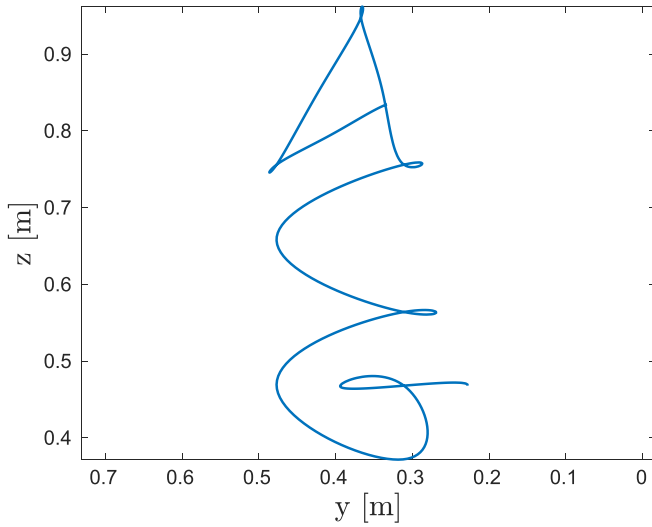
Fig. 2. The path to time-parametrize, represented on the writing plane, i.e. (y, z) in the base frame.
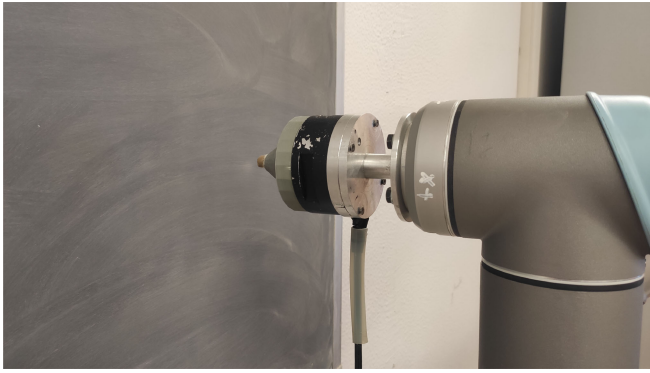

Fig. 3. Experimental setup: the UR10 robot is equipped with a force/torque sensor, on top of which a mechanical interface is mounted, with a piece of chalk inserted in it.

The reader should be aware that DP is not the only technique suitable for implementing a TOTP algorithm, but the choice of the dynamic model (1), with the inclusion of viscous friction effects, and the corresponding λ-parametrization (7), breaks the limitations of the convex optimization approach [1], as also pointed out in [3], [4].

Also, it is worth remarking that, since the proposed methodology only replaces the limits to be imposed in the optimization process, it can be employed to optimize the time law with respect to any desirable performance index.

V. EXPERIMENTAL RESULTS

The interaction task is a blackboard writing task, i.e. the 6-DOF UR10 manipulator is required to write the Automatic Control Group (ACG) acronym (illustrated in Fig. 2) on a blackboard using some chalk. The EE is made of a piece of chalk inserted in a properly 3D modeled and printed mechanical interface.

The experimental setup is shown in Fig. 3. With reference to the base frame, the contact direction is the x axis, while the free motion directions (that define the "writing plane") are the y and z axes. The wrenches are sensed via a flange-mounted 6-DOF

TABLE I
TORQUE AND VELOCITY LIMITS USED FOR TOTP

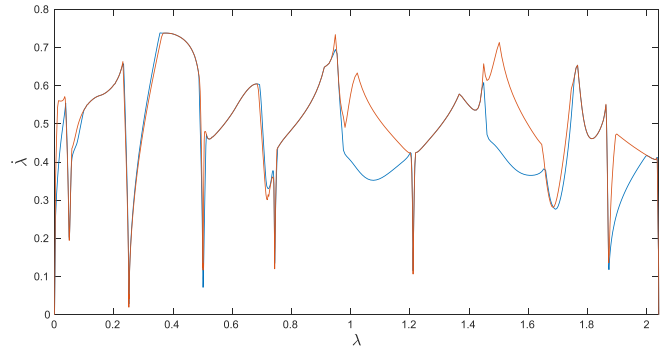| axis	joint name	$\bar{q} = -\dot{q}$ [°/s]	$\overline{\tau} = -\underline{\tau}$ [Nm]
1	base	40	80
2	shoulder	40	80
3	elbow	60	100
4	wrist 1	60	28
5	wrist 2	60	28
6	wrist 3	60	28

Fig. 4. Phase plane trajectory on the $(\lambda$-$\dot{\lambda})$ plane. In blue, the external wrenches are considered; in orange, they are neglected.

force/torque sensor. The chalk can tolerate up to $\overline{f}_N = 80$ N, while a clean stroke is guaranteed with $\underline{f}_N = 1$ N. The friction coefficient $\mu = 0.519$ has been experimentally estimated from force measurements. The torque and velocity limits at the manipulator's joints are reported in Table I.

As regards the planning process, the inequalities (25) are verified by evaluating the vectors (8)–(11) for each λ, employing the UR10's dynamic model identified in [16].

The admittance controller uses the parameters $\boldsymbol{M}_d = \text{diag}\{0.1, 0.1, 0.02\}$, $\boldsymbol{K}_D = \text{diag}\{300, 300, 1200\}$ and $\boldsymbol{K}_P = \text{diag}\{5500, 5500, 625\}$ so as to be compliant and damped along the contact axis and stiff and less damped along the motion/writing plane. The controller receives a force reference of $f_d = 20$ N in order to ensure a stable contact.

The path showed in Fig. 2 is time-parametrized using TOTP with DP. The grid dimensions are set to $N_\lambda = 500$ and $N_{\dot{\lambda}} = 5000$, with $\dot{\lambda} \in [0; 1.0]$. The Phase plane trajectory (PPT), as resulted from planning, is plotted (in blue) in Fig. 4, and compared with the PPT planned excluding interaction wrenches in the dynamic model (in orange). In TOTP, the trajectory tracking time is inversely proportional to the area beneath the PPT [11]. It is immediate to notice that the action of exerting wrenches on the blackboard has a global effect of lowering the PPT, resulting in a slower motion.

In terms of the solution to the optimization problem, the minimum of the cost function (i.e., the time, according to (32)) found by the algorithm described in Section IV is 4.51 seconds, while adopting the state-of-the-art DP algorithm gives an optimal time of 4.07 seconds, thus yielding a faster (and, if interaction does happen during the execution, practically unfeasible) trajectory. As the optimization problem remains the same, employing other compatible state-of-the-art optimization techniques should produce a similar time.
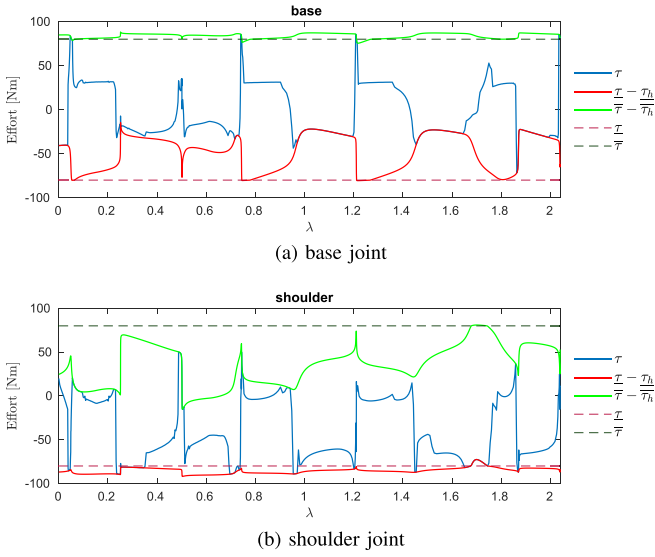
Fig. 5. Planned torques accounting for exerted wrenches and (modified) limits for first (a) and second (b) joint.

Fig. 6. Planned torques ignoring exerted wrenches and (modified) limits for first (a) and second (b) joint.

Despite the global effect, it is interesting to notice that, locally, for some λ, the PPT with interaction wrenches is higher than the contact-free one. In those regions, the manipulator is gaining dynamic capabilities in exerting wrenches on the environment. This fact is more easily analyzed in the torque domain.

The effect of (28) and (29) is to impose constraints on $\ddot{\lambda}$, in fact changing the torque limits $\underline{\tau}$ and $\overline{\tau}$ by path-dependent offsets $\underline{\tau}_h(\lambda)$ and $\overline{\tau}_h(\lambda)$. The plots in Fig. 5 prove that the planned torques do not break the modified limits and saturate over them when the joint velocities do not saturate. Yet, it is possible that the available torques are beyond $\overline{\tau}$ (or $\underline{\tau}$), with $\overline{\tau} - \overline{\tau}_h(\lambda)$ (or $\underline{\tau} - \underline{\tau}_h(\lambda)$) being less strict than the technological torque limits. For example, this is the case of hard acceleration/deceleration phases where the torque needed at joint j to exert the tangential force compensates the term $a_j(\lambda)\ddot{\lambda}$, indeed increasing the residual torque, to be employed for time minimization. Conversely, the most common effect is that the tangential force produces, at joint j, a torque that opposes to the accelerating/decelerating one, in fact limiting the dynamic capabilities of the manipulator. In Fig. 5, only base and shoulder joints are shown as they are the limiting joints for the "ACG" path. The other joints adapt to allow for trajectory tracking.

In contrast, Fig. 6 shows the torques planned without including external wrenches in the algorithm, thus proving the effectiveness of adopting the proposed framework. It is immediate to notice that, for some joint j, the planned torque can break the (modified) limit in certain portions of the path. For example, in Fig. 6(a), τ overcomes the lower bound $\underline{\tau} - \underline{\tau}_h$ for $\lambda \in [1; 1.2]$. This phenomenon has a critical practical outcome, causing a possible failure at execution. In fact, if the robot does exert wrenches, then the torques actually available for joint actuation would not be sufficient for performing the motion as it was planned, resulting in a path deviation.

The admittance controller, with the selected tuning, guarantees that the measured force is within bounds, as evident from
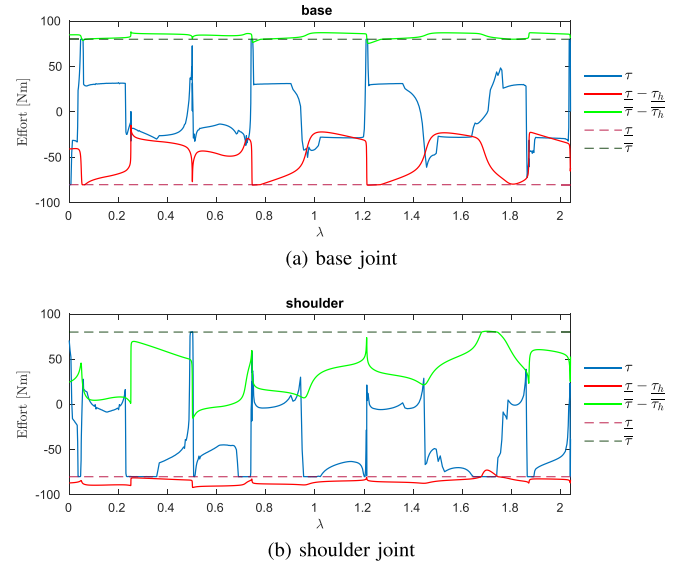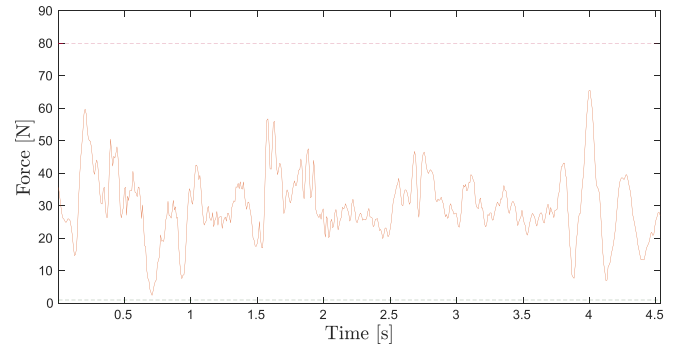

Fig. 7. Contact force along the normal direction (solid line) during the execution of the time-optimal planned trajectory and contact force limits (dashed lines).

Fig. 7, confirming that the execution respects the task specifications considered at planning level. As a consequence, the chalk does not get damaged, but a clear stroke can be delivered.

In order to stress the necessity of explicitly adding the wrench bounds in the planning process, it is worth recalling that only with this inclusion the robot's joint torques and velocities are guaranteed to respect the manipulator's physical capabilities, accounting for both motion and interaction efforts. On the contrary, if interaction wrenches are disregarded in planning, then the only solution for a safe task execution is to down-scale the PPT, therefore manually lowering the trajectory's speed. Clearly, this approach would yield the following drawbacks:
1) the planning process will no longer be *optimal* with respect to the actual manipulator's capabilities;
2) the time parametrization will no longer benefit from the less strict joint torque limits resulting from the interaction, as previously discussed in this section.

Fig. 8 provides a qualitative illustration of the trajectory tracking performance. The path is accurately tracked, considering the type of task (writing is effective as long as letters can be
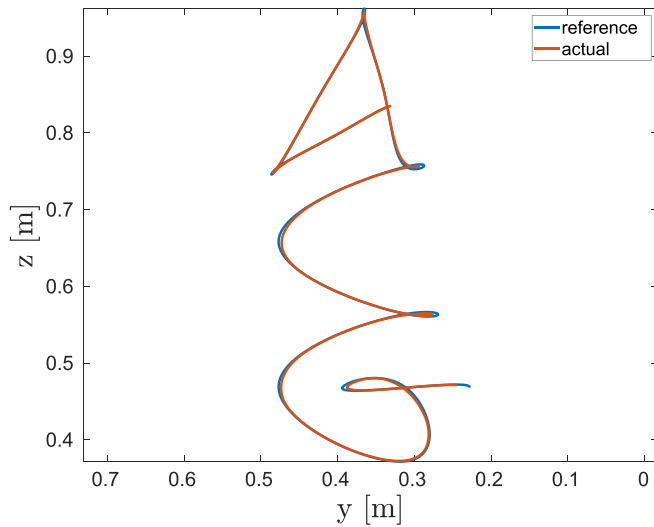
Fig. 8.   Comparison between reference and tracked paths after the trajectory execution.

distinguished from one another), with the higher deviations arising at the segments of highest curvature. This is also where the robot hardly accelerates and decelerates, while coping with external forces. It is anyhow worth remarking that the actual trajectory and force tracking performances depend on the controller adopted in the execution phase, as also noted in similar works [9]. For example, trajectory tracking performances can be improved by making the robot stiffer along the motion/writing plane, but improving the controller performance goes beyond the scope of this communication.

VI. CONCLUSION

This letter deals with optimal trajectory planning for robotic manipulators with interaction with an unmodeled environment.

The proposed strategy to consider interaction is imposing limits on the wrenches to exert on the environment, also accounting for uncertainties on their dynamic characterization. In this communication, for simplicity, limits are imposed on linear forces only, but the same approach can be easily extended to momenta.

This methodology is compatible with every implementation of the planning algorithm and additional joint space and task space constraints can be included without any modification, regardless of the performance index to optimize.

The validation experiment consists in a time-optimal planned trajectory executed on a real UR10 robot, employing an admittance controller to deal with the interaction at control level. Indeed, the performances are only limited by the capabilities of the controller to track forces and positions accurately, while moving at very high speeds. The inclusion of wrenches at planning level affects the trajectory tracking time: while it is

intuitive to expect that interaction slows motion down, in some cases, interaction can be beneficial to motion, indeed improving the dynamic capabilities of the robot.

Restricting the wrench limits requires the interaction controller to accurately track the wrench reference. It would be interesting, in future works, to assess the capabilities of a different control strategy, e.g., direct force control, to accurately track the force reference while coping with the high-speed dynamics of time-optimal motion. Also, controllers introduce their own dynamics which could be considered at planning level.

REFERENCES

[1] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.

[2] J.-W. Ma, S. Gao, H.-T. Yan, Q. Lv, and G.-Q. Hu, "A new approach to time-optimal trajectory planning with torque and jerk limits for robot," *Robot. Auton. Syst.*, vol. 140, Jun. 2021, Art. no. 103744.

[3] E. Ferrentino, A. D. Cioppa, A. Marcelli, and P. Chiacchio, "An evolutionary approach to time-optimal control of robotic manipulators," *J. Intell. Robotic. Syst.*, vol. 99, no. 2, pp. 245–260, Nov. 2019.

[4] D. Kaserer, H. Gattringer, and A. Müller, "Nearly optimal path following with jerk and torque rate limits using dynamic programming," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 521–528, Apr. 2019.

[5] S.-J. Kim, D.-S. Choi, and I.-J. Ha, "A comparison principle for state-constrained differential inequalities and its application to time-optimal control," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 967–983, Jul. 2005.

[6] J. Dong and J. A. Stori, "A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization," *J. Dyn. Syst. Meas. Control*, vol. 128, no. 2, pp. 379–390, Jul. 2005.

[7] K. Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Automat. Control*, vol. 31, no. 6, pp. 491–500, Jun. 1986.

[8] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," *J. Dyn. Syst. Meas. Control*, vol. 109, no. 2, pp. 88–96, Jun. 1987.

[9] D. Kaserer, H. Gattringer, and A. Müller, "Time optimal motion planning and admittance control for cooperative grasping," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2216–2223, Apr. 2020.

[10] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, Sep. 1985.

[11] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Control*, vol. 30, no. 6, pp. 531–541, Jun. 1985.

[12] C. Li, Z. Zhang, G. Xia, X. Xie, and Q. Zhu, "Efficient force control learning system for industrial robots based on variable impedance control," *Sensors*, vol. 18, no. 8, Aug. 2018, Art. no. 2539.

[13] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.

[14] D. Kaserer, H. Gattringer, and A. Müller, "Admittance control of a redundant industrial manipulator without using force/torque sensors," in *Proc. IEEE 42nd Annu. Conf. Ind. Electron. Soc.*, Firenze, Italy, 2016, pp. 5310–5315.

[15] A. Wahrburg, J. Bös, K. D. Listmann, F. Dai, B. Matthias, and H. Ding, "Motor-current-Based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 2, pp. 879–886, Apr. 2018.

[16] C. Gaz, E. Magrini, and A. De Luca, "A model-based residual approach for human-robot collaboration during manual polishing operations," *Mechatronics*, vol. 55, pp. 234–247, Nov. 2018.