Hybrid control of manipulators in human-robot coexistence scenarios

Andrea Casalino, Paolo Rocco, Maria Prandini

Abstract— **This paper addresses safe motion planning for a manipulator, in environments shared with human operators. We formulate the problem as an optimal control problem for a hybrid system integrating three different operation modes: nominal, soft safety, and hard safety. The manipulator is assigned a nominal trajectory to reach a target position. If no human is present, then, the manipulator tracks the nominal trajectory; if a human enters its workspace, it tries to avoid it but without adopting too sharp and abrupt actions, except when strictly needed for safety. The decision on when and to what mode to commute is taken online, via a model predictive control approach involving a constrained optimization program with binary variables setting the active/nonactive status of the operating modes. The resulting control input is applied in a receding horizon fashion and the nominal trajectory towards the target is re-computed based on the current state of the system. The proposed approach is applied to a realistic simulation environment and appears computationally feasible and promising.**

I. Introduction

Modern industrial production plants are characterized more and more by the presence of robots sharing their workspace with humans: the most repetitive tasks are assigned to robots and the high cognitive ones to appropriately trained human operators. Since industrial robots are able to move at a high speed and they could severely injure humans [1], a significant research effort was fostered aiming at the development of safe motion control design methodologies when considering coexistence scenarios with humans, see e.g. [2], [3], or [4] for industrial robotics applications, while [5] refer to service robotics scenarios.

We shall consider a manipulator whose task is reaching some target configuration. If no human operator is within its workspace, then, the manipulator can move following a nominal point-to-point trajectory, which is computed ignoring the presence of humans so as to be optimal for the assigned task. When a human gets too close to the manipulator, some action has to be taken. An option is to interrupt the task execution and slow down/arrest the manipulator [2]. Alternatively, a corrective trajectory can be planned online with the aim of dodge the human and, at the same time, keep driving the manipulator to the desired target position [3]. Given the hardly predictable nature of the phenomenon, and in order to avoid an unnecessary reduction of productivity, one can adopt a control strategy where, at each time step, the presence of humans is monitored and a control action is designed so as to compromise between tracking a nominal optimal trajectory (performance) and avoiding humans (safety). This is actually the approach that we adopt in the present paper.

More specifically, we formulate the safe motion planning problem as an optimal control problem for a hybrid system that can commute among three different operation modes: nominal, soft safety, and hard safety. In the nominal operation mode, the manipulator follows a nominal trajectory to reach the target position, neglecting the presence of humans. Commercial robots are typically provided with an algorithm for trajectory generation, and we shall use this algorithm for the nominal operation mode. In the soft safety operation mode, the manipulator tries to avoid humans but without adopting too sharp and abrupt actions, which are taken in the hard safety mode but only if needed. We design the hard and soft safety mode controller by integrating the approaches presented in [2], [3], and [6].

Optimization is used to decide online when and to what mode to commute, according to the Model Predictive Control (MPC) strategy where, at each time step, one looks at some short term prediction horizon, solves a constrained optimization problem to decide the operation mode and the value to possibly assign to the hard control input, and applies the resulting control action. Optimization is repeated at the next time step over a moving time-window, till the desired target is reached. The nominal trajectory towards the target is re-computed online, according to the current state of the system. Within the MPC scheme, human occupancy is predicted at each time step. The MPC prediction horizon length should then compromise between the contrasting objectives of 1) allowing for a reliable human occupancy prediction and the online solution of the resulting constrained optimization problem, and 2) avoiding myopic solutions as in alternative methods like [3] that looks only one-step ahead for human avoidance. A different look-ahead time horizon is adopted for the nominal trajectory design and the MPC control problem. This is to avoid that the shorter look-ahead horizon used in MPC negatively affects performance, leading to greedy, myopic choices with respect to the goal of reaching the target position.

Hybrid systems represent a powerful modeling framework to describe complex systems where both discrete and continuous input/state variables are present. Various models have been introduced in the literature and we adopt here the computational oriented one in [7], which is called Mixed Logic Dynamic (MLD). MLD systems are characterized by piecewise affine continuous dynamics and described via affine equations and inequalities involving both continuous and discrete variables. This makes problems that can be

This work is partially supported by the European Commission under the project UnCoVerCPS, grant number 643921.

The authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza L. Da Vinci 32, 20133, Milano, Italy (e-mail: name.surname@polimi.it).

rephrased as optimization programs involving linear objective functions and constraints easy to solve via mixed integer linear programming tools [8]. Indeed, many problems have been addressed for MLD systems, including identification [9], model reduction, stability analysis, stabilization, and control design [10]. Other modeling framework and approaches available in the literature, like for instance [11], offer the possibility to include nonlinear continuous dynamics but are computationally more intensive, which poses a challenge for real-time applications.

The rest of this paper is organized as follows: Section II describes the proposed approach for motion planning of a manipulator in a human-robot coexistence scenario. In particular, the first part of Section II explains the adopted model and how safety specifications are formulated, while the second part presents the proposed hybrid control design method for safe motion planning via optimization. Section III provides a numerical experiment to show the efficacy of the approach. Finally, Section IV draws some concluding remarks.

II. Proposed Approach

In this section, we introduce an approach to safe motion planning for a manipulator that is assigned the task of reaching some target configuration in an environment shared with human operators. Section II-A presents the kinematic model of the manipulator as well as the constraints that model the safety specifications, while Section II-B introduced the proposed hybrid control strategy.

A. Kinematic model and safety specifications

Industrial manipulators can be seen as composed of a series of rotating joints. Motion control acts at a kinematic level by assigning the values of the joint accelerations $\boldsymbol{u}$ (which are control variables), with a certain sample time interval T_s. The state of the system is composed of the joint angular displacements of the manipulator, which are denoted as $\boldsymbol{q}$, and their derivatives $\dot{\boldsymbol{q}}$. The equations governing the state evolution are then given by the discrete-time double integrator:

$$\begin{cases} \dot{\boldsymbol{q}}_{k+1} = \dot{\boldsymbol{q}}_k + T_s \boldsymbol{u}_k \\ \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + T_s \dot{\boldsymbol{q}}_k + 0.5 T_s^2 \boldsymbol{u}_k \end{cases} \tag{1}$$

Safety specifications adopted for motion planning mainly follows from the approach presented in [2], where the safety constraint is expressed in terms of the joint speeds to avoid collisions between a manipulator and some fixed obstacle. The basic idea is to impose that the manipulator trajectory satisfied the condition that the distance traveled when braking is smaller than or equal to the distance from the obstacle. If the acceleration is constant over the time frame $[kT_s, (k+1)T_s)$, then, one can determine the displacements of every point in the kinematic chain of the manipulator as the integral with respect to time of the velocity, and then impose the constraint[1]

$$E(\dot{\boldsymbol{q}}_k + \dot{\boldsymbol{q}}_{k+1}) \leq H, \tag{2}$$

where E and H are functions of the configuration $\boldsymbol{q}_k$ and are taken as constant over a short time period. This assumption is satisfied in practice when the manipulator follows a smooth trajectory since in that case the variation of the geometrical Jacobian of the manipulator is negligible. We here extend the constraint expressed by equation (2), to impose collision avoidance during the entire horizon $[kT_s, (k+N)T_s)$, as follows:

$$E \cdot (\dot{\boldsymbol{q}}_k + 2\sum_{i=1}^{h-1} \dot{\boldsymbol{q}}_{k+i} + \dot{\boldsymbol{q}}_{k+h}) \leq H,\ h = 1, 2, \ldots, N. \tag{3}$$

Note that we have assumed here that matrices E and H keeps constant over $[kT_s, (k+N)T_s)$ and equal to the value associated with $\boldsymbol{q}_k$. This assumption is valid if its overall length NT_s is small.

To tackle the avoidance of moving obstacles, like humans, [3] has introduced the concept of swept volumes. The approach in [3] is model-based, in that a kinematic model is adopted for the human operator, who is seen as a manipulator whose motion is not controllable, but can be predicted. Indeed we can suppose to have sensors perceiving the scene and tracking the motion of human operators. It is then possible (see [3] for more details) to estimate the position and velocities of the human articulations and, under the assumption of bounded accelerations, compute for every anatomical part (limbs, head, etc.) the so-called swept volume, i.e., a shape containing all of the possible future reachable positions within a certain prediction horizon. Swept volumes represent the fixed obstacles to avoid when ensuring safety constraint expressed in (3). In [3], swept volumes were adopted to design the proper dodging maneuver by solving an optimization problem. Here, we adopt a similar approach but, differently from [3], we adopt a look-ahead time horizon that is larger than one single step, according to an MPC approach. Our goal is to design a switching scheme to control the manipulator and guarantee safety while not deteriorating excessively its productivity. The proposed switching control strategy is described in Section II-B. The extension of the safety constraint expressed in (3) jointly with the design of a switching scheme based on an MPC approach represent the main contribution of this work.

B. Hybrid control strategy for safe motion planning

In this section, we shall start describing three kinds of acceleration input for the system (1), i.e.,

- a nominal acceleration $\tilde{\boldsymbol{u}}$
- a soft corrective acceleration $\boldsymbol{u_{Dodge}}$
- a hard corrective acceleration $\boldsymbol{f}$

that are associated with the three operation modes: normal, soft safety, and hard safety. The first two kinds of accelerations are designed via a trajectory generation algorithm,

[1] The formulation has been adapted in comparison to the one introduced in [2], where a constant velocity was taken into account

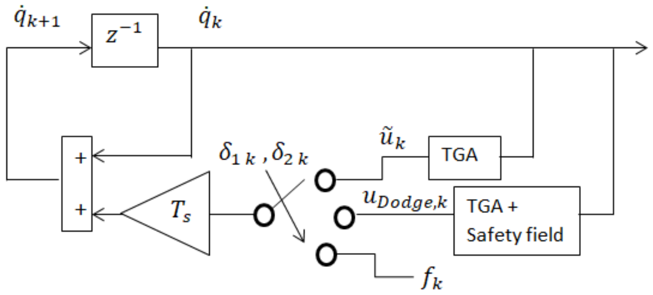

Fig. 1: Switching control scheme implementing the 3 operational modes and using a Trajectory Generation Algorithm (TGA.).

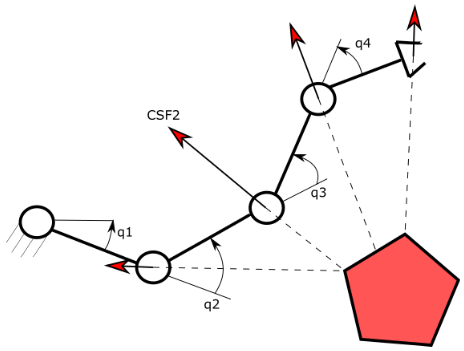

Fig. 2: A generic manipulator and an obstacle (in red): arrows symbolize the safety field induced by the obstacle.

whereas the third one is chosen together with the switching signal among the three modes, by solving an optimal control problem for a suitably defined hybrid system. The resulting system is schematically drawn in Figure 1, where δ_1 and δ_2 are binary variables implementing the switching signal among the three modes with input $\tilde{\boldsymbol{u}}$, $\boldsymbol{u_{Dodge}}$, and $\boldsymbol{f}$.

1) Nominal operation mode: Every industrial manipulator has an internal trajectory generator, that is able to compute a trajectory from a starting configuration $\boldsymbol{q}_k$ to an ending one $\boldsymbol{q}_f$. This open loop trajectory is realized by a sequence of acceleration inputs $\{\tilde{\boldsymbol{u}}_k, \tilde{\boldsymbol{u}}_{k+1}, ..., \tilde{\boldsymbol{u}}_{k+N-1}\}$ and is designed via some trajectory generation algorithm, which accounts for kinematic limitations on maximum/minimum joint accelerations and maximum/minimum velocities, but not for the presence of obstacles. In robotics, common choices for trajectory generation are trapezoidal speed profile, cycloidal, minimum jerk trajectory, polynomial law and others.

Here, we refer to nominal acceleration as the sequence of acceleration inputs determined by a trajectory generation algorithm to drive the manipulator from its current configuration to the target one. In our simulation example, we shall use the open source Reflexxes library [12] to compute nominal accelerations, even though the proposed approach remains applicable to any other trajectory generation algorithm. If we apply at a certain time k an acceleration different from $\tilde{\boldsymbol{u}}_k$, we will reach a state at $k+1$ different from the nominal one. From that state, we can recompute a new sequence of accelerations that will drive the system to the target configuration.

2) Soft safety operation mode: The concept of safety field introduced in [6] is used for designing the soft corrective acceleration $\boldsymbol{u_{Dodge}}$. A safety field is basically a repulsive field define all over the configuration space, which is produced by the set of obstacles present in the scene (also the operator swept volumes). The safety field can be interpreted as a set of forces acting on some points along the kinematic chain of a manipulator as illustrated in Figure 2. [6] shows how to compute a configuration $\boldsymbol{q}_{ev}$ induced by the safety field such that the dodging maneuver driving to $\boldsymbol{q}_{ev}$ has a high chance of satisfying the safety constraint, even if this is not guaranteed 100%. By exploiting the properties of the geometrical Jacobian of the manipulator J, [6] defines $\boldsymbol{q}_{ev}$ as follows:

$$\boldsymbol{q}_{ev} = \boldsymbol{q}_k + J_N^{\dagger}(\boldsymbol{x}_{trg} - \boldsymbol{x}_k) + \boldsymbol{q}_{CSF} - J_N^{\dagger} \cdot J_N \boldsymbol{q}_{CSF} \tag{4}$$

where $J_N^{\dagger}$ is the pseudoinverse of the end effector geometrical Jacobian; $\boldsymbol{x}_{trg}$ and $\boldsymbol{x}_k$ are respectively the end effector position in the cartesian space for the target final configuration and the configuration at time step k; $\boldsymbol{q}_{CSF} = \sum_{k=1}^{N} J_h^{\top} CSF_h$, where CSF_h is the safety field acting on the h-th point of the kinematic chain, while J_h is the geometrical Jacobian referred to the same point.

For the configuration $\boldsymbol{q}_{ev}$, the manipulator results farther from the obstacles generating the safety field, while at the same closer to $\boldsymbol{x}_{trg}$. By using a trajectory generation algorithm, like the one implemented in the Reflexxes library, one can compute a series of accelerations that lead the manipulator to $\boldsymbol{q}_{ev}$ from the current state, while accounting for kinematic limitations of the manipulator. The obtained sequence of acceleration inputs, truncated at time $(k+N-1)T_s$, represents the soft corrective acceleration $\{\boldsymbol{u}_{Dodge,k}, \boldsymbol{u}_{Dodge,k+1}, \ldots, \boldsymbol{u}_{Dodge,k+N-1}\}$ associated with the soft safety operation mode.

3) Hard safety operation mode: The third possible acceleration input $\boldsymbol{f}$ is a hard corrective acceleration which makes the manipulator follow a trajectory that is safe. In our scheme, it is the only acceleration that is not computed by a trajectory generator algorithm but is determined together with the switching signals so as to enforce the constraints expressed in equation (3), in those cases when other modes cannot. As a matter of fact, the nominal acceleration input neglects the presence of obstacles, whereas the soft corrective acceleration input based on the safety field is not guaranteed to be 100% safe, hence, it is sometimes necessary to apply the acceleration $\boldsymbol{f}$ to ensure the satisfaction of the safety constraint. Usage of this mode is penalized when designing the switching signals so as to avoid non smooth motion.

4) Optimal switching rule design: We now have to decide at every time step k which mode the system has to follow and, in case the hard safety mode is chosen, we have to compute an optimal value for $\boldsymbol{f}_k$. These decisions are taken jointly by formulating and solving at each time step k a suitably defined optimization problem over the time horizon from k to $k+N$, according to the receding horizon strategy in MPC. For the rest of this section, we will describe how to build such a problem and compute a solution. We start by introducing the equations governing the evolution of the switched system that comprises the three possible acceleration inputs:

$$\begin{cases}\dot{\boldsymbol{q}}_{k+1} = & \dot{\boldsymbol{q}}_k + T_s\tilde{\boldsymbol{u}}_k + \delta_{1k}T_s(-\tilde{\boldsymbol{u}}_k + \boldsymbol{u}_{Dodge,k})\\ & +\delta_{2k}T_s(-\tilde{\boldsymbol{u}}_k + \boldsymbol{f}_k)\\ \boldsymbol{q}_{k+1} = & \boldsymbol{q}_k + 0.5T_s(\dot{\boldsymbol{q}}_k + \dot{\boldsymbol{q}}_{k+1})\end{cases} \tag{5}$$

Note that, differently from (1), we do not make explicit the dependence of $\boldsymbol{q}$ from the control input $\boldsymbol{u}$ in (5). Variables δ_1 and δ_2 in (5) are Boolean control inputs that serve the purpose of defining the operation mode. The continuous control input $\tilde{\boldsymbol{u}}$ and $\boldsymbol{u}_{Dodge}$ are computed by a trajectory generation algorithm as previously described and are then treated as non controllable inputs with known values in the optimization problem formulation. The only continuous input to be set is $\boldsymbol{f}$.

Equation (5) is bilinear because of the product between the optimization variables δ_2 and $\boldsymbol{f}$. We can however reduce it to a linear expression by introducing an auxiliary variable $\boldsymbol{z} = \delta_2 T_s \boldsymbol{f}$ and enforcing the following set of constraints (these are results from the propositional calculus theory, see [7]) over the time horizon of interest

$$\begin{cases} m\delta_{2\,k+i} \leq z_{k+i} \leq M\delta_{2\,k+i}\\ M(\delta_{2\,k+i}-1) \leq z_{k+i} - T_s\boldsymbol{f}_{k+i} \leq m(\delta_{2\,k+i}-1)\end{cases} \tag{6}$$

$i = 0, \dots, N-1$, where m and M are suitable lower and upper bounds on the values taken by $T_s\boldsymbol{f}$, that is: $m = -T_s a_{max}$ and $M = T_s a_{max}$, a_{max} being the vector of maximal possible joint accelerations. This kinematic limitations must be applied to $\boldsymbol{f}$ as well, by imposing

$$-a_{max} \leq \boldsymbol{f}_{k+i} \leq a_{max},\, i = 0, \dots, N-1. \tag{7}$$

For ease of notation, we shall define: $r = T_s\tilde{\boldsymbol{u}}$; $\Delta d = T_s(-\tilde{\boldsymbol{u}} + \boldsymbol{u}_{Dodge})$ and $\Delta r = -T_s\tilde{\boldsymbol{u}}$. Accordingly, equation (5) can then be rewritten as:

$$\begin{cases}\dot{\boldsymbol{q}}_{k+1} = \dot{\boldsymbol{q}}_k + r_k + \delta_{1k}\Delta d_k + \delta_{2k}\Delta r + \boldsymbol{z}_k\\ \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + 0.5T_s(\dot{\boldsymbol{q}}_k + \dot{\boldsymbol{q}}_{k+1})\end{cases}$$

Now, it is possible to unroll the system dynamics and describe the evolution of $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$ during the horizon from k to $k+N$ in the following compact form:

$$\begin{cases}\dot{\boldsymbol{Q}} = \dot{\boldsymbol{Q}}_k + R + \Delta D\Delta_1 + \Delta R\Delta_2 + B_z Z\\ \boldsymbol{Q} = \boldsymbol{Q}_k + 0.5T_s\dot{\boldsymbol{Q}}_k + C\dot{\boldsymbol{Q}}\end{cases} \tag{8}$$

where we set

$$\dot{\boldsymbol{Q}} = \begin{bmatrix}\dot{\boldsymbol{q}}_{k+1}^\top & \cdots & \dot{\boldsymbol{q}}_{k+N}^\top\end{bmatrix}^\top \quad \boldsymbol{Q} = \begin{bmatrix}\boldsymbol{q}_{k+1}^\top & \cdots & \boldsymbol{q}_{k+N}^\top\end{bmatrix}^\top$$

$$\dot{\boldsymbol{Q}}_k = \begin{bmatrix}\dot{\boldsymbol{q}}_{k}^\top & \cdots & \dot{\boldsymbol{q}}_{k}^\top\end{bmatrix}^\top \quad \boldsymbol{Q}_k = \begin{bmatrix}\boldsymbol{q}_{k}^\top & \cdots & \boldsymbol{q}_{k}^\top\end{bmatrix}^\top$$

$$\Delta_i = \begin{bmatrix}\delta_{i\,k} & \cdots & \delta_{i\,k+N-1}\end{bmatrix}^\top,\, i = 1, 2$$

$$R = \begin{bmatrix}r_k^\top & (r_k + r_{k+1})^\top & \cdots (r_k + \cdots + r_{k+N-1})^\top\end{bmatrix}^\top$$

$$Z = \begin{bmatrix}z_k^\top & z_{k+1}^\top & \cdots z_{k+N-1}^\top\end{bmatrix}^\top$$

$$\Delta D = \begin{bmatrix}\Delta d_k & 0 & & 0\\ \Delta d_k & \Delta d_{k+1} & & 0\\ \vdots & \vdots & \cdots & 0\\ \Delta d_k & \Delta d_{k+1} & & \Delta d_{k+N-1}\end{bmatrix}$$

$$B_z = \begin{bmatrix}I_n & 0 & & 0\\ I_n & I_n & & 0\\ \vdots & \vdots & \cdots & 0\\ I_n & I_n & & I_n\end{bmatrix} \quad C = T_s\begin{bmatrix}\frac{1}{2}I_n & 0 & & 0\\ I_n & \frac{1}{2}I_n & \cdots & 0\\ \vdots & \vdots & & \vdots\\ I_n & I_n & & \frac{1}{2}I_n\end{bmatrix}$$

with I_n denoting the identity matrix of size n, where n is the number of degree of freedom of the manipulator. ΔR is obtained similarly to ΔD, replacing Δd with Δr.

We next describe all constraints that must hold on the control and state variables of the system in equation (8).

We start by the reformulation of the safety specifications in (3) according to the introduced notations:

$$\tilde{E} \cdot (\dot{\boldsymbol{Q}}_k - \dot{\boldsymbol{Q}} + 2B_z\dot{\boldsymbol{Q}}) \leq \tilde{H} \tag{9}$$

where matrix $\tilde{E}$ is block-diagonal, and every block is equal to E, while $\tilde{H} = \begin{bmatrix}H^\top & \cdots & H^\top\end{bmatrix}^\top$. After some manipulations we can get the following expression:

$$-\tilde{E} \cdot \begin{bmatrix}B_z & \Delta R & \Delta D\end{bmatrix} \cdot \begin{bmatrix}Z\\ \Delta_1\\ \Delta_2\end{bmatrix} \leq \tilde{H} + \tilde{E} \cdot (R - 2B_z\dot{\boldsymbol{Q}}) \tag{10}$$

The bound on the maximal and minimal joint speed can be written as follows:

$$-v_{max} \leq \dot{\boldsymbol{Q}}_k + \begin{bmatrix}B_z & \Delta R & \Delta D\end{bmatrix} \cdot \begin{bmatrix}Z\\ \Delta_1\\ \Delta_2\end{bmatrix} + R \leq v_{max} \tag{11}$$

where v_{max} is an appropriately defined vector associated with the maximal velocity.

Regarding variables Δ_1 and Δ_2 it is not sufficient to simply impose that they are Boolean. Indeed, we want to enforce the fact that at each time step $k+i$, the system (5) can evolve following only a single mode at a time. We have therefore to add to our problem N constraints, one for each time step $k+i$, $i = 0, 1, \dots, N-1$, ensuring that:

$$\Delta_1 + \Delta_2 \leq \mathbf{1}_N, \tag{12}$$

where $\mathbf{1}_N$ is a column vector of dimension N with all elements equal to 1.

Setting $F = \begin{bmatrix}f_k^\top & f_{k+1}^\top & \cdots f_{k+N-1}^\top\end{bmatrix}^\top$, we can group all control variables along the reference time horizon in a single vector $\boldsymbol{v} = \begin{bmatrix}F^\top & Z^\top & {\Delta_1}^\top & {\Delta_2}^\top\end{bmatrix}^\top$.

Among all admissible values for $\boldsymbol{v}$ (see the constraints in (6), (7), and (10)–(12)), our objective is selecting the best one that drives the the manipulator to the target configuration and makes it accomplish its task. To this purpose we introduce the following cost function $\mathcal{J}(\boldsymbol{v})$:

$$\mathcal{J}(\boldsymbol{v}) = \|\boldsymbol{Q} - \boldsymbol{Q}_f\|_{W_q} + \|\begin{bmatrix}\Delta_1^\top & \Delta_2^\top\end{bmatrix}^\top\|_{W_\delta} + \|F\|_{W_f} \tag{13}$$

where $\boldsymbol{Q}_f = \begin{bmatrix}\boldsymbol{q}_f^\top & \cdots & \boldsymbol{q}_f^\top\end{bmatrix}^\top$, $\boldsymbol{q}_f$ being the desired target configuration. The operator $\|x\|_M$ is equal to $x^\top Mx$. W_q, W_δ and W_f are positive definite matrices of appropriate

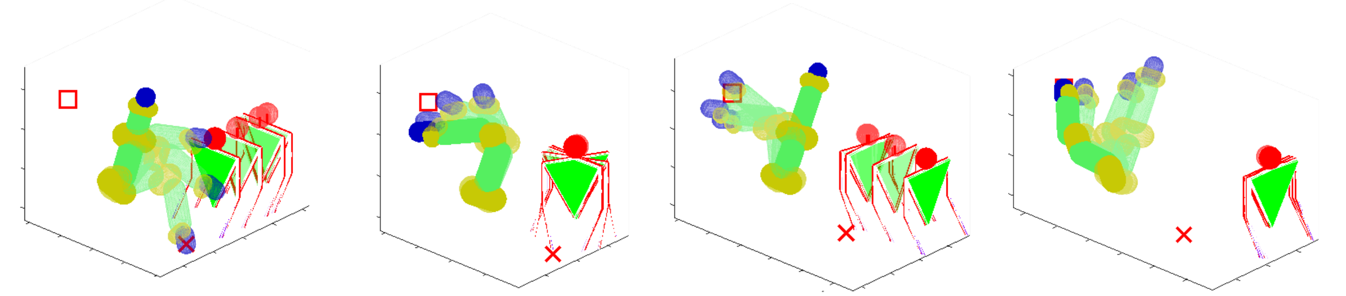

Fig. 3: Trajectory followed by the manipulator and the human during part 1 of the pick and place task. A red cross indicates the pick position, while a red square the place one. From left to right, snapshots refer to the time intervals: [0, 3.05]; [3.55, 6.55]; [7.05, 11.55]; [11.55, 13.05] (in seconds). Within $[0, 6.55]$ seconds, the manipulator deviates from its nominal path (which is a simple straight line in joint space connecting pick and place) to avoid collisions with the human, which is getting closer. Frames within $[7.05, 13.05]$ show that the manipulator follows a nominal trajectory leading to the place position.

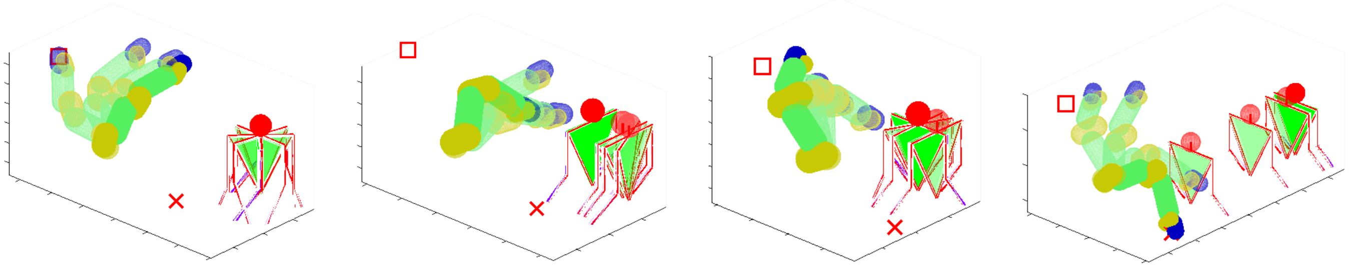

Fig. 4: Trajectory followed by the manipulator and the human during part 2 of the pick and place task. A red cross indicates the pick position, while a red square the place one. From left to right, snapshots refer to the time intervals: [13.2,16.2]; [16.7, 20.3]; [20.8, 26.05]; [26.55, 31.45] (in seconds). A dodging maneuver is required in the time window $[16.7, 26.05]$. Then the manipulator returns to the initial pick configuration following a nominal trajectory.

dimensions. In particular, W_q is block-diagonal so that $\|\boldsymbol{Q}-\boldsymbol{Q}_f\|_{W_q} = \sum_{i=1}^{N} \|\boldsymbol{q}_{k+i}-\boldsymbol{q}_f\|_{W_{q,i}}$ penalizes the distance from the target configuration, possibly giving a different weight at each time step. The adoption of the soft or the hard corrective accelerations are penalized so as to favor the use of the nominal acceleration input defining the nominal trajectory for task accomplishment.

In order to make $\mathcal{J}(\boldsymbol{v})$ explicit as a function of $\boldsymbol{v}$, one can plug into equation (13) the expression for $\boldsymbol{Q}$ given in (8), thus finally obtaining a quadratic form of the type:

$$\mathcal{J}(\boldsymbol{v}) = \boldsymbol{v}^\top A \boldsymbol{v} + G\boldsymbol{v} \tag{14}$$

neglecting terms that are independent of the optimization variables and hence do not affect the value of the minimizers.

We finally are in a position to formulate the constrained optimization problem that is solved at every time step k, till the target configuration is reached:

$$\begin{aligned} &\text{minimize } \mathcal{J}(\boldsymbol{v}) \text{ in (14)} \\ &\text{subject to: (6), (7), (10), (11) and (12)} \end{aligned} \tag{15}$$

It is easily seen that this is a Mixed Integer Quadratic Program (MIQP), since it involves continuous and discrete optimization variables. One can then use a MIQP solvers like Cplex [13] to compute at every step k the optimal control input $\boldsymbol{v}$, i.e., the operation mode and the hard corrective acceleration input if the hard safety mode is selected.

Both switching signals δ_1 and δ_2 could present some chattering behavior. To alleviate this problem, it is convenient to introduce in the cost function J additional terms that penalize two consecutive values for δ_1 (or δ_2) that are different: $\mathcal{J}^{'}(\boldsymbol{v}) = \mathcal{J}(\boldsymbol{v}) + \sum_{i=0}^{N-1}(\delta_{1\,k+i}-\delta_{1\,k+i+1})\cdot(\delta_{1\,k+i}-\delta_{1\,k+i+1}) + \sum_{i=0}^{N-1}(\delta_{2\,k+i}-\delta_{2\,k+i+1})\cdot(\delta_{2\,k+i}-\delta_{2\,k+i+1})$, which is still a quadratic form.

It is worth pointing out the importance of the soft safety mode. Applying only the hard corrective acceleration input to avoid violating the safety constraint could drive the manipulator in some local minimum for $\mathcal{J}$. Instead the safety field is able to make it escape from this kind of configurations.

III. Numerical simulation example

We assess the performance of our approach on a numerical simulation example where a manipulator performs a pick and place task. The scene is shared with an operator that walks close to the robot. We consider as a manipulator the left arm of the ABB IRB 14000, which has 7 d.o.f. In this experiment, we impose accelerations only to four joints, i.e, the ones that are closest to the fixed base, and set constant and equal to zero the position of the other three. The same kinematic model introduced in [3] is considered for the human. Differently from [3], here we assume to know the actual values for positions and velocities of the human joints (in a more realistic context they can be estimated exploiting

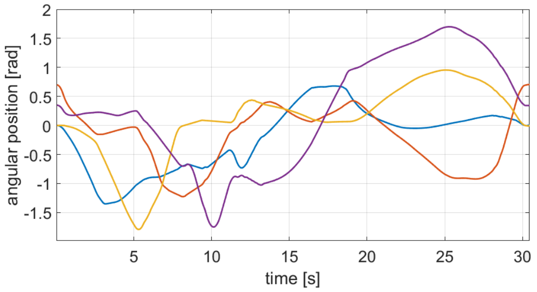

Fig. 5: Trajectory of the angular position of 4 joints.

sensors, as in [3]).

The value of the sample time T_s is set equal to 0.05 s, while the MPC strategy is applied with reference to a prediction horizon of length $N \cdot T_s$, where $N = 5$. The cost function $\mathcal{J}(\boldsymbol{v})$ in (14) is adopted where W_δ is block diagonal with on the diagonal the identity matrices I_5 and $1.5 \cdot I_5$, and also W_q and W_f are block diagonal but with all matrices on the diagonal identical and given by $W_{q,i} = 2 \cdot I_4$, $i = 1, \ldots, 5$, for W_q, and $W_{f,i} = 0.1 \cdot I_4$, $i = 1, \ldots, 5$, for W_f.

Figures 3 and 4 plot some snapshots of the trajectories followed by the manipulator and the human. Figure 3 refers to part 1, where the manipulator moves from pick position to place position. Figure 4 refers to part 2, where the manipulator returns to the starting pick position. The end effector position for pick is indicated with a red cross, while that for place is indicated with a red square. To approximate the occupancy of the links, several capsules are introduced. As can be seen, the manipulator sometimes adopts corrective maneuvers to dodge the operator, yet it keeps trying to get closer to its target position in both part 1 and 2 of its task. In Figure 5, the evolution of joints angular displacements $\boldsymbol{q}(t)$ is reported for the same experiment, and it clearly shows that the overall trajectory is quite smooth. The total time to compute the trajectory was 12.77 s, which means that about 0.02 seconds were needed at every time step of duration 0.05 to solve the constrained optimization problem (15).

The percentage of time spent in each mode for the proposed switching system is reported in Table I, where it is compared with the results obtained for the same pick and place task and human trajectory, but adopting the strategy in [3] which commutes between a normal mode and a human avoidance mode where some corrective acceleration is applied (see [3] for details). The total time required to complete the task with our approach was 30.5 s, while it is 34.2 s (12.1 % more) with the approach in [3]. This time reduction is obtained because of the adoption of the MPC strategy that predicts the future behavior along some (short) look-ahead time horizon to optimize the choice of the control action.

IV. Conclusions

Safe motion planning in human-robot coexistence scenario is one of the main topics of interest in the area of human robot collaboration.

	nominal	soft corrective	hard corrective	avoidance
our paper	0.315	0.142	0.542	-
[3]	0.435	-	-	0.565

TABLE I: Pick and place simulation example: percentage of time spent per mode in our approach and in the one in [3].

In this work, we have shown how to exploit MPC for hybrid systems to design a controller that makes a manipulator perform its task in a safe way when sharing its environment with a human operator. The adopted strategy reduces the problem to an optimization program of the MIQP form, which in our simulations appears computationally affordable for online implementation. One could actually reduce the computational time by stopping the computations when a feasible (and hence safe) solution is found, at the price of deteriorating performance.

References

[1] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schaffer, "On making robots understand safety: Embedding injury knowledge into control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.

[2] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nov 2013, pp. 1131–1136.

[3] M. Ragaglia, A. M. Zanchettin, and P. Rocco, "Safety-aware trajectory scaling for human-robot collaboration with prediction of human occupancy," in *2015 Int. Conf. on Advanced Robotics*, July 2015, pp. 85–90.

[4] F. Flacco, T. Kroger, A. D. Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *2012 IEEE Int. Conf. on Robotics and Automation*, May 2012, pp. 338–345.

[5] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon, "Planning human-aware motions using a sampling-based costmap planner," in *2011 IEEE Int. Conf. on Robotics and Automation*, May 2011, pp. 5012–5017.

[6] M. P. Polverini, A. M. Zanchettin, and P. Rocco, "Real-time collision avoidance in human-robot interaction based on kinetostatic safety field," in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sept 2014, pp. 4136–4141.

[7] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407 – 427, 1999.

[8] T. M. Cavalier, P. M. Pardalos, and A. L. Soyster, "Modeling and integer programming techniques applied to propositional calculus," *Computers and Operations Research*, vol. 17, no. 6, pp. 561 – 570, 1990.

[9] A. Bemporad, J. Roll, and L. Ljung, "Identification of hybrid systems via mixed-integer programming," in *40th IEEE Conf. on Decision and Control*, vol. 1, 2001, pp. 786–792.

[10] M. F. Belazreg, K. Halbaoui, M. H. Boulheouchat, and D. Boukhetala, "Modelling, simulation and control of hybrid system integrating logic, dynamics, and constraints using hybrid automaton, apros and mixed integer quadratic optimization algorithm," in *2016 8th Int. Conf. on Modelling, Identification and Control (ICMIC)*, Nov 2016, pp. 634–642.

[11] M. Rungger and O. Stursberg, "Optimal control for deterministic hybrid systems using dynamic programming," *IFAC Proceedings Volumes*, vol. 42, no. 17, pp. 316 – 321, 2009, 3rd IFAC Conference on Analysis and Design of Hybrid Systems.

[12] T. Kroger, "Opening the door to new sensor-based robot applications; the reflexxes motion libraries," in *2011 IEEE Int. Conf. on Robotics and Automation*, May 2011, pp. 1–4.

[13] "IBM ILOG CPLEX Optimizer," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, Last 2010.