



HAL
open science

Online optimal motion generation with guaranteed safety in shared workspace

Pu Zheng, Pierre-Brice Wieber, Olivier Aycard

► **To cite this version:**

Pu Zheng, Pierre-Brice Wieber, Olivier Aycard. Online optimal motion generation with guaranteed safety in shared workspace. ICRA 2020 - International Conference on Robotics and Automation, May 2020, Paris, France. pp.1-6. hal-02496057

HAL Id: hal-02496057

<https://hal.science/hal-02496057>

Submitted on 2 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online optimal motion generation with guaranteed safety in shared workspace

Pu Zheng¹, Pierre-Brice Wieber², Olivier Aycard¹

Abstract—With new, safer manipulator robots, the probability of serious injury due to collisions with humans remains low (5%), even at speeds as high as 2 m.s^{-1} . Collisions would better be avoided nevertheless, because they disrupt the tasks of both the robot and the human. We propose in this paper to equip robots with exteroceptive sensors and online motion generation so that the robot is able to perceive and react to the motion of the human in order to reduce the occurrence of collisions. It's impossible to guarantee that no collision will ever take place in a partially unknown dynamic environment such as a shared workspace, but we can guarantee instead that, if a collision takes place, the robot is at rest at the time of collision, so that it doesn't inject its own kinetic energy in the collision. To do so, we adapt a Model Predictive Control scheme which has been demonstrated previously with two industrial manipulator robots avoiding collisions while sharing their workspace. The proposed control scheme is validated in simulation.

I. INTRODUCTION

New, safer manipulator robots can share their workspace with humans, thanks to advanced mechanical and control design that make sure that potential collisions between the robot and humans result in a low (5%) probability of serious injury even at speeds as high as 2 m.s^{-1} [1]. Collisions would better be avoided nevertheless, not only because of the remaining risk, however low it is, but also because each collision disrupts the tasks of both the robot and the human. We propose in this paper to equip such robots with exteroceptive sensors and online motion generation so that the robot is able to perceive and react to the motion of the human in order to reduce the occurrence of collisions and avoid disrupting their respective tasks. We don't consider here physical interactions between the robot and human, such as a handing an object or holding it together, only situations where they share the same workspace and have to work separately with as little interference as possible.

Collision avoidance is classically implemented by restricting the motion of the robot in the direction of the human, either with a repulsive potential field [2], damping [3], momentum limitations [4], control barrier functions [5], invariance control [6]. All these approaches share two major problems. The first is that these restrictions are defined arbitrarily, only loosely related to the dynamics of the robot, and generate therefore suboptimal behaviors such as unnecessary

detours or stops. The second is that they assume that the proposed restriction can always be enforced and the collision avoided, which may not be true when multiple constraints are affecting simultaneously the motion of the robot. In that case, their behavior is undefined and a collision could happen in a completely uncontrolled way. It's impossible to guarantee that no collision will ever take place in a partially unknown dynamic environment such as a shared workspace, but we can guarantee instead that, if a collision takes place, the robot is at rest at the time of collision, so that it doesn't inject its own kinetic energy in the collision [7]. This is called *passive motion safety*, and this is what we aim for.

In this paper, we adapt a Model Predictive Control (MPC) scheme which has been demonstrated previously with two industrial manipulator robots avoiding collisions while sharing their workspace [8]. But passive motion safety wasn't enforced. We propose here to do so by imposing as a terminal constraint that the robot is always able to stop in the end of the prediction horizon [9]. This corresponds to making sure that you can always stop within the limits of where you can see when driving in a fog. Perception and prediction of the human motion are naturally crucial and complex issues in this case. They need a discussion on their own, so we suppose in this paper that perception of the human motion is provided.

Concerning the prediction of human motion, we need conservative, worst-case estimates for effective collision avoidance. A manipulator robot at full speed (2 m.s^{-1}) can stop in approximately 0.15 s and 0.15 m [10]. In this amount of time, a human hand starting from rest can cover more than 1 m for extreme motions such as boxing [11]. So, if the robot was asked to stop before being punched by a human, the necessary safety distance would be more than 1.15 m, which would strongly limit the possibility to share workspaces. However, human reaching motions are typically 10 times slower, with maximum speed approximately 1 m.s^{-1} and acceleration 5 m.s^{-2} [12]. We propose to consider these slower estimates to design the collision avoidance and passive motion safety capacities of the robot. If the human happens to move faster and a collision happens before the robot can stop, we would still benefit from the low probability of serious injury discussed above [1].

With respect to the ISO Technical Specification 15066 on collaborative robots, our approach combines aspects of *speed and separation monitoring* when actively avoiding collisions, with aspects of *power and force limiting* if a collision eventually occurs. The focus of this paper is on active collision avoidance, while we rely on the manipulator

This work was supported by IDEX university Grenoble Alpes

¹ Pu Zheng and Olivier Aycard are with Univ. Grenoble Alpes, LIG, 38000 Grenoble, France
pu.zheng@univ-grenoble-alpes.fr
olivier.aycard@univ-grenoble-alpes.fr

² Pierre-Brice Wieber is with Univ. Grenoble Alpes, Inria, 38000 Grenoble, France pierre-brice.wieber@inria.fr

robot's mechanical and control design to lower the risk of injury in case of collision. This latter aspect could be handled more directly as in [13].

The classical approach to monitor the distance between a robot and its environment is with the Gilbert–Johnson–Keerthi distance algorithm, which provides a pair of closest points between two objects [14]. But monitoring only a pair of closest points can lead to catastrophic failures (collisions), since closest points can change abruptly when objects move [15]. We propose instead to use the existence of a separating plane between two objects as evidence that they don't collide, as this is immune to changes of closest points [16]. We can actually impose the existence of separating planes over whole segments of motion of the two objects instead of individual positions. This provides continuous-time trajectory safety over whole intervals of time at very low computational cost [17].

This paper is organized as follows. We present the general MPC approach in Section II and emphasize the role of the terminal constraint to guarantee safety. Collision avoidance using separating planes is detailed in Section III. Simulations of the proposed method on a two-link planar robot are presented in Section IV. Conclusion and future work are discussed in Section V.

II. A MODEL PREDICTIVE CONTROL APPROACH

Consider the motion of an n link serial manipulator with joint position $q \in \mathbb{R}^n$ and piecewise constant acceleration over periods of time Δt , represented as a linear discrete time system

$$x_{k+1} = Ax_k + Bu_k \quad (1)$$

where

$$x_k = \begin{bmatrix} q_k \\ \dot{q}_k \end{bmatrix} \in \mathbb{R}^{2n}, \quad (2)$$

$$A = \begin{bmatrix} I & I\Delta t \\ 0 & I \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \quad (3)$$

$$B = \begin{bmatrix} I\frac{\Delta t^2}{2} \\ I\Delta t \end{bmatrix} \in \mathbb{R}^{2n \times n} \quad (4)$$

with I an $n \times n$ identity matrix and $u_k = \ddot{q}_k$ the piecewise constant acceleration.

The idea of Model Predictive Control is to solve at each sampling time an optimal control problem over a finite horizon starting at the current state x_0 . Consider a control sequence $\{u_0, u_1, \dots, u_{N-1}\}$ of length N . A recursive application of the linear discrete time system (1) provides the resulting sequence of states $\{x_1, \dots, x_N\}$:

$$\underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} A \\ \vdots \\ A^N \end{bmatrix}}_{P_x} x_0 + \underbrace{\begin{bmatrix} B & 0 & 0 \\ \vdots & \ddots & 0 \\ A^{N-1}B & \dots & B \end{bmatrix}}_{P_u} \underbrace{\begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{\mathbf{u}}. \quad (5)$$

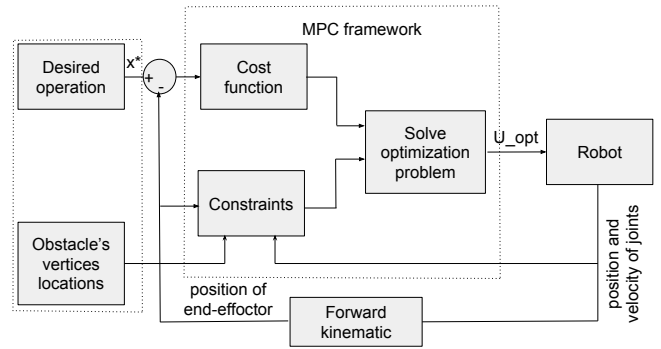


Fig. 1. Control scheme. The first bloc gives the data about the desired task and obstacle's vertices location, then this information plus robot's data allows to formulate the cost function and constraint for the second MPC bloc.

The objective of the MPC scheme is to

$$\underset{\mathbf{u}}{\text{minimize}} \quad \sum_{k=0}^{N-1} \|x_{k+1} - x_{k+1}^{des}\|_Q^2 + \|u_k - u_k^{des}\|_R^2 \quad (6a)$$

$$\text{subject to} \quad \forall k \in \{0, \dots, N-1\}, \quad \underline{u} \leq u_k \leq \bar{u}, \quad (6b)$$

$$\forall k \in \{1, \dots, N\}, \quad \underline{q} \leq q_k \leq \bar{q}, \quad (6c)$$

$$\forall k \in \{1, \dots, N-1\}, \quad \underline{\dot{q}} \leq \dot{q}_k \leq \bar{\dot{q}}, \quad (6d)$$

$$\dot{q}_N = 0 \quad (6e)$$

with \underline{q} , \bar{q} , $\underline{\dot{q}}$, $\bar{\dot{q}}$, \underline{u} , \bar{u} indicating minimum and maximum joint position, speed and acceleration (we assume that $\underline{\dot{q}} \leq 0 \leq \bar{\dot{q}}$ and $\underline{u} \leq 0 \leq \bar{u}$). Here, the objective is to track a desired joint state trajectory x_k^{des} with acceleration u_k^{des} , so this optimal control problem takes the form of a simple linearly constrained Quadratic Program (QP) which can be solved efficiently with off-the-shelf solvers. A straightforward alternative is to consider a desired cartesian motion of the end effector of the robot, using its forward kinematic model, but that makes the problem non-linear. In the end, the robot follows the obtained optimal control sequence until the next sampling time, and a new optimal control problem is solved.

A key element of this approach is the terminal constraint (6e). It may seem unnecessarily constraining the motion of the robot, but it provides recursive feasibility, guaranteeing that when the optimal control sequence is applied to the robot, it always leads to new states of the robot where the optimal control problem (6) is once again feasible. This allows guaranteeing that all constraints in (6) will always be satisfied [18], which is crucial. More precisely, this terminal constraint imposes that the robot is at rest at the end of the prediction horizon. When collision avoidance is introduced in this MPC scheme, this is what provides the passive motion safety guarantee, making sure that the robot is able to stop and stay at rest before any collision happens in the future. This is a central aspect of the approach we propose.

III. CONTINUOUS COLLISION AVOIDANCE

We represent the different parts of the robot and person as polyhedra with vertices $\{r^1, r^2, \dots\}$ and $\{p^1, p^2, \dots\}$

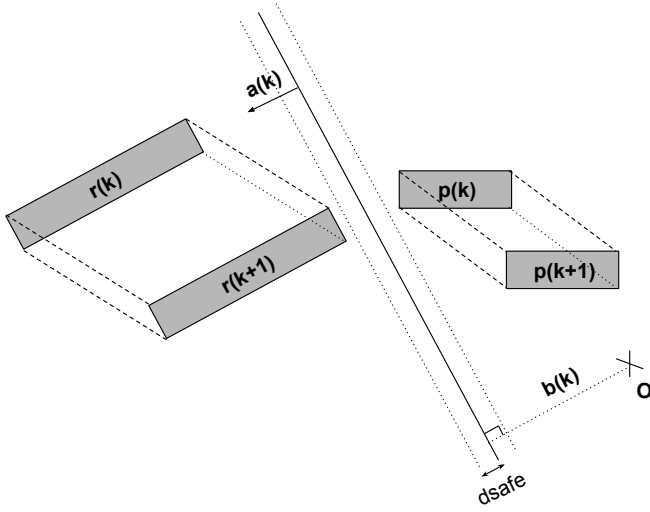


Fig. 2. If there exists a plane such that all vertices r^i of the robot stay on one side of the plane and all vertices p^j of the person stay on the other side between instants k and $k + 1$, we have evidence that they don't collide over this interval of time.

(respectively). As illustrated in Fig. 2, if there exists a plane defined by a normal vector $a_k \in \mathbb{R}^3$ and a scalar constant $b_k \in \mathbb{R}$ such that all vertices $p^j \in \mathbb{R}^3$ of the person stay on one side between instants k and $k + 1$:

$$a_k^T p_k^j \leq b_k, \quad (7a)$$

$$a_k^T p_{k+1}^j \leq b_k, \quad (7b)$$

while all vertices $r^i \in \mathbb{R}^3$ of the robot stay on the other side, with some additional distance d :

$$a_k^T r_k^i \geq b_k + d, \quad (8a)$$

$$a_k^T r_{k+1}^i \geq b_k + d, \quad (8b)$$

then we have evidence that they don't collide over this interval of time [17]. These constraints (for all vertices p^j and r^i) with decision variables a_k and b_k can be directly added for all $k \in \{0, \dots, N-1\}$ to the MPC scheme (6), computing simultaneously the separating planes and the corresponding collision-free trajectory of the robot. A computationally more efficient approach is to first compute separating planes with the robot trajectory obtained at the previous sampling time, and then, to add only the constraints (8) to the MPC scheme (6) with the separating planes fixed [16].

1) *Compute separating planes:* Given the robot trajectory computed at the previous sampling time and the current human motion prediction, we can find planes that maximize the distance d :

$$\underset{a_k, b_k, d}{\text{maximize}} \quad d \quad (9)$$

subject to constraints (7) and (8). If the current human motion happens to outpace the one previously predicted, it might not be possible to satisfy the constraints (8) with the previously computed robot trajectory and a positive distance d . In that case, we temporarily accept a negative "distance", finding the plane (a_k, b_k) which is closest to separating the previously computed robot trajectory from the current human

motion prediction. When this plane is used to compute a new collision-free robot trajectory in the next step, a fixed positive safety distance d_{safe} is enforced anew. Note that the constraints (7) and (8) are linear with respect to a_k , b_k and d . For the maximization problem to be well-posed, we also need to bound the vector a_k to a unit norm ($\|a_k\| \leq 1$). This is a nonlinear constraint, but it can be approximated efficiently with linear constraints [16]:

$$-\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \leq a_k \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (10)$$

$$1 - \varepsilon \leq a_k^T a_k^p \leq 1 \quad (11)$$

where a_k^p is the vector a_k computed at the previous sampling time and ε is a small positive constant. This way, the maximization (9) turns out to be a standard Linear Program. We found preferable, however, to smooth the variations of separating planes, considering instead a regularized objective

$$\underset{a_k, b_k, d}{\text{minimize}} \quad -d + \alpha d^2 + \beta \|a_k - a_k^p\|^2 + \beta |b_k - b_k^p|^2 \quad (12)$$

with a_k^p , b_k^p the separating plane obtained at the previous sampling time and small weights α and β . This results in a linearly constrained QP, solved efficiently with off-the-shelf solvers.

2) *Compute a collision-free trajectory:* We can compute now a collision-free robot trajectory with the separating planes computed in the previous step. In the constraints (8), the positions of all vertices r^i depend on the kinematics of the robot, which is usually a nonlinear function of the joint position q . We propose to linearize the kinematics of the robot around the trajectory q_k^p computed at the previous sampling time:

$$\forall k \in \{1, \dots, N\}, \quad r_k^i = r^i(q_k) \approx r^i(q_k^p) + J(q_k^p)(q_k - q_k^p) \quad (13)$$

with $J(q_k^p)$ the Jacobian of the kinematics of the robot. This way, the constraints (8) are turned into linear functions of q and the MPC scheme (6) with these additional constraints is kept in the form of a linearly constrained QP, solved efficiently with off-the-shelf solvers.

If we can't find a trajectory satisfying these constraints because the current human motion has outpaced the collision avoidance capacities of the robot, we can continue with the trajectory obtained at the previous sampling time. It was designed to stop gracefully before any collision happens, due to the terminal constraint (6e), at least with what could be predicted of the human motion. As long as our prediction is valid, this allows to enforce passive motion safety. If our prediction happens to be invalid because the human made a completely unexpectedly fast motion, we can still count on the intrinsic mechanical safety of the robot, with low (5%) probability of serious injury even at speeds as high as 2 m.s^{-1} [1]

To summarize, for each time interval $[k, k + 1]$, $k \in$

$\{0, \dots, N-1\}$, we compute separating planes with QPs:

$$\min_{a_k, b_k, d} -d + \alpha d^2 + \beta \|a_k - a'_k\|^2 + \beta \|b_k - b'_k\|^2 \quad (14a)$$

$$\text{s.t. } \forall j, a_k^T p_k^j \leq b_k, \quad (14b)$$

$$\forall j, a_k^T p_{k+1}^j \leq b_k, \quad (14c)$$

$$\forall i, a_k^T r_k^i \geq b_k + d, \quad (14d)$$

$$\forall i, a_k^T r_{k+1}^i \geq b_k + d, \quad (14e)$$

$$-\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \leq a_k \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (14f)$$

$$1 - \varepsilon \leq a_k^T a_k^p \leq 1 \quad (14g)$$

Then, we use these separating planes to compute the collision-free robot trajectory with a QP:

$$\min_{\mathbf{u}} \sum_{k=0}^{N-1} \|x_{k+1} - x_{k+1}^{des}\|_Q^2 + \|u_k - u_k^{des}\|_R^2 \quad (15a)$$

$$\text{s.t. } \forall k \in \{0, \dots, N-1\}, \underline{u} \leq u_k \leq \bar{u}, \quad (15b)$$

$$\forall k \in \{1, \dots, N\}, \underline{q} \leq q_k \leq \bar{q}, \quad (15c)$$

$$\forall k \in \{1, \dots, N-1\}, \underline{\dot{q}} \leq \dot{q}_k \leq \bar{\dot{q}}, \quad (15d)$$

$$\dot{q}_N = 0, \quad (15e)$$

$$\forall k \in \{0, \dots, N-1\}, \forall i,$$

$$a_k^T r^i(q_k^p) + a_k^T J(q_k^p)(q_k - q_k^p) \geq b_k + d_{safe}, \quad (15f)$$

$$\forall k \in \{0, \dots, N-1\}, \forall i,$$

$$a_k^T r^i(q_{k+1}^p) + a_k^T J(q_{k+1}^p)(q_{k+1} - q_{k+1}^p) \geq b_k + d_{safe} \quad (15g)$$

where the sequence of states (q_k, \dot{q}_k) is linearly related to the control sequence \mathbf{u} through (5).

IV. SIMULATION RESULTS

The proposed MPC scheme is implemented on a 7-DoF manipulator robot, simulated with ROS. Maximum joint speed and acceleration are respectively $\frac{\pi}{2}$ rad.s⁻¹ and 10 rad.s⁻². The terminal constraint (6e) imposes that the robot is at rest at the end of the prediction horizon in order to provide a passive motion safety guarantee. This manipulator robot at full speed needs approximately 0.15 s to stop. The prediction horizon must therefore be longer than that, but it can't be much longer since the prediction of the human motion quickly becomes imprecise [19], [20]. We opt for a prediction horizon of length 0.25 s, with $\Delta t = 0.05$ s and $N = 5$. A human arm is also modeled, as a 10 cm large, 50 cm long cylinder with a safety distance $d_{safe} = 20$ cm. The resulting QPs (14) and (15) are solved with qpOASES [21], with computation times that easily allow a sampling frequency above 100 Hz (see Table I).

The original MPC scheme (6) with nonlinear kinematics of the robot and nonlinear separating planes constraints (7)-(8) could be solved by iterating the QPs (14) and (15), following a Sequential QP approach. We can see in Fig. 3 that this typically converges in few iterations, with the second iteration and subsequent ones making corrections of

	Separating Plane QP (14)	Collision-free trajectory QP (15)
Number of variables	5	35
Number of constraints	21	86
Computation time	0.7 ms	6.1 ms

TABLE I
COMPUTATION TIME.

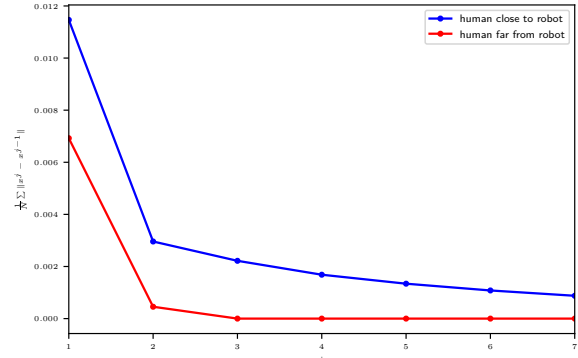


Fig. 3. Corrections (in meters) in end-effector position when iterating QPs (14) and (15).

only a few millimeters. This shows that the first iteration already provides perfectly good approximate solutions. We will proceed therefore with only one cycle of QPs (14) and (15) at each sampling period.

In Fig. 4 (a), two green spheres represent alternating desired positions for the end-effector of the manipulator robot. In Fig. 4 (b), a human arm covers the left-most goal, blocking the motion of the robot. The robot maintains as a result a fixed position at the prescribed safety distance $d_{safe} = 20$ cm. The corresponding separating plane is represented in dark transparent gray. Once the human arm moves away, the robot resumes its motion and finally reaches its goal (Fig. 4 (c)).

When the human arm is not interfering, the robot can alternate between the two desired positions at a horizontal speed up to 1 m.s⁻¹ as shown in the first 4 seconds in Fig. 6, and in Fig. 5 (a) and (b). When the human arm comes in the way (Fig. 5 (c) and (d)), the robot automatically adapts its motion to keep executing its task, alternating between the two desired positions while maintaining the prescribed safety distance $d_{safe} = 20$ cm above the human arm. This introduces some vertical motion of the end-effector and results in slower horizontal motion (seconds 4 to 9 in Fig. 6). When the human arm moves away, the robot resumes its full speed (Fig. 5 (e) and (f)).

V. CONCLUSION

This paper introduces a Model Predictive Control scheme to generate online optimal collision-free trajectories in a workspace shared with a human. The collision avoidance between the robot and the human is ensured by the existence

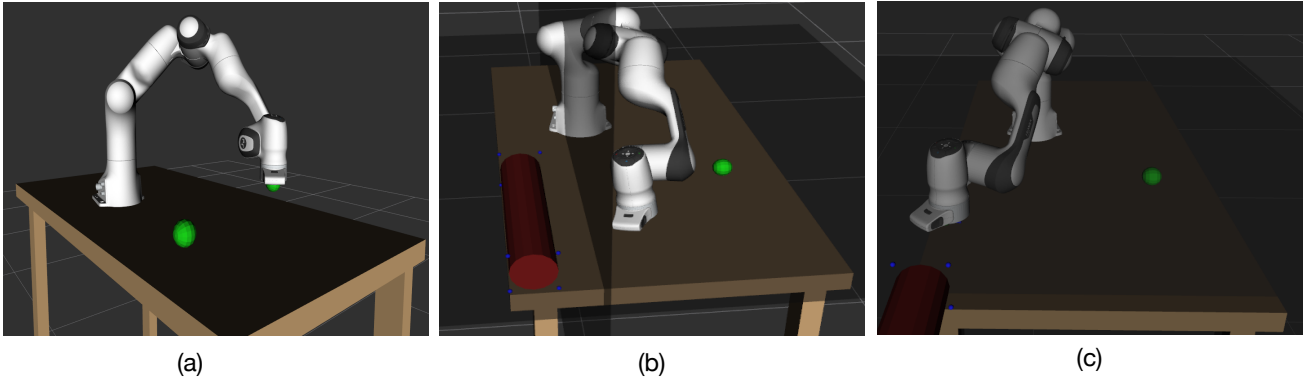


Fig. 4. (a) Two green spheres represent alternating desired positions for the end-effector of the manipulator robot. (b) A human arm covers the left-most goal, blocking the motion of the robot. The robot maintains as a result a fixed position at the prescribed safety distance $d_{safe} = 20$ cm. The corresponding separating plane is represented in dark transparent gray. (c) Once the human arm moves away, the robot resumes its motion and finally reaches its goal.

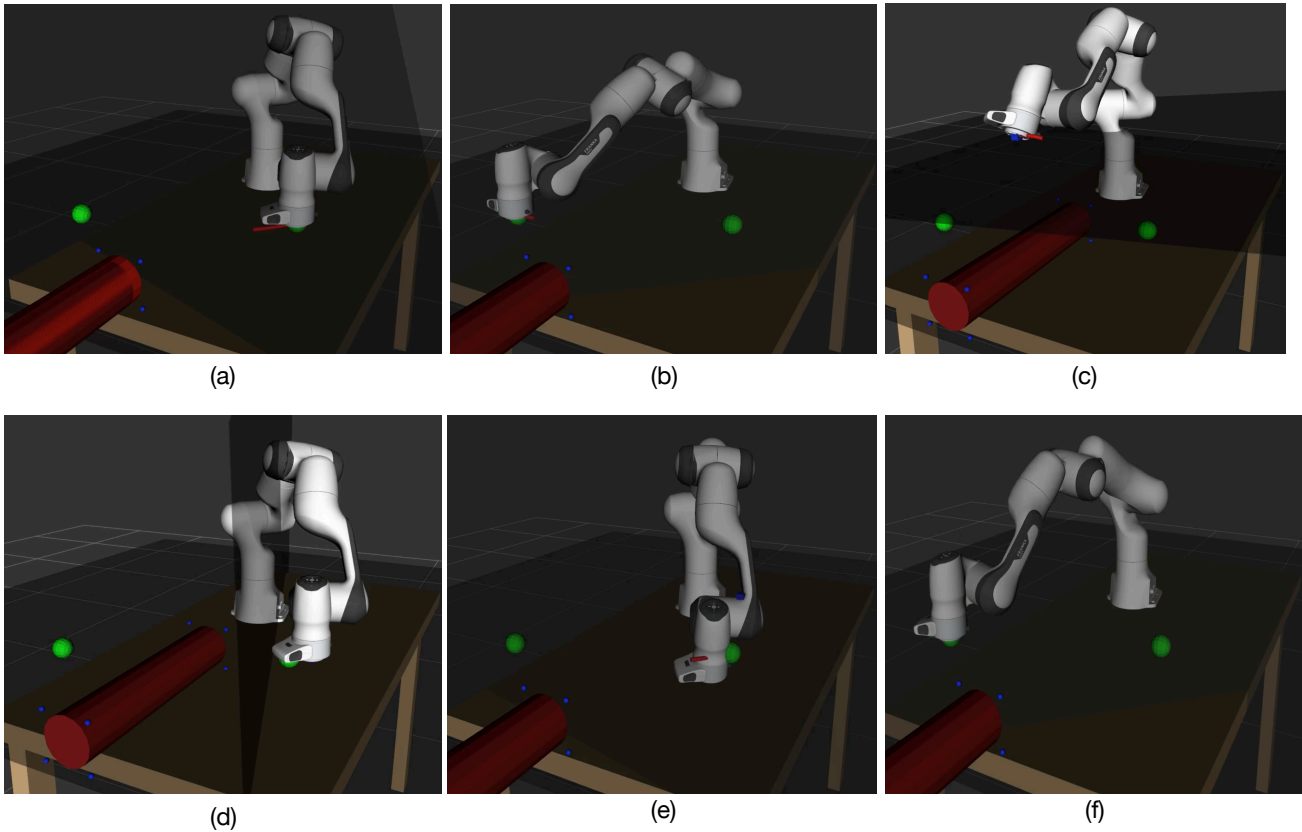


Fig. 5. (a)-(b) The robot alternates between the two desired positions at full speed when the human arm is not interfering. (c)-(d) When the human arm comes in the way, the robot automatically adapts its motion to keep executing its task, alternating between the two desired positions while maintaining the prescribed safety distance $d_{safe} = 20$ cm above the human arm. This introduces some vertical motion of the end-effector. (e)-(f) When the human arm moves away, the robot resumes its full speed.

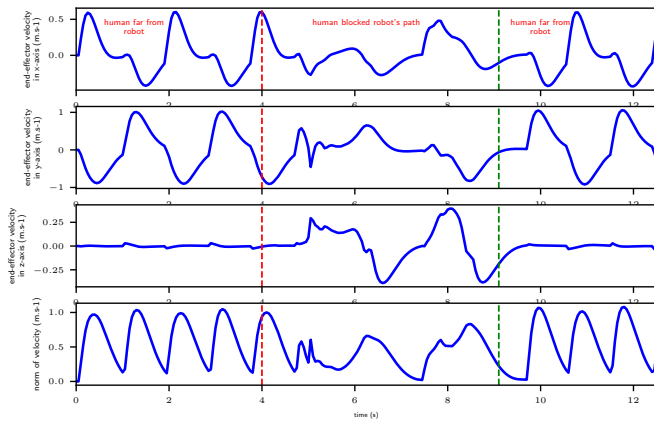


Fig. 6. End-effector’s cartesian velocity can be up to 1 m.s⁻¹ when the human arm is not interfering, as shown in the first 4 seconds. When the human arm comes in the way, the robot automatically adapts its motion. This introduces some vertical motion of the end-effector to maintain the prescribed safety distance and results in slower horizontal motion (seconds 4 to 9). When the human arm moves away, the robot resumes its full speed.

of a separating plane between them. Unlike other methods, we don’t assume that the collision can always be avoided, because that’s impossible to guarantee in a partially unknown dynamic environment such as a shared workspace. We guarantee instead that, if a collision can’t be avoided, the robot is at rest at the time of collision, so that it doesn’t inject its own kinetic energy in the collision. We propose a linear-quadratic formulation which provides a working solution just a few millimeters away from the optimal solution to the original nonlinear problem, and which can be obtained in a few milliseconds with an off-the-shelf QP solver.

Our collision avoidance and passive motion safety guarantees rely entirely on being provided an adequate perception and prediction of the human motion. If these are not accurate, there is a risk that a collision occurs before the robot manages to stop. In that case, we rely on the manipulator robot’s mechanical and control design to lower the risk of injury. Our next objective is to develop this perception and prediction of the human motion, based on data fusion from a collection of 3D cameras looking at the shared workspace.

REFERENCES

- [1] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, “The dlr lightweight robot: design and control concepts for robots in human environments,” *Industrial Robot: an international journal*, vol. 34, no. 5, pp. 376–385, 2007.
- [2] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.
- [3] B. Faverjon and P. Tournassoud, “A local based approach for path planning of manipulators with a high number of degrees of freedom,” 1987.
- [4] C.-S. Tsai, J.-S. Hu, and M. Tomizuka, “Ensuring safety in human-robot coexistence environment,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4191–4196, IEEE, 2014.
- [5] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

- [6] M. Kimmel and S. Hirche, “Invariance control for safe human–robot interaction in dynamic environments,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1327–1342, 2017.
- [7] S. Bouraine, T. Fraichard, and H. Salhi, “Provably safe navigation for mobile robots with limited field-of-views in dynamic environments,” *Autonomous Robots*, vol. 32, no. 3, pp. 267–283, 2012.
- [8] S. Al Homsy, A. Sherikov, D. Dimitrov, and P.-B. Wieber, “A hierarchical approach to minimum-time control of industrial robots,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2368–2374, IEEE, 2016.
- [9] N. Bohórquez, A. Sherikov, D. Dimitrov, and P.-B. Wieber, “Safe navigation strategies for a biped robot walking in a crowd,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 379–386, IEEE, 2016.
- [10] Franka Emika, “Panda,” tech. rep., Accessed.[Online]. Available: <https://www.franka.de>.
- [11] D. Kimm and D. V. Thiel, “Hand speed measurements in boxing,” *Procedia Engineering*, vol. 112, pp. 502–506, 2015.
- [12] T. Grujić and M. Bonković, “Measurement and analysis of human hand kinematics,” *World Academy of Science, Engineering and Technology International Journal of Biomedical and Biological Engineering*, vol. 9, no. 2, p. 97, 2015.
- [13] P. Svamy, M. Tesar, J. K. Behrens, and M. Hoffmann, “Safe physical HRI: Toward a unified treatment of speed and separation monitoring together with power and force limiting,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7580–7587, Nov 2019.
- [14] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, “A fast procedure for computing the distance between complex objects in three-dimensional space,” *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
- [15] F. Kanehiro, F. Lamiraux, O. Kanoun, E. Yoshida, and J.-P. Laumond, “A local collision avoidance method for non-strictly convex polyhedra,” *Proceedings of robotics: science and systems IV*, 2008.
- [16] S. Brossette and P.-B. Wieber, “Collision avoidance based on separating planes for feet trajectory generation,” in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 509–514, IEEE, 2017.
- [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] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [19] S. Toyer, A. Cherian, T. Han, and S. Gould, “Human pose forecasting via deep markov models,” in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2017.
- [20] P. Kratzer, M. Toussaint, and J. Mainprice, “Motion prediction with recurrent neural network dynamical models and trajectory optimization,” *arXiv preprint arXiv:1906.12279*, 2019.
- [21] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, “qpocps: A parametric active-set algorithm for quadratic programming,” *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.