Model Predictive Impedance Control

Maciej Bednarczyk, Hassan Omran, Bernard Bayle

Model Predictive Impedance Control

Maciej Bednarczyk[1], Hassan Omran[1] and Bernard Bayle[1]

***Abstract*— Robots are more and more often designed in order to perform tasks in synergy with human operators. In this context, a current research focus for collaborative robotics lies in the design of high-performance control solutions, which ensure security in spite of unmodeled external forces. The present work provides a method based on Model Predictive Control (MPC) to allow compliant behavior when interacting with an environment, while respecting practical robotic constraints. The study shows in particular how to define the impedance control problem as a MPC problem. The approach is validated with an experimental setup including a collaborative robot. The obtained results emphasize the ability of this control strategy to solve constraints like speed, energy or jerk limits, which have a direct impact on the operator's security during human-robot compliant interactions.**

***Index Terms*— Impedance control, collaborative robotics, physical human-robot interaction**

I. INTRODUCTION

The robotics community has been developing for many years now the idea that novel robotic systems should be designed and controlled to perform tasks in contact with their environment, and in particular with human operators. This trend has been increased by the development of collaborative robots. This new type of robot is designed to increase the physical capabilities of human operators, or limit their fatigue. A subsequent research effort has already been devoted to introduce collaborative robots in factories [1], [2]. One of the main challenges is to provide control laws allowing convenient and safe interactions between the human, the robot and the environment [3].

In robotics, when no interaction with the environment is needed, motion control strategies are used. At the contrary, in the presence of interactions, direct force control strategies are preferred for fine force tracking. However, they require good models of the interaction and of the environment, and are not compatible with unpredictable interactions [4]. For this reason, interaction control offers a compromise in order to deal with both force and motion during interactions. Impedance control (IC) [5] is a widely used interaction control method, particularly efficient for human-robot interactions. It consists in imposing an impedance model for the relationship between the manipulator and the environment. Since its introduction by Hogan, IC has attracted researchers' attention, leading to improvements in handling robots flexibility [6] or variable parameters [7], to mention just a few.

In addition to the interaction management, the robot controller should handle several constraints, such as speed and energy limits [8], limited jerk [9] or actuators saturation. These limits allow ensuring safety [10] as well as good ergonomics of the human-robot interaction [11]. However, it is not straightforward to respect these constraints using classical IC. Model Predictive Control (MPC) is an advanced control method to deal with multi-variable systems with multiple constraints. First proposed for process control in the petrochemical industry [12], MPC has successfully replaced classical methods in many applications [13]. By measuring the current state of the system, MPC predicts the evolution of the system variables over some horizon and computes the optimal input with respect to some performance index, while respecting imposed constraints.

MPC has been only recently used for interaction control. In [14], MPC is used as a high-level controller for path set-point generation, taking into account contact forces due to interactions with the environment. The authors of [15] investigate a direct force control strategy based on predictive path following using MPC. In [16], MPC is combined with admittance control for dealing with stability issues while interacting with a stiff environment. Even more recently, non-linear MPC is formulated by adding admittance dynamics into a path following problem [17]. However, in all these applications, the choice of the cost functions is not directly related to the desired compliant behavior.

To the best of our knowledge, no research has yet been done to combine IC and MPC advantages. Our contribution in this work is to propose an appropriate MPC formulation to obtain a prescribed compliant behavior, while respecting a set of practical robotic constraints. The paper is organized as follows. In section II, we introduce the concepts of system dynamics and IC, later used in the paper. Section III describes the design method for the proposed MPC controller. Section IV provides experimental results featuring a collaborative robot. Finally, conclusions and perspectives are given in Section V.

II. CONTROL BACKGROUND

A. Rigid Body Dynamics and Linearized Model

Consider the model of an n-joint serial robotic manipulator operating in an m-dimensional task space

$$M(q)\ddot{q} + C(\dot{q}, q) + G(q) = \tau_c - J(q)^T f_{ext} \qquad (1)$$

where $\ddot{q}, \dot{q}, q \in \mathbb{R}^n$ are joint accelerations, velocities and positions, respectively, $\tau_c \in \mathbb{R}^n$ are the commanded joint torques, $f_{ext} \in \mathbb{R}^m$ is the end-effector wrench, $J(q) \in \mathbb{R}^{m \times n}$ is the robot Jacobian matrix and $M(q) \in \mathbb{R}^{n \times n}$, $C(\dot{q}, q) \in \mathbb{R}^n$ and $G(q) \in \mathbb{R}^n$ are the inertia matrix and the Coriolis and gravity terms, respectively. A classical solution [4] to linearize equation (1) is to apply

[1]ICube, UMR 7357, University of Strasbourg, CNRS, 1 place de l'Hôpital, 67091, Strasbourg, France. m.bednarczyk@unistra.fr

the control law $\tau_c = M(q)\nu + C(\dot{q}, q) + G(q) + J(q)^T f_{ext}$ with ν the new control input, resulting in the double integrator model in joint space

$$\ddot{q} = \nu \tag{2}$$

The robot dynamics can then be expressed in task space by derivating the forward differential kinematic model, which leads to

$$\ddot{p} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} \tag{3}$$

where $p \in \mathbb{R}^m$, $\dot{J}(q, \dot{q}) \in \mathbb{R}^{m \times n}$ denote the robot end-effector pose and the time derivative of the Jacobian matrix. Using (2) and (3) one can express a new input u such that $\nu = J^+ u - J^+ \dot{J}\dot{q}$, with $J^+ \in \mathbb{R}^{n \times m}$ the pseudo-inverse of J, resulting in a double integrator model in task space

$$\ddot{p} = u \tag{4}$$

B. Impedance Control

In order to dynamically link the system positions, velocities and accelerations with the external forces, the input u of system (4) is computed in order to obtain

$$M_v(\ddot{p}_r - \ddot{p}) + D_v(\dot{p}_r - \dot{p}) + K_v(p_r - p) = f_{ext} \tag{5}$$

So, the system subject to the external forces f_{ext} is characterized by its impedance with an apparent virtual mass $M_v \in \mathbb{R}^{m \times m}$, a desired damping $D_v \in \mathbb{R}^{m \times m}$ and a desired stiffness $K_v \in \mathbb{R}^{m \times m}$, tracking a reference motion p_r. Note that M_v, D_v and K_v are symmetric positive definite matrices. The desired behavior (5) can be obtained by imposing

$$u = \ddot{p}_r + M_v^{-1}(D_v(\dot{p}_r - \dot{p}) + K_v(p_r - p) - f_{ext}) \tag{6}$$

to the linearized system (4).

IC can also be represented using a state space model. In the following, we will consider external forces with slow variations so that $\dot{f}_{ext} \approx 0$. This simplification corresponds to many practical cases but could seem restrictive in the case of unmodeled contact. However, if a better model of the external force is available, it can be included at this stage into the state-space model. This aspect of the problem goes far beyond the scope of this paper, justifying the previous assumption. With this assumption, the dynamic model of equation (4) can be written as

$$\dot{x} = A_c x + B_c u \tag{7}$$

with

$$x = \begin{bmatrix} \dot{p} \\ p \\ f_{ext} \end{bmatrix}, A_c = \begin{bmatrix} 0 & 0 & 0 \\ I_m & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} I_m \\ 0 \\ 0 \end{bmatrix}$$

The external force has been integrated into the state variable in order to better deal with this disturbance, inspired by the Internal Model Principle [18]. Note however that the objective is not to cancel the disturbance, but rather to have an appropriate response with respect to it.

Let us now write the IC using a state space formalism. To do so, note that (6) can be written as

$$u = u^r + K(x^r - x) \tag{8}$$

with

$$u^r = \ddot{p}_r \in \mathbb{R}^m \tag{9}$$

$$x^r = \begin{bmatrix} \dot{p}_r^T & p_r^T & 0 \end{bmatrix}^T \in \mathbb{R}^{3m} \tag{10}$$

$$K = M_v^{-1} \begin{bmatrix} D_v & K_v & I_m \end{bmatrix} \in \mathbb{R}^{m \times 3m} \tag{11}$$

where r is some reference vector that can be chosen according to the desired task. By substituting (8) into (7), the state space model of the impedance controlled system can be written as

$$\dot{x} = (A_c - B_c K)x + B_c(u^r + Kx^r) \tag{12}$$

IC can therefore be seen as a state-feedback tracking problem.

Since the controller is implemented numerically, it is necessary to find a discrete model of the system. The discretization of equation (7) with zero order hold yields

$$x_{k+1} = Ax_k + Bu_k \tag{13}$$

where k represents the current step, x_k is the discrete state at step k and A, B and C are the discrete state space matrices. The sampling period T_s is selected short enough to emulate the continuous controller (8), which gives

$$u_k = u_k^r - K(x_k - x_k^r) \tag{14}$$

with x_k^r and u_k^r the discrete forms of x^r and u^r.

III. Model Predictive Impedance Control

MPC consists in solving at every step a finite-horizon optimal control problem with constraints. In order to deal at the same time with unpredictable interactions as does the IC, and different types of limits as does the MPC, a novel controller, called Model Predictive Impedance Controller (MPIC) is proposed.

When no constraints are active, the MPIC should be equivalent to IC, i.e. to the static state feedback control (14). Several methods have been proposed in the literature to make a MPC behave as a state feedback such as the LMI-based inverse optimality method [19], [20] or the controller matching based on QP matrices as in [21]. However, these methods either do not guarantee the exact matching between the MPC and the desired state feedback controller, or do not allow online computation. To obtain the controller of equation (14), the cost function $\mathcal{J}$ of the MPC problem has to be designed such that its optimum is zero when the desired behavior is obtained. To do so, a zero-value cost function over a control horizon H can be written, as in [19].

$$\mathcal{J} = \sum_{k=0}^{H-1} (u_k + K(x_k - x_k^r) - u_k^r)^T R(u_k + K(x_k - x_k^r) - u_k^r) \tag{15}$$

which is optimal for the desired control law, for any $R = R^T > 0 \in \mathbb{R}^{m\times m}$. Furthermore, $\mathcal{J}$ can be written as

$$\mathcal{J} = \sum_{k=0}^{H-1} \begin{bmatrix} u_k - u_k^r \\ x_k - x_k^r \end{bmatrix}^T \begin{bmatrix} R & S^T \\ S & Q \end{bmatrix} \begin{bmatrix} u_k - u_k^r \\ x_k - x_k^r \end{bmatrix} \tag{16}$$

where $Q = K^T R K \in \mathbb{R}^{3m\times 3m}$ and $S = K^T R \in \mathbb{R}^{3m\times m}$ are the weights of the particular terms satisfying the conditions $Q = Q^T \geq 0$ and $Q - SR^{-1}S^T \geq 0$, and H is the control horizon. In [22] it is pointed out that the cross-terms between input and state in the quadratic cost function allow reproducing any state-feedback control strategy by making this equivalence as the primary control objective.

With the discrete state-space plant dynamics (13), a MPC problem can be written as the tracking of some references, with feedback x_k^r and u_k^r terms

$$\min_{u_k} \mathcal{J} = \min_{u_k} \sum_{k=0}^{H-1} \begin{bmatrix} u_k - u_k^r \\ x_k - x_k^r \end{bmatrix}^T \begin{bmatrix} R & S^T \\ S & Q \end{bmatrix} \begin{bmatrix} u_k - u_k^r \\ x_k - x_k^r \end{bmatrix} \tag{17}$$
$$s.t. \begin{cases} x_{k+1} &= Ax_k + Bu_k \\ x_0 &= \text{measured current state} \end{cases}$$

where x_0 is the initial state measured at each control cycle. In the following, the input rate of change $\Delta u_k = u_k - u_{k-1}$ is used instead of the input u_k, a choice often made for tracking problems. First, let us define

$$\begin{aligned} \boldsymbol{\Delta u} &= [\Delta u_0^T \ \dots \ \Delta u_{H-1}^T]^T \in \mathbb{R}^{mH} \\ \boldsymbol{u} &= [u_{-1}^T \ u_0^T \ \dots \ u_{H-1}^T]^T \in \mathbb{R}^{m(H+1)} \\ \boldsymbol{x} &= [x_0^T \ \dots \ x_{H-1}^T]^T \in \mathbb{R}^{3mH} \\ \boldsymbol{\rho} &= [u_0^{rT} \ \dots \ u_{H-1}^{r\,T} \ \ x_0^{rT} \ \dots \ x_{H-1}^{r\,T}]^T \in \mathbb{R}^{4mH} \end{aligned} \tag{18}$$

the vectors containing all the input rates of change, inputs, states and references over the control horizon H. Notation u_{-1} represents the last applied input. The minimization problem (17) can be transformed into a Quadratic Program (QP) form. To do so, it has first to be expressed in matrix form such that

$$\min_{\boldsymbol{\Delta u}} \mathcal{J} = \min_{\boldsymbol{\Delta u}} \begin{bmatrix} \boldsymbol{\Delta u} \\ \boldsymbol{x} \\ \boldsymbol{u} \\ \boldsymbol{\rho} \end{bmatrix}^T \boldsymbol{\Phi}^T \begin{bmatrix} \boldsymbol{\mathcal{R}} & \boldsymbol{\mathcal{S}}^T \\ \boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{Q}} \end{bmatrix} \boldsymbol{\Phi} \begin{bmatrix} \boldsymbol{\Delta u} \\ \boldsymbol{x} \\ \boldsymbol{u} \\ \boldsymbol{\rho} \end{bmatrix} \tag{19}$$

$$s.t. \begin{cases} \boldsymbol{x} = & \boldsymbol{\mathcal{A}} x_0 + \boldsymbol{\mathcal{C}} \boldsymbol{u} \\ \boldsymbol{u} = & \boldsymbol{\mathcal{T}} \boldsymbol{\Delta u} + \boldsymbol{\mathcal{I}} u_{-1} \end{cases} \tag{20}$$

with $\boldsymbol{u}$, $\boldsymbol{\Delta u}$, $\boldsymbol{x}$ and $\boldsymbol{\rho}$ defined by (18) and

$$\begin{aligned} \boldsymbol{\mathcal{Q}} &= diag(Q, \ \dots \ , \ Q) \in \mathbb{R}^{3mH\times 3mH} \\ \boldsymbol{\mathcal{R}} &= diag(R, \ \dots \ , \ R) \in \mathbb{R}^{mH\times mH} \\ \boldsymbol{\mathcal{S}} &= diag(S, \ \dots \ , \ S) \in \mathbb{R}^{3mH\times mH} \\ \boldsymbol{\mathcal{I}} &= [I_m \ \dots \ I_m]^T \in \mathbb{R}^{m(H+1)\times m} \end{aligned}$$

and block matrices

$$\boldsymbol{\mathcal{T}} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ I_m & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ I_m & \dots & \ddots & 0 \\ I_m & \dots & \dots & I_m \end{bmatrix} \in \mathbb{R}^{m(H+1)\times mH}$$

$$\boldsymbol{\Phi} = \left[\begin{array}{c|c|cc|cc} I_{mH} & 0 & I_{mH} & 0 & -I_{mH} & 0 \\ 0 & I_{3mH} & 0 & 0 & 0 & -I_{3mH} \end{array}\right] \in \mathbb{R}^{4mH\times m(5H+1)}$$

$\boldsymbol{\mathcal{C}} \in \mathbb{R}^{3mH\times m(H+1)}$ is the H-steps state reachability matrix and $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{3mH\times 3m}$ the H-steps free evolution matrix [21], such that

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & B & 0 & \dots & 0 \\ 0 & AB & B & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & A^{H-2}B & A^{H-3}B & \dots & B \end{bmatrix} \quad \boldsymbol{\mathcal{A}} = \begin{bmatrix} I_{3m} \\ A \\ A^2 \\ \vdots \\ A^{H-1} \end{bmatrix}$$

Problem (19) is a constrained QP that could be solved numerically. However, previously, a simplification of the problem can be performed by including the constraints (20) into the cost function, so that

$$\mathcal{J} = \begin{bmatrix} \boldsymbol{\Delta u} \\ x_0 \\ u_{-1} \\ \boldsymbol{\rho} \end{bmatrix}^T \boldsymbol{\Psi}^T \boldsymbol{\Phi}^T \begin{bmatrix} \boldsymbol{\mathcal{R}} & \boldsymbol{\mathcal{S}}^T \\ \boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{Q}} \end{bmatrix} \boldsymbol{\Phi} \boldsymbol{\Psi} \begin{bmatrix} \boldsymbol{\Delta u} \\ x_0 \\ u_{-1} \\ \boldsymbol{\rho} \end{bmatrix} \tag{21}$$

with

$$\boldsymbol{\Psi} = \begin{bmatrix} I_{mH} & 0 & 0 & 0 \\ \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{T}} & \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{I}} & 0 \\ \boldsymbol{\mathcal{T}} & 0 & \boldsymbol{\mathcal{I}} & 0 \\ 0 & 0 & 0 & I_{4mH} \end{bmatrix} \in \mathbb{R}^{m(5H+1)\times m(5H+4)}$$

By defining matrices $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{mH\times mH}$, $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{mH\times m(5H+4)}$ and $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{m(5H+4)\times m(5H+4)}$ such that

$$\begin{bmatrix} \boldsymbol{\mathcal{H}} & \boldsymbol{\mathcal{F}} \\ \boldsymbol{\mathcal{F}}^T & \boldsymbol{\mathcal{L}} \end{bmatrix} = \boldsymbol{\Psi}^T \boldsymbol{\Phi}^T \begin{bmatrix} \boldsymbol{\mathcal{R}} & \boldsymbol{\mathcal{S}}^T \\ \boldsymbol{\mathcal{S}} & \boldsymbol{\mathcal{Q}} \end{bmatrix} \boldsymbol{\Phi} \boldsymbol{\Psi} \tag{22}$$

the optimization problem (19) becomes

$$\min_{\boldsymbol{\Delta u}} \mathcal{J} = \min_{\boldsymbol{\Delta u}} \boldsymbol{\Delta u}^T \boldsymbol{\mathcal{H}} \boldsymbol{\Delta u} + 2[x_0^T \ \ u_{-1}^T \ \ \boldsymbol{\rho}^T] \boldsymbol{\mathcal{F}}^T \boldsymbol{\Delta u} \tag{23}$$

where the term corresponding to $\boldsymbol{\mathcal{L}}$ has been removed as it is constant. Because of the hypotheses on the weight matrices, $\boldsymbol{\mathcal{H}} > 0$ and the equation (23) is a convex QP. Note here, that as the problem is convex the solution corresponding to the zero cost function is the unique global solution. This means, that when unconstrained, the optimal input given by the MPIC controller will behave as the desired state-feedback. Without constraints, (23) has the following analytic solution

$$\boldsymbol{\Delta u} = -\boldsymbol{\mathcal{H}}^{-1} \boldsymbol{\mathcal{F}} \begin{bmatrix} x_0^T & u_{-1}^T & \boldsymbol{\rho}^T \end{bmatrix}^T \tag{24}$$

In MPC, only the first input is applied at each iteration, i.e. $u_0 = u_{-1} - \begin{bmatrix} I_m & 0 & \dots & 0 \end{bmatrix} \boldsymbol{\Delta u}$. Note that this applies only when there are no other constraints than the system model. However, in most practical cases, additional constraints have to be considered. In this case, no analytical solution exists and a numerical solver has to be used. In order to do so, the constraints need to be written in the standard form

$$\boldsymbol{\Gamma} \boldsymbol{\Delta u} \leq \boldsymbol{\Pi} + \boldsymbol{\Omega} \begin{bmatrix} x_0^T & u_{-1}^T \end{bmatrix}^T \tag{25}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{c\times mH}$, $\boldsymbol{\Omega} \in \mathbb{R}^{c\times 4m}$ and $\boldsymbol{\Pi} \in \mathbb{R}^c$ are formulated based on c problem constraints.

IV. Evaluation

A. Experimental setup

The performance of the proposed controller was evaluated on an experimental setup featuring a collaborative robot following a reference trajectory. We chose to perform the experiments on a simple exemplary environment in order to ensure good repeatability of the task, and so to provide comparable results. For this reason, no interaction with the user occurs during the experiments presented in the paper. However, human-robot interactions are shown in the accompanying video. The reference was a smooth polynomial trajectory on a planar surface, placed slightly under the table in order to ensure contact of the end-effector with the (x,y)-plane. In addition, a rigid obstacle was placed on the path generating unmodeled disturbances. The experimental setup is shown in Fig. 1.

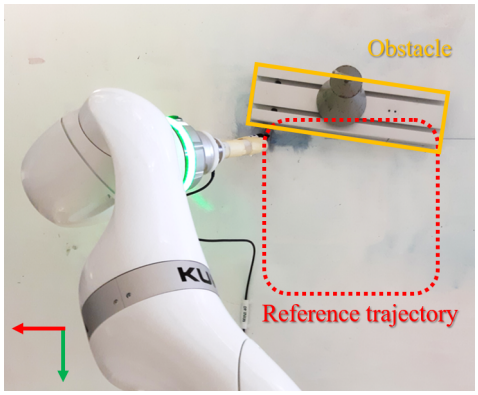


Fig. 1. Experimental setup with reference trajectory and obstacle.

Both IC and MPIC were compared, throughout four different experiments. At first, we show the equivalence of IC and MPIC when no constrains are applied. Then, position, velocity and input constraints are successively imposed to the MPIC controller for validation of its performances, and to illustrate the advantages it offers.

B. Implementation and hardware specifications

The experiments were carried out with a KUKA *iiwa* 14 collaborative robot. An ATI-Mini40 force-torque sensor was mounted on the robot to measure the end-effector wrench. The proposed controller addresses the robot at joint torque level, which is not a classical approach for controlling this type of manipulators. As the internal structure of the low level controllers of the KUKA *iiwa* are not well known, with some identification attempts such as in [23], [24], we based our understanding of the low level control architecture on [25]. The robot software allows the tuning of joint impedance parameters, which were set to zero and the torque commands generated by the MPIC were fed to the system using KUKA's FRI (Fast Robot Interface) protocol. This protocol was also used to measure the joint positions, whereas the joint velocities were computed using a filtered derivative with a cut-off frequency at 50Hz. End-effector positions and velocities were computed using the forward kinematic and forward differential kinematic models, respectively. The control and the FRI communication were sampled at 500Hz. For solving the QP problem (23) with constraints, the C++ solver *qpOASES* [26] was used. The impedance parameters were chosen such that $M_v = diag(5,\ 5,\ 5,\ 2,\ 2,\ 2)$, $K_v = diag(300,\ 300,\ 50,\ 500,\ 500,\ 500)$, and $D_v = diag(77.5,\ 77.5,\ 31.6,\ 63.2,\ 63.2,\ 63.2)$. K_v was chosen in order to ensure: 1) a good contact in the direction normal to the (x,y)-plane, 2) a relatively low stiffness in the x and y directions, which represents a trade-off in compliance and tracking performance, and 3) a high stiffness in orientation in order to have a nearly constant wrist pose. D_v was selected in order to have a damping ratio of 1 for the impedance model. The prediction horizon for the MPIC was set to $H = 5$, as currently the solver struggles to solve the constrained QP problem for a longer horizon in the given time. As the input given by the MPIC when operating close to a constraint can be noisy, a fixed limit on the input rate of change $-0.1 \leq \Delta u \leq 0.1$ was set.

C. Experimental results

1) Comparing IC and MPIC without constraints: The IC law (6) was implemented on the system and tested in the same conditions as the MPIC. Fig. 2 allows comparing the two controllers. At position level (Fig. 2a) one can observe the almost perfect matching of the two trajectories. The position error between IC and MPIC has been evaluated to $\bar{e}_x = 0.24\text{mm}$ and $\bar{e}_y = 0.12\text{mm}$ with a standard deviation $\sigma_x = 1.3\text{mm}$ and $\sigma_y = 0.7\text{mm}$. This match can also be noticed at velocity (Fig. 2b) and acceleration (Fig. 2c) levels where only a small difference on the x-axis can be spotted.

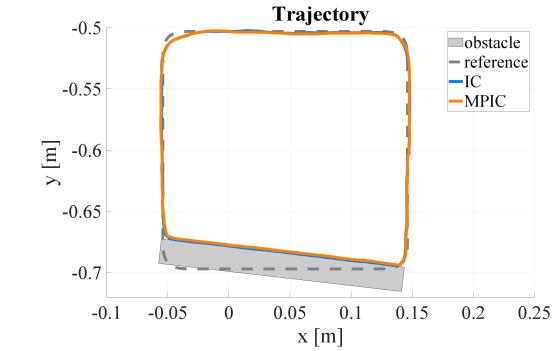
We believe that this difference is caused by some friction on the surface. One can also notice that both controllers do not perfectly match the reference trajectory. The main reason lies in the fact that the impedance model has a rather low stiffness, which impacts reference tracking performance. However, at some points, as it is the case in the left upper corner of the trajectory in Fig. 2a, the performance of both controllers seems to be particularly reduced, even though they behave identically. We believe that this can be caused by the internal controller of the KUKA robot that has some issues tracking torque commands as pointed out in [23], [24].

We were interested in verifying the impact of these small defects due to the (not open) low level controllers of KUKA *iiwa*. To do so, we identified the real impedance parameters estimated from experimental data, and compared them to the ones chosen in the impedance model. A linear least square minimization was performed on the experimental data of IC to retrieve these parameters. The resulting estimations are given in Table I.

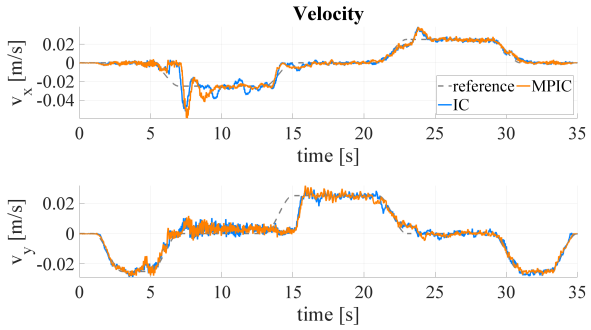
TABLE I

Impedance parameters retrieved from data.

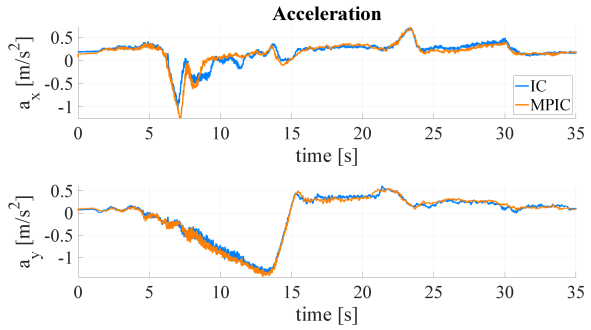
	M_v	K_v	D_v
x	5.40	290.04	76.72
y	5.39	280.27	72.80



(a) End-effector trajectories under IC and MPIC.



(b) End-effector velocities under IC and MPIC.



(c) End-effector accelerations under IC and MPIC.

Fig. 2. Experimental comparison of the trajectories, velocities and accelerations of the robotic manipulator end-effector, following a reference trajectory. Case 1: IC and MPIC without active constraints and with an obstacle on the path.

It turns out that the system impedance during the experiments fits rather well the imposed impedance model, with a 8% max error on M_v, a 6.6% max error on K_v, and a 6.1% max error on D_v, for the x and y components. This result shows that the uncertainties on the internal controller do not heavily impact the performance and do not interfere with the robot-environment interaction. Typically, a human operator will not be able to discriminate such small differences in the inertia, damping and stiffness properties [27].

2) Position constrained MPIC: In this experiment, position constraints were specified to the MPIC. Fig. 3 shows the resulting end-effector trajectories as well as the position constraint, which was set such that $p_x \geq -0.006\text{m}$. One can observe that the two controllers have the same behavior when working in the allowed task space. When approaching the limit, MPIC stops behaving as IC to meet the constraint.

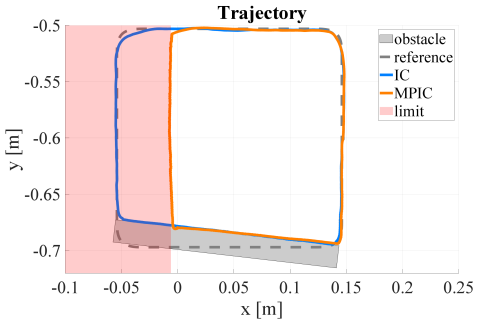
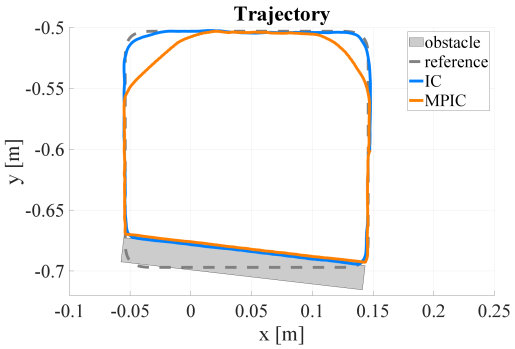
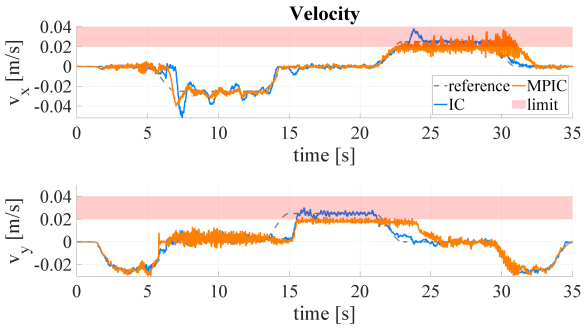


Fig. 3. End-effector trajectories. Case 2: IC and MPIC with position constraints ($x \geq -0.006\text{m}$) and with an obstacle on the path.

3) Velocity constrained MPIC: We show next, that it is also possible to apply velocity constraints with the MPIC controller. Fig. 4 compares the behavior of MPIC and IC at position and velocity level. In Fig. 4b, one can see that given velocity constraints $v_x \leq 0.02\text{m/s}$, $v_y \leq 0.02\text{m/s}$, the MPIC is able to limit the system velocity below the desired threshold. As in the previous paragraph, this also results in the modification of the desired trajectory, as shown Fig. 4a.



(a) End-effector trajectories under IC and MPIC.

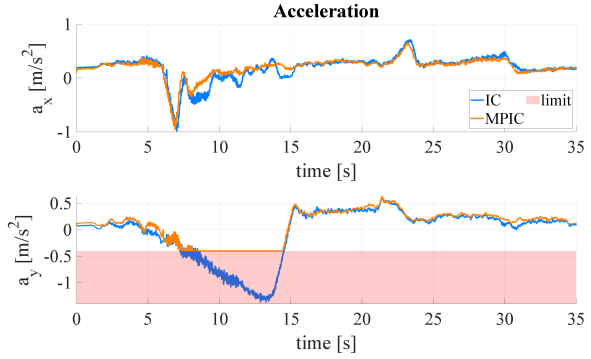


(b) End-effector velocities under IC and MPIC.

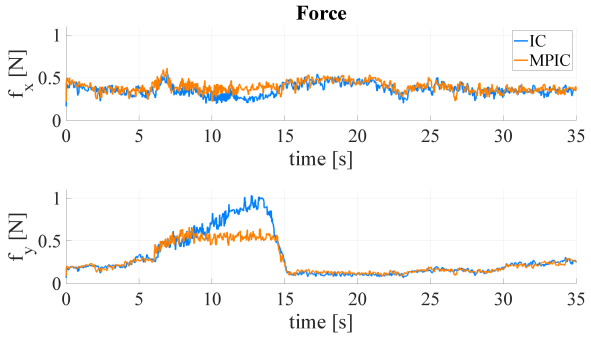
Fig. 4. Experimental comparison of the trajectories and velocities of the robotic manipulator end-effector following a reference trajectory. Case 3: IC and MPIC with active velocity constraints ($v_x \leq 0.02\text{m/s}$, $v_y \leq 0.02\text{m/s}$) and with an obstacle on the path.

4) Input constrained MPIC: Finally, the MPIC was tested in the case of input constraints resulting in the limitation of the commanded acceleration of the system. Fig. 5a allows comparing the accelerations obtained via IC and MPIC for a

constraint $a_y \geq -0.4\mathrm{m/s}^2$. As observed previously, both controllers have an equivalent behavior when inside the allowed acceleration range and MPIC limits acceleration when exceeding the limit. It is worth noticing here that limiting the acceleration will affect the contact force between the robot and the obstacle, as shown in Fig. 5b. This feature is of particular interest as it acts as a saturation that can limit the input to some maximally allowed threshold. In addition, it also acts directly on the system maximum acceleration and the resulting force applied by the robot, which is an important feature for interaction tasks.



(a) End-effector accelerations under IC and MPIC.



(b) Force measured at the end-effector under IC and MPIC.

Fig. 5. Experimental comparison of acceleration and measured force applied by the robotic manipulator end-effector. Case 4: IC and MPIC control with acceleration constraints ($a_y \geq -0.4\mathrm{m/s}^2$) and with an obstacle on the path.

V. Conclusion

In this paper, we have proposed a method to synthesize an impedance controller (IC) as a model predictive controller (MPC), based on the state-space representation of IC. This new controller is called Model Predictive Impedance Controller (MPIC). We have shown that MPIC combines the features of IC with the advantages of MPC, allowing to impose constraints on different system parameters, such as state or input. The controller has been tested with an experimental setup featuring a collaborative robot, and compared to IC. MPIC behaves exactly as IC when no constraints are active and it respects position, velocity and acceleration constraints when such constraints are imposed.

These results open the way to many possible applications, in particular to ensure safety in critical applications of collaborative robotics. Future work will tend towards including a human operator into the interaction dynamics to take full advantage of the prediction capability of MPIC to anticipate interactions. Another enhancement will be to improve the current setup, so that MPIC might be capable of adapting the impedance parameters online during the execution of the task. In addition, a theoretical stability analysis will be investigated in order to specify possible instability issues of MPIC, as currently only local stability can be guaranteed. One way to do so would be to impose some additional constraints that limit the system global energy, or to mathematically prove that when operating close to boundaries the system remains stable.

Acknowledgment

This work was supported by the *Investissements d'Avenir* program of the French Government, more specifically by Robotex ANR-10-EQPX-44, Labex CAMI ANR-11-LABX-0004 and the IDEX of the University of Strasbourg.

References

[1] M. Goodrich and A. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[2] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, Jun 2018.

[3] S. Buerger and N. Hogan, "Complementary stability and loop shaping for improved human-robot interaction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 232–244, Apr 2007.

[4] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer International Publishing, 2016.

[5] N. Hogan, "Impedance control: An approach to manipulation," *Journal of Dynamlc Systems, Measurement, and Contral*, vol. 107, pp. 1–7, Mar 1985.

[6] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*, ser. Springer tracts in advanced robotics. Springer, 2008.

[7] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, Oct 2016.

[8] P. Lasota, T. Fong, and J. Shah, "A survey of methods for safe human-robot interaction," *Foundations and Trends in Robotics*, vol. 5, no. 3, pp. 261–349, 2017.

[9] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 42–52, Feb 2003.

[10] S. Haddadin, *Physical Safety in Robotics*. Springer Fachmedien Wiesbaden, 2015, pp. 249–271.

[11] J. Kim, W.and Lee, L. Peternel, N. Tsagarakis, and A. Ajoudani, "Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 68–75, Jan 2018.

[12] J. Richalet, A. Rault, J. Testud, and J. Papon, "Algorithmic control of industrial processes," *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, vol. 10, pp. 1119–1167, 01 1976.

[13] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul 2003.

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

[15] 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, Jul 2017.

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

[17] K. J. Kazim, J. Bethge, J. Matschek, and R. Findeisen, "Combined predictive path following and admittance control," in *2018 Annual American Control Conference (ACC)*, Jun 2018, pp. 3153–3158.

[18] B. Francis and W. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, Sep 1976.

[19] E. N. Hartley and J. M. Maciejowski, "Designing output-feedback predictive controllers by reverse-engineering existing lti controllers," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2934–2939, Nov 2013.

[20] M. Priess, R. Conway, J. Choi, J. Popovich, and C. Radcliffe, "Solutions to the Inverse LQR Problem With Application to Biological Systems Analysis," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 770–777, Mar. 2015.

[21] S. Di Cairano and A. Bemporad, "Model predictive control tuning by controller matching," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 185–190, Jan 2010.

[22] E. Kreindler and A. Jameson, "Optimality of linear control systems," *IEEE Transactions on Automatic Control*, vol. 17, no. 3, pp. 349–351, June 1972.

[23] V. Chawda and G. Niemeyer, "Toward controlling a kuka lbr iiwa for interactive tracking," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1808–1814.

[24] ——, "Toward torque control of a kuka lbr iiwa for physical human-robot interaction," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017, pp. 6387–6392.

[25] A. Albu-Schffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *The International Journal of Robotics Research*, vol. 26, no. 1, p. 2339, Jan 2007.

[26] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpoases: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, Dec 2014.

[27] L. A. Jones, "Kinesthetic sensing," in *in Human and Machine Haptics*. MIT Press, 2000.