2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Efficient ISO/TS 15066 Compliance through Model Predictive Control

Andrea Pupa[1] and Cristian Secchi[1]

Abstract— **In the actual industrial scenarios, human operators and robots work together sharing the workspace. Such proximity requires special attention in ensuring safety for the human operator, which is often translated in collision avoidance behaviour or high speed reduction. Adhering safety however is not the only aspect that must be taken into account. For many tasks, such as welding, it is crucial to ensure that the robot performs exactly the planned path. To optimize robot performance while complying with safety regulations, this work introduces a novel optimal nonlinear control problem. It prioritizes path preservation, exploiting redundancy to minimize task execution time, while explicitly adhering to the constraints imposed by ISO/TS 15066. To achieve high-performance outcomes, the control problem is addressed using the Model Predictive Control (MPC) approach. The proposed strategy has been experimentally validated in both simulations and a real-world industrial task involving a Kuka LWR4+ robot.**

I. Introduction

In recent years, industrial scenarios have experienced a deep transformation with the advent of collaborative robotics. This paradigm shift has necessitated the introduction of new safety standards to regulate and adapt to this emerging technology [1]. These standards, namely ISO 10218-1 and ISO 10218-2 [2], [3], define four distinct collaborative modes: *safety-rated monitored stop* (SMS), *hand guiding* (HG), *speed and separation monitoring* (SSM) and *power and force limiting* (PFL). Additionally, technical specification ISO/TS 15066 [4] provides guidelines for assessing the risk associated with each collaborative mode. In real industrial applications, the SSM mode is often preferred due to its simplicity of implementation. In this collaborative mode, the robot speed is dynamically adapted based on the distance with the human operator.

As a result, numerous researchers have directed their efforts toward developing strategies to ensure the safety of human operators and achieve collision-free trajectories. One widely employed approach involves the use of potential fields. In [5] the authors exploit the concept of static and kinetostatic danger fields to implement a collision behaviour with human operators. Similarly, in [6] the runtime collision-free trajectories are generated at runtime building a potential field around the whole robot body. In [7] an efficient control scheme for safe human-robot collaboration (HRC) is proposed, leveraging attractive and repulsive danger fields to impose constraints on control inputs.

[1] A. Pupa and C. Secchi are with the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy. E-mail: {andrea.pupa, cristian.secchi}@unimore.it

Project funded under the National Recovery and Resilience Plan (NRRP), Mission 04 Component 2 Investment 1.5 – NextGenerationEU, Call for tender n. 3277 dated 30/12/2021. Award Number: 0001052 dated 23/06/2022.

Despite their simplicity of implementation and effectiveness in straightforward environments, potential fields do not ensure the execution of tasks. Their susceptibility to local minima can potentially compromise task completion. To address this challenge, many researchers have focused on developing optimization-based strategies. In [8] the authors present a control algorithm that addresses safety while trying to preserve the desired trajectory. The strategy exploits the use of control barrier functions [9] around the robot body and the human body to maintain a collision-free trajectory while adhering to ISO/TS 15066. In [10] the authors propose to exploit the use of a Model Predictive Control (MPC) to ensure that the robot safely replans its motion in the presence of a human operator.

However, there are scenarios where deviating from the planned path is not feasible for task execution, such as industrial operations in highly cluttered environments or orientation constraints. This can be addressed by exploiting trajectory scaling algorithms to efficiently adjust the robot speed along the planned path. In [11] a trajectory scaling algorithm that ensures a safe robot behaviour is proposed. In particular, the velocity limits imposed by SSM and PFL are merged to define a new safe velocity limit and used as an upper bound in the trajectory scaling algorithm. In [12] the authors propose to integrate an online planning strategy with a safety-aware scaling optimization algorithm. The robot moves along a planned path adapting the speed based on the safety standards constraint. When the reduction of the speed is too high, the current path has become largely inefficient. The planner is then triggered to plan a new collision-free trajectory, allowing the robot to reach the desired goal faster. These scaling algorithms, however, do not consider the possibility of redundancy and may be largely inefficient when dealing with them, i.e. tasks that require to constrain only the end-effector pose. For instance, a scaling algorithm might cause a robot to slow down unnecessarily when a part of its body, such as the elbow, is near a human operator, while the end-effector could still follow the desired trajectory. Consequently, leveraging the redundancies within the robot manipulator can prove advantageous in avoiding unnecessary reductions in speed and enhancing overall performance. This has been partially addressed in [13] where the authors propose a MPC approach to deal with task scaling technique while respecting the kinematic constraints. However, there is no guarantee of preserving the geometrical path when it is included as a term in the cost function. A more robust approach would be to enforce this preservation by exploiting the Hierarchical Quadratic Programming (HQP) formulation [14], where the

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

constraints are imposed in a hierarchic way. Additionally, it is important to underline that [13] does not explicitly consider the constraint imposed by safety regulations, i.e. it is not suitable for real industrial scenarios.

Drawing inspiration from HQP [14], in this paper an extension of the framework introduced in [12] is proposed. A novel optimal nonlinear control problem for task scaling is proposed. Starting from a collision-free trajectory in the task space, the optimal control framework minimizes the deviations from the nominal trajectory such that the robot end-effector will follow exactly the desired path while explicitly adhering to constraints imposed by ISO/TS 15066. When the human operator approaches the robot body without obstructing the end-effector, the optimization problem leverages redundancies to prevent unnecessary speed reductions without deviating from the desired path, outperforming [12]. On the contrary, when the human operator obstructs the robot at the end-effector, the proposed approach is able to guarantees a safe collaboration as the method proposed in [12], without any loss of performance. Furthermore, the proposed optimization problem can be easily extended to other tasks [14], making it applicable to real industrial scenarios.

Summarizing, the main contributions of this paper are:

- A novel optimal nonlinear control problem that explicitly considers redundancy in order to improve the adaptability of the robot without affecting the performance.
- An extensive validation, both in simulated and real environments, proving the effectiveness of the framework.

The paper is organized as follows: in Sec. II the problem statement is presented, while Sec. III details the proposed optimal control problem with a deep explanation of all the constraints and the cost function. Lastly, Sec. IV provides the experimental validation, both in simulated and real environments, while Sec. V addresses the conclusion and future works.

II. Problem Statement

Consider an HRC application where a n-DoFs velocity-controlled robot manipulator has to collaborate with a human operator in order to accomplish a common job. The velocity-controlled robot is modeled as:

$$\dot{\boldsymbol{q}} = \boldsymbol{u}, \tag{1}$$

where $\dot{\boldsymbol{q}} \in \mathbb{R}^n$, and $\boldsymbol{u} \in \mathbb{R}^n$ are the joint velocities and the controller input, respectively.

During the collaboration, the robot has to perform a set of tasks each of which is associated with a task-space trajectory $\boldsymbol{x}_{des}(t) \in \mathbb{R}^m$, with $m \leq n$, that goes from an initial pose $\boldsymbol{x}_{des}(t_i) = \boldsymbol{x}_i \in \mathbb{R}^m$ to a desired final pose $\boldsymbol{x}_{des}(t_f) = \boldsymbol{x}_f \in \mathbb{R}^m$. Such trajectory is considered already optimal, i.e. it is computed with standard algorithms that optimize a desired criteria, e.g. [15]–[17]. To describe the robot motion in task space, it is possible to use the kinematic model:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}_T(\boldsymbol{q})\boldsymbol{u}, \tag{2}$$

where $\boldsymbol{J}_T \in \mathbb{R}^{m \times n}$ represents the task-space Jacobian, i.e. the Jacobian that maps the joint velocities $\dot{\boldsymbol{q}}$ in the task-space of dimension m. From equation (2) it is possible to compute the input for the system:

$$\boldsymbol{u} = \boldsymbol{J}_T^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}}_{des} + \boldsymbol{P}_{ns}\boldsymbol{u}_{ns}, \tag{3}$$

where $\boldsymbol{P}_{ns} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{u}_{ns}$ is the projector in the null space and the null space input, respectively.

Consider tasks whose execution is guaranteed only if the robot end-effector precisely follows the path associated with the trajectory. In such cases, this trajectory can be decomposed by applying a path-velocity decomposition:

$$\boldsymbol{x}_{des}(t) = \boldsymbol{x}_{des}(s(t)) \qquad t \in \left[t_i, t_f\right], \tag{4}$$

$$\dot{\boldsymbol{x}}_{des}(t) = \boldsymbol{x}'_{des}(s(t))\dot{s} \qquad t \in \left[t_i, t_f\right], \tag{5}$$

where $s \in \mathbb{R}$ is the curvilinear abscissa that parametrizes the geometrical path $\boldsymbol{x}_{des}(s(t))$, $\boldsymbol{x}'_{des}(s(t))$ is the vector tangent to the desired path, and $\dot{s}$ constitutes the magnitude of the robot velocity. Thanks to this decomposition, it is possible to modulate the robot speed without changing the computed path, i.e. acting on $\dot{s}$.

Furthermore, it is also crucial that the robot does not harm the human operator. Indeed, to ensure a safe collaboration, the robot behaviour must be compliant with the safety limits imposed by ISO/TS 15066 standard. In particular, focusing on the SSM collaborative mode, it is possible to derive an upper bound on the robot speed towards the human operator [12]:

$$\begin{aligned} v_{rh}(t) \leq & \sqrt{v_h(t)^2 + (a_{max}T_r)^2 - 2a_{max}K(t)} + \\ & - a_{max}T_r - v_h(t), \end{aligned} \tag{6}$$

where $K(t) = C + Z_d + Z_r - S_p(t)$. $v_{rh}(t) \in \mathbb{R}$ and $v_h(t) \in \mathbb{R}$ are the scalar velocity of the robot towards the human operator and the scalar velocity of the human operator towards the robot, respectively. $a_{max} \in \mathbb{R}$ is the maximum deceleration and $T_r \in \mathbb{R}$ is the robot reaction time. C is the intrusion distance, i.e. the distance that a part of the body can intrude into the sensing field before it is detected, while Z_d and Z_r are the position uncertainties of the human operator inside the workspace and of the robot system, respectively. Lastly, S_p represents the protective separation distance. For this reason, the shared workspace is equipped with a monitoring unit that allows tracking the human operator during the collaboration.

This work aims at ensuring a safe and efficient HRC collaboration. The primary objective is to ensure that the robot end-effector follows exactly the desired path for correct task execution. However, there may be situations when the robot speed needs to be reduced to ensure compliance with the safety regulations, as expressed in (6). To achieve this, this paper introduces a novel optimal control problem that computes the optimal robot input $\boldsymbol{u}$ over time such that:

- the robot end-effector path is exactly as the desired path associated with $\boldsymbol{x}_{des}(t)$.
- the robot speed is reduced as low as possible in the task space, i.e. $\dot{s}$ is as close as possible to its desired value.

- the robot speed towards the human operator is lower than the admissible value computed according to the ISO/TS 15066.

III. Optimal Control Problem

This section aims at presenting and illustrating the proposed optimal control problem. The goal of this problem is to reduce as little as possible the speed of the robot while ensuring that the executed end-effector path is exactly the desired one and that the robot behaviour is compliant with the ISO/TS 15066.

A. Path-Preserving Constraint

To ensure the correct execution of the task, it is essential for the robot end-effector to exactly follow the planned path. Thus the robot speed can be modulated at runtime without changing the overall path performed. This can be achieved by multiplying the rate of change of curvilinear abscissa, denoted as $\dot{s}$, by a scaling factor represented as $\alpha \in [0, 1]$. Scaling all the joint velocities, as proposed in [12], can lead to inefficient behaviour, i.e. the robot speed may be scaled too much. Indeed, since only the task path must be respected, it is possible to exploit the robot redundancies to increase the distance between the robot and the human. This approach may help reduce speed modulations when the robot is close to the human operator with links other than the end-effector.

For ease of notation, the trajectory can be parameterized so that the rate of change of curvilinear abscissa remains constant over time, i.e. $\dot{s}(t) = \bar{\dot{s}}$. While this is a common technique exploited in literature [13], it is worth underlining that all the formulation can be easily extended to a case with a time-varying $\dot{s}(t)$. The path-preserving constraint is expressed as follows:

$$\boldsymbol{J}_T(\boldsymbol{q})(t)\boldsymbol{u}(t) = \boldsymbol{x}'_{des}(s(t))\alpha(t)\bar{\dot{s}}. \tag{7}$$

Imposing the constraint in (7) with $\boldsymbol{u}$ as optimization variables implicitly ensures that the robot can perform movements in the null space to maximize the desired cost function, which is detailed in Sec. III-D. This is demonstrated by plugging equation (3) in the constraint equation where, for ease of notation, the dependency with the time has been removed:

$$\boldsymbol{J}_T(\boldsymbol{q})\boldsymbol{u} = \alpha\boldsymbol{x}'_{des}(s)\bar{\dot{s}}, \tag{8}$$

$$\boldsymbol{J}_T(\boldsymbol{q})(\boldsymbol{J}_T^{\dagger}(\boldsymbol{q})\alpha\boldsymbol{x}'_{des}(s)\bar{\dot{s}} + \boldsymbol{P}_{ns}\boldsymbol{u}_{ns}) = \alpha\boldsymbol{x}'_{des}(s)\bar{\dot{s}}, \tag{9}$$

$$\alpha\boldsymbol{x}'_{des}(s)\bar{\dot{s}} + \boldsymbol{0}\boldsymbol{u}_{ns} = \alpha\boldsymbol{x}'_{des}(s)\bar{\dot{s}}, \tag{10}$$

where the definition $\boldsymbol{J}_T(\boldsymbol{q})\boldsymbol{P}_{ns} = \boldsymbol{0}$ has been used [18]. From (10) it is clear that using $\boldsymbol{u}$ as a variable to optimize the null-space input $\boldsymbol{u}_{ns}$ is automatically computed without affecting the resulting task.

B. Speed and Separation Monitoring Constraint

This constraint aims to ensure that the robot velocity toward the human operator remains lower than the maximum admissible speed, as defined by equation (6). This constraint is obtained by projecting the Cartesian velocity of each link into the robot-human direction and then imposing the ISO/TS 15066 upper bound. Mathematically, this can be expressed by:

$$\boldsymbol{n}_{rh_i}(t)\boldsymbol{J}_{p_i}(\boldsymbol{q}_{p_i}(t))\boldsymbol{u}_{p_i} \leq v_{max_i}(t) \quad \forall i \in \{1, \ldots, n\}, \tag{11}$$

where $\boldsymbol{n}_{rh_i}$ is the unit vector representing the direction from the i-th robot link to the human, which can be computed, e.g. representing both the human operator and the robot as capsules [19] or using the weighted distance [20]. $\boldsymbol{J}_{p_i}$ is the partial Jacobian of the i-th link, while $\boldsymbol{q}_{p_i}$ and $\boldsymbol{u}_{p_i}$ are the subvectors of $\boldsymbol{q}$ and $\boldsymbol{u}$ containing the first i-th joints and inputs, respectively. For ease of notation, a new variable called modified Jacobian is introduced:

$$\boldsymbol{J}_{r_i}(\boldsymbol{q}_{p_i}(t), t) = \boldsymbol{n}_{rh_i}(t)\boldsymbol{J}_{p_i}(\boldsymbol{q}_{p_i}(t)) \tag{12}$$

During the execution, $\boldsymbol{n}_{rh_i}$ changes over time based on both the robot movements and the human operator movements. Thus, it is necessary to estimate the behaviour of the human operator by leveraging techniques that are already available in literature, e.g. [21], [22].

C. Kinematic Constraint

To be feasible, the optimization problem must be compliant with the kinematic constraint of the system. These can be summarized as:

$$\begin{aligned} &\boldsymbol{q}_{min} \leq \boldsymbol{q}(t) \leq \boldsymbol{q}_{max}, \\ &\dot{\boldsymbol{q}}_{min} \leq \boldsymbol{u}(t) \leq \dot{\boldsymbol{q}}_{max}, \\ &0 \leq s(t) \leq s_{max}, \end{aligned} \tag{13}$$

where $\boldsymbol{q}_{min}$ and $\boldsymbol{q}_{max}$ are the lower and upper bound of the robot joints, while $\dot{\boldsymbol{q}}_{min}$ and $\dot{\boldsymbol{q}}_{max}$ are the one on the input, i.e. the velocity. Lastly, s_{max} represents the upper bound of the curvilinear abscissa and depends on the parametrization chosen.

D. Cost Function

The goal of the optimization problem is to ensure that the robot speed along the path is modulated as little as possible. In other words, it aims to guarantee that the final robot trajectory will be as close as possible to the desired one $\boldsymbol{x}_{des}(t)$. This means that the scaling factor α should ideally remain equal to its desired maximum value, i.e. $\alpha_{max} = 1$. Furthermore, it is in general necessary to penalize unnecessary movements, e.g. at the end of the path a movement of the robot in the null space is undesired. This means that joint speed must be penalized. Thus the cost function can be expressed as the sum of these two costs:

$$\frac{1}{2}\int_{t_i}^{t_f} \|1 - \alpha(\tau)\|^2_{\boldsymbol{W}_\alpha} + \|\boldsymbol{u}(\tau)\|^2_{\boldsymbol{W}_{\boldsymbol{u}}} \mathrm{d}\tau, \tag{14}$$

where $\boldsymbol{W}_\alpha$ and $\boldsymbol{W}_{\boldsymbol{u}}$ represent two weighting factors that are used to penalize the scaling factor error and the joint speed, respectively. In this proposed formulation, $\boldsymbol{W}_\alpha >> \boldsymbol{W}_{\boldsymbol{u}}$ to prioritize the path following.

E. Final Problem

Plugging all the constraints and the cost function together, it is possible to obtain the following control problem:

$$
\begin{aligned}
\min_{\boldsymbol{u}(\cdot),\alpha(\cdot)} \quad & \frac{1}{2}\int_{t_i}^{t_f} \|1-\alpha(\tau)\|^2_{\boldsymbol{W}_\alpha} + \|\boldsymbol{u}(\tau)\|^2_{\boldsymbol{W}_{\boldsymbol{u}}} \mathrm{d}\tau, \\
s.t. \quad & \boldsymbol{J}_T(\boldsymbol{q}(t))\boldsymbol{u}(t) = \boldsymbol{x}'_{des}(s(t))\alpha(t)\dot{\bar{s}}, && \forall t \in [t_i, t_f], \\
& \boldsymbol{J}_{r_i}(\boldsymbol{q}_{p_i}(t))\boldsymbol{u}_{p_i}(t) \leq v_{max_i}(t) && \forall i \in \{1,\dots,n\}, \\
& && \forall t \in [t_i, t_f], \\
& \boldsymbol{q}_{min} \leq \boldsymbol{q}(t) \leq \boldsymbol{q}_{max} && \forall t \in [t_i, t_f], \\
& \dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}}(t) \leq \dot{\boldsymbol{q}}_{max} && \forall t \in [t_i, t_f], \\
& 0 \leq s(t) \leq s_{max} && \forall t \in [t_i, t_f], \\
& s(t) = \int_{t_i}^{t} \alpha(\tau)\dot{\bar{s}}\mathrm{d}\tau && \forall t \in [t_i, t_f], \\
& \boldsymbol{q}(t) = \int_{t_i}^{t} \boldsymbol{u}(\tau)\mathrm{d}\tau && \forall t \in [t_i, t_f], \\
& 0 \leq \alpha(t) \leq 1 && \forall t \in [t_i, t_f].
\end{aligned}
\tag{15}
$$

Where the last three constraints represent the evolution of the system, modeled as (1), the evolution of the curvilinear abscissa, and the limit on the scaling factor α. Since the optimization variables are both $\boldsymbol{u}$ and α, the robot will be free to adjust the joint configuration to keep $\alpha = 1$, implicitly exploiting the redundancies of the system according to (3) and (10).

Even if highly nonlinear, the optimization problem (15) has always a feasible solution. When the human operator hinders the end-effector in such a way that the task can not be fulfilled at the desired speed, the trajectory in the task space is scaled until reaching (in the worst case) a null speed, i.e. $\alpha = 0$ and $\boldsymbol{u} = 0$. The guarantee stands because the model in equation (1) does not take into account the acceleration limits. When the human operator, instead, hinders the body of the robot, e.g. the elbow, the optimization problem exploits the redundancy of the robot to avoid undesired speed reduction.

The main drawback of (15) is its high complexity. Due to the nonlinearities, solving the optimal control problem over the entire time horizon, from t_i to t_f, is a computationally demanding and it is difficult to achieve in real time for online control. Nonetheless, it is possible to achieve good performance by solving it over a shorter receding horizon, by leveraging the MPC formulation [13]. It is worth to underline that, differently from the approach in [13], the proposed optimization problem in equation (15) forces the robot end-effector to stay on the path thanks to the first constraint. Furthermore, thanks to how it is formalized, the optimization problem can be easily extended to the execution of other tasks with lower priority exploiting the HQP formulation [14].

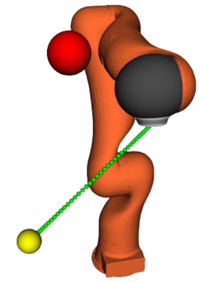


Fig. 1: Simulated environment. The red sphere represents the human, the yellow sphere represents the final goal, and the green spheres represent the path.

IV. Experimental Validation

This section is dedicated to demonstrating the effectiveness of the optimization problem presented in (15).

Firstly, the approach is validated in a simulated environment, highlighting the advantages of an extended control horizon compared to local optimization even in situation where the human operator is not moving. This improvement is intuitive, as the MPC approach enables the robot to proactively address situations where it might approach the human operator, with a consequent speed reduction for safety reasons, and joint limits. By anticipating and avoiding such speed reductions, the overall performance is enhanced and the execution time is reduced.

Secondly, the approach undergoes validation in a real-world experiment where the robot has to linearly push a reel, simulating a machine-tending scenario. This validation demonstrates the effectiveness of the presented strategy and its practical applicability, even in real settings where a human operator is present, potentially hindering task execution.

A. Simulated Environment

In this experiment, a simulated Kuka LWR4+, a 7 DoFs collaborative robot, has to follow a linear trajectory with the end-effector, while the orientation is free to change and is not constrained during the task execution. The initial robot configuration id $\boldsymbol{q} = \{0, 0.5, 0, 2.07, 0, -1.3, 0\}$ while the desired linear movement is $\Delta\boldsymbol{x}_{des} = \{0.15, -0.3, -0.15\}$. Starting from this, it is possible to compute a trajectory $\boldsymbol{x}_{des}(t)$ as a third-order polynomial trajectory. Lastly, for simplicity, the human operator is approximated as a sphere in the scene whose position is constant and equal to $P_h = \{0.06, -0.15, 0.7\}$. The overall scene is represented in Fig. 1.

The simulation time has been chosen equal to $T_s = 2\ ms$ and the simulation experiments have been solved by exploiting the Matlab fmincon solver. To illustrate the effectiveness and the advantages of the MPC approach, two distinct cases are examined:

- Control and prediction horizon equal to $H = 1$.
- Control and prediction horizon equal to $H = 5$.

The results are illustrated in Fig. 2, where it is possible to note that choosing a low prediction horizon, i.e. $H = 1$, the robotic system reaches a minimum and it is not able to reach

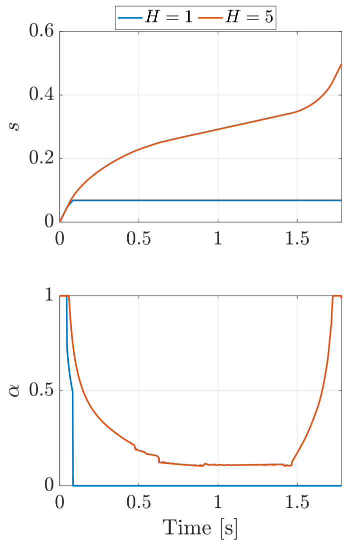
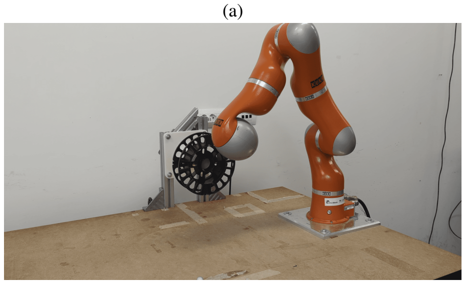


Fig. 2: Evolution of s and α for different horizons.

(a)



(b)

Fig. 3: Setup of the experiments. a) Reel insertion performed by a UR10e collaborative robot. b) Reproduction of the machine tending task.

the end of the path. By using a greater horizon, the system can predict this issue and complete the overall task.

B. Real Experiment

In the real experiment, the Kuka LWR4+ has to perform a machine-tending task, i.e. it has to linearly insert a reel that must be changed. This task has taken inspiration from the IMA use case of the ROSSINI project[1], which is illustrated in Fig. 3a. To perform the insertion phase, the task has been reproduced in a laboratory setting, as illustrated in Fig. 3b.

The tracking of the human operator[2] has been performed exploiting seven OptiTrack Prime^{X} cameras along with the Motive software. The OptiTrack has been also exploited to estimate the human velocity. Starting from this information it is possible to predict the future human operator position with the method proposed in [21]. These tracking components were integrated into the system, which was developed using ROS Melodic Morenia meta-operating system and executed on a Intel(R) Core(TM) i7-10510U with Ubuntu 18.04 operating system. The optimization problem in (15) is solved by exploiting acados framework [23] with the SQP-RTI algorithm [24].

Concerning the frequencies, the communication with the Kuka LWR4+ works at $500\ Hz$, while the OptiTrack works at a frequency of $240\ Hz$. The optimization problem convergence is strictly related to the prediction horizon selected. With a prediction horizon of 8, it is solved in approximately $10\ ms$ allowing it to directly control the system. It is worth underlining that by appropriately adjusting the robot reaction time T_r in equation (6), it is possible to ensure that the overall robot behaviour remains safe.

To validate and show the effectiveness for the real scenario, these experiments have been performed:

- Insertion of the reel without the presence of the human operator. This experiment provides insight into how the robot would behave without any safety constraints when inserting the reel.
- Insertion of the reel while the human operator hinders the robot elbow. This experiment demonstrates that in the presence of the human operator, the robot exploits the redundancies to ensure the correct execution of the task.
- Insertion of the reel with the approach proposed in [12] while the human operator hinders the robot elbow. This experiment is used to compare the proposed approach with our previous work.
- Insertion of the reel while the human operator hinders the end-effector. This demonstrates that, in the worst case, the proposed approach behaves similarly to the scaling algorithm proposed in [12].

Furthermore, the results can also be appreciated in the accompanying video.

In the first experiment, since the human operator is not inside the scene, the speed of the robot is never modulated, i.e. $\alpha = 1$. A single snapshot of this experiment, that can be used as a comparison for the others, is illustrated in Fig. 4a.

The second experiment, instead, illustrated how the proposed approach behaves when the human operator hinders the robot. In particular, during the initial phase, the human operator approaches the robot elbow. As expected, thanks to the optimization problem formulated in (15) the robot is still able to perform the task while exploiting the redundancy to increase the distance between the elbow and the human operator. This is illustrated in Fig. 4b, where it is important to note the different robot configuration w.r.t. Fig. 4a. Furthermore, Fig. 5a shows the evolution of the end-effector position during the experiment, demonstrating that the performed path is equal to the planned one, while Fig.

[1]`https://www.youtube.com/watch?v=o3vp4j14GFs`

[2]For the sake of simplicity, the human operator is approximated only with a capsule embedding the arm.

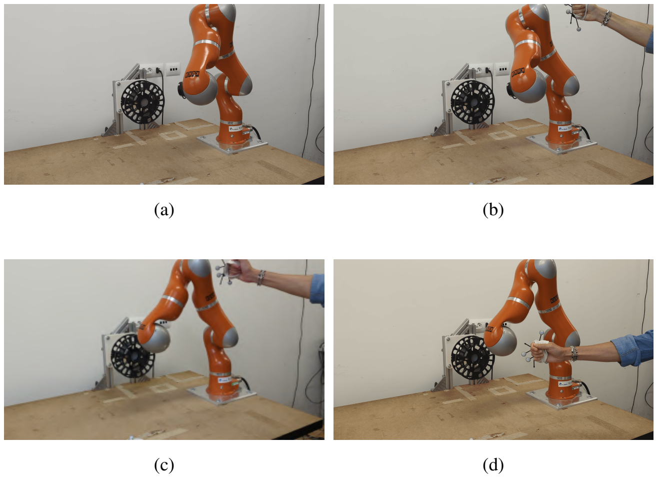


Fig. 4: Snapshots of the experiments. a) First experiment. Robot configuration when the human is not in the scene. b) Second experiment. Robot configuration when the human hinders the elbow. c) Third experiment. The robot stops with the approach proposed in [12]. d) Fourth experiment. The robot stops when the human hinders the end-effector.

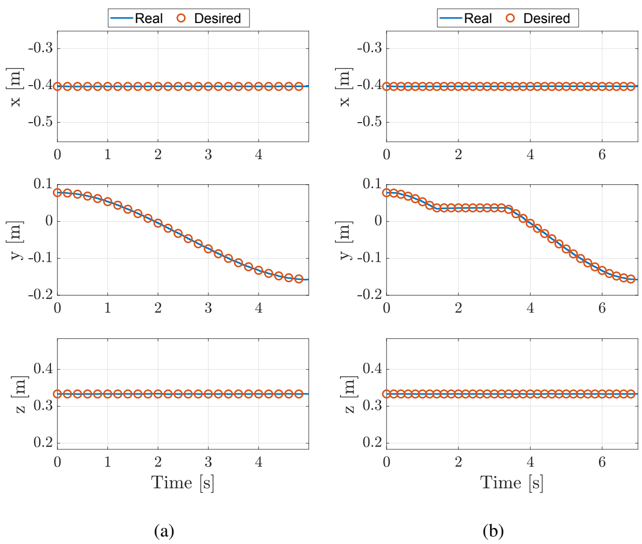


Fig. 5: End-effector position during the second and fourth experiment, respectively.

6a shows that the scaling factor is always equal to one while adhering to (6).

In the third experiment, the control problem is solved with the optimization problem proposed in [12]. Exploiting that architecture, all the robot velocities are modulated to ensure safety. Since the human operator is very close to the robot elbow, the safety requirement imposed by the ISO/TS 15066 constraint imposes that the robot can not move towards the human operator, i.e. $v_{max} = 0$. Thus, the only admissible solution is to stop the robot, as shown in Fig. 4c. A quantitative comparison in that case would not be fair, since the stopping time would depend on how much time the human operator hinders the robot.

In the last experiment, the human operator hinders both the end-effector and the elbow. In particular, when the end-

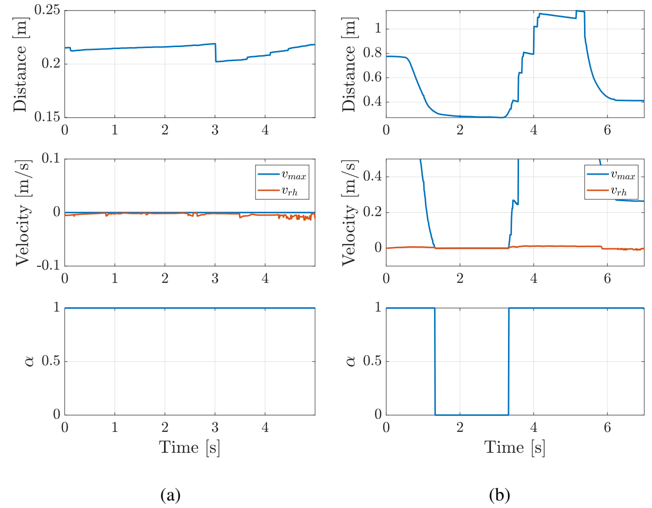


Fig. 6: a) Data regarding the ISO/TS 15066 constraint for the second and fourth experiment, respectively. Distance between the human operator and the most interesting link, i.e. elbow for the second and end-effector for the fourth, maximum speed according to the ISO/TS15066, and the scaling factor computed.

effector is approaching the human operator the solution computed by the solver is $\alpha = 0$ and $\boldsymbol{u} = \mathbf{0}$, stopping the robot. This is illustrated in Fig. 5b, where the end-effector position remains equal from $T = 1.2\ s$ to $T = 3.4\ s$, and in Fig. 6b. Fig. 4d, instead, shows the real robot stopped. It is important to underline that, as demonstrated in the previous experiment, the proposed strategy can achieve the same performances that would be achieved by exploiting the framework outlined in [12]. Thus, the proposed approach is an extension of [12] that achieves the same performances when the human operator obstructs the planned task path but it drastically reduces the execution time when the operator approaches the robot body.

V. Conclusions and Future Works

Starting from the work done in [12], in this paper, a novel nonlinear control problem is proposed. The goal of the control problem is to efficiently handle the safety constraint imposed by the ISO/TS 15066, minimizing the execution time, while ensuring that the performed end-effector path is not changed. Due to the high computational complexity, the problem is solved by exploiting a model predictive control formulation, achieving good performances. The proposed strategy has been validated both in a simulated and in a real scenario, proving its effectiveness and adaptability.

Future works will aim at exploiting how the proposed approach behaves when adding other constraints with lower priority, such as increasing the distance to the joint limits. Furthermore, it would be possible to embed the work in a more complex architecture and use the result of this optimization problem as a reference for a faster low-level controller. Then, the formulation can be extended to consider also the PFL constraint and not only the SSM constraint.

REFERENCES

[1] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[2] *ISO 10218-1:2011(E). Robots and Robotic Devices–Safety Requirements for Industrial Robots–Part 1: Robots*, International Organization for Standardization Std., Jul. 2011.

[3] *ISO 10218-2:2011(E). Robots and Robotic Devices–Safety Requirements for Industrial Robots–Part 2: Robot systems and integration*, International Organization for Standardization Std., Jul. 2011.

[4] *ISO/TS 15066:2016(E). Robots and robotic devices–Collaborative robots*, International Organization for Standardization Technical Specification, Feb. 2016.

[5] A. Levratti, G. Riggio, C. Fantuzzi, A. De Vuono, and C. Secchi, "Tirebot: A collaborative robot for the tire workshop," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 129–137, 2019.

[6] J.-H. Chen and K.-T. Song, "Collision-free motion planning for human-robot collaborative safety under cartesian constraint," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[7] K. Merckaert, B. Convens, C.-j. Wu, A. Roncone, M. M. Nicotra, and B. Vanderborght, "Real-time motion control of robotic manipulators for safe human–robot coexistence," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102223, 2022.

[8] F. Ferraguti, M. Bertuletti, C. T. Landi, M. Bonfè, C. Fantuzzi, and C. Secchi, "A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5921–5928, 2020.

[9] F. Ferraguti, C. Talignani Landi, A. Singletary, H.-C. Lin, A. Ames, C. Secchi, and M. Bonfe, "Safety and efficiency in robotics: The control barrier functions approach," *IEEE Robotics & Automation Magazine*, pp. 2–14, 2022.

[10] A. Oleinikov, S. Kusdavletov, A. Shintemirov, and M. Rubagotti, "Safety-aware nonlinear model predictive control for physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5665–5672, 2021.

[11] N. Lucci, B. Lacevic, A. M. Zanchettin, and P. Rocco, "Combining speed and separation monitoring with power and force limiting for safe collaborative robotics applications," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6121–6128, 2020.

[12] A. Pupa, M. Arrfou, G. Andreoni, and C. Secchi, "A safety-aware kinodynamic architecture for human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4465–4471, 2021.

[13] M. Faroni, M. Beschi, and N. Pedrocchi, "An mpc framework for online motion planning in human-robot collaborative tasks," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 1555–1558.

[14] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[15] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[16] A. Pupa, P. Robuffo Giordano, and C. Secchi, "Optimal energy tank initialization for minimum sensitivity to model uncertainties (forthcoming)," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.

[17] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[18] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, pp. 27–39, 2008.

[19] F. Ferraguti, C. T. Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," *Robotics and Autonomous Systems*, vol. 124, p. 103388, 2020.

[20] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 338–345.

[21] C. T. Landi, Y. Cheng, F. Ferraguti, M. Bonfè, C. Secchi, and M. Tomizuka, "Prediction of human arm target for robot reaching movements," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5950–5957.

[22] Y. Wang, X. Ye, Y. Yang, and W. Zhang, "Collision-free trajectory planning in human-robot interaction through hand movement prediction from vision," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 305–310.

[23] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.

[24] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.