Journal of Intelligent & Robotic Systems (2024) 110:107
https://doi.org/10.1007/s10846-024-02146-8

REGULAR PAPER

Experimental Assessment of a Vision-Based Obstacle Avoidance Strategy for Robot Manipulators: Off-line Trajectory Planning and On-line Motion Control

Cecilia Scoccia[1] · Barnaba Ubezio[2] · Giacomo Palmieri[1] · Michael Rathmair[3] · Michael Hofbaur[3]

Received: 15 May 2023 / Accepted: 10 July 2024 / Published online: 23 July 2024
© The Author(s) 2024

Abstract
Human-Robot Interaction is an increasingly important topic in both research and industry fields. Since human safety must be always guaranteed and accidental contact with the operator avoided, it is necessary to investigate real-time obstacle avoidance strategies. The transfer from simulation environments, where algorithms are tested, to the real world is challenging from different points of view, e.g., the continuous tracking of the obstacle and the configuration of different manipulators. In this paper, the authors describe the implementation of a collision avoidance strategy based on the potential field method for off-line trajectory planning and on-line motion control, paired with the Motion Capture system Optitrack PrimeX 22 for obstacle tracking. Several experiments show the performance of the proposed strategy in the case of a fixed and dynamic obstacle, disturbing the robot's trajectory from multiple directions. Two different avoidance modalities are adapted and tested for both standard and redundant robot manipulators. The results show the possibility of safely implementing the proposed avoidance strategy on real systems.

Keywords Human-robot collaboration · Obstacle avoidance · Visual tracking · Joint velocity control

1 Introduction

Industrial robot manipulators, widely used in many industrial sectors, have always operated in automated production in an isolated way, in order to perform repetitive or dangerous tasks. These robots work with predefined movements and are therefore unable to adapt their behavior independently to always ensure safety within the workstation, implying the introduction of restrictive safety criteria. Over the years, the approach has evolved from a partially shared workspace between humans and robot manipulators to a total collaboration, thanks to the use of sensors to detect human presence and other obstacles. Thus, Human-Robot Collaboration (HRC), i.e., close cooperation between the two parties to complete a common task, has been enabled.

The strength of the HRC lies in the possibility of delegating repetitive, heavy and risky tasks to collaborative robots (cobots), while the human carries out all the value-added operations that require intelligence and decision-making skills. The workspace, even in the absence of physical barriers, allows for more functional workstation designs, where the operator is constantly monitored. The robot needs to be constantly monitored as well, through the use of advanced sensors, in order to ensure a safe collaboration. Often, the area surrounding the robot cell is divided into several zones, and the environment is inspected by a vision system. The manipulator performs the tasks at a previously set velocity and, when the human enters the zones where a collision might occur,

✉ Cecilia Scoccia
c.scoccia@staff.univpm.it

Barnaba Ubezio
Barnaba.Ubezio@aau.at

Giacomo Palmieri
g.palmieri@staff.univpm.it

Michael Rathmair
Michael.Rathmair@joanneum.at

Michael Hofbaur
Michael.Hofbaur@joanneum.at

[1] Department of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, Brecce Bianche, 12, Ancona 60131, Italy

[2] Institute of Smart System Technologies, University of Klagenfurt, Klagenfurt 9020, Austria

[3] JOANNEUM RESEARCH, Robotics, Klagenfurt 9020, Austria

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

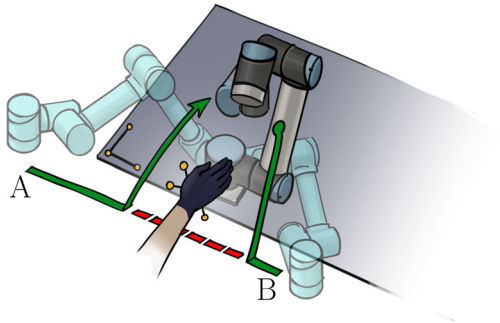

Fig. 1 Collision avoidance scenarios during a linear trajectory where the robot moves from point A to point B and is disturbed by a human operator's hand. Avoidance of an obstacle entering within the end-effector trajectory

the robot slows down or immediately stops before or in case of contact. The safety standards and specifications EN ISO 10218-2 [1] and ISO/TS 15066 [2] describe such modality as Speed and Separation Monitoring (SSM), where the minimum distance allowed for full-speed operation of the robot is calculated based on the tracking measurements of the operator entering the area. For productivity reasons, it is generally accepted to slow down the robot, so that high-force impacts are prevented, instead of directly stopping the machine. Ultimately, avoiding any contact/collision by re-planning the robot's trajectory in real-time, as shown in Figs. 1, would be preferable.

1.1 Challenges in Obstacle Avoidance

The implementation of the latter task becomes more challenging when a possible intrusion within the area of the robot's motion comes from different directions and not just within the end-effector trajectory, as represented in Fig. 2. Furthermore, obstacles may be fixed or moving. Traditional obstacle avoidance assumes static obstacles, but robots would also need to anticipate the intentions and movements of dynamic elements, such as moving objects and humans in the vicinity of the manipulator. When human presence is

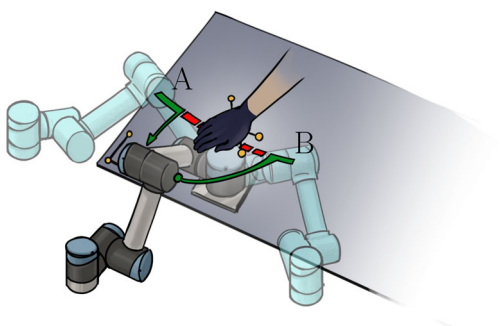

Fig. 2 Avoidance of an obstacle entering the robot's elbow trajectory

expected, making safety a key issue, the obstacle avoidance strategies should ensure that the robot makes safe decisions, even if it means taking a longer path or avoiding risky maneuvers.

The use of predefined maps to plan a path for a robot, which is one of the classical adopted techniques, may pose limitations. First of all, predefined maps may not account for dynamically changing environments so the robot might not be able to respond to unexpected obstacles that appear after the map was created. In addition, the maps cover only certain area and some environments could be too complex for adequate mapping, making it difficult for the robot to navigate effectively.

From the sensing point of view, effective obstacle avoidance requires the use of sensors for obstacle detection and tracking, ideally dealing with environmental conditions such as low lighting or occlusion. Properly selecting such sensors is crucial for the practical implementation of the control algorithms.

Finally, optimal path planning and motion control are necessary. Indeed, an optimal trajectory generation algorithm must ensure smoothness in both position and velocity to be effectively utilized in a real system's controller.

1.2 Contribution

The proposed work describes the implementation of an obstacle avoidance strategy, addressing the aforementioned challenges, with several experimental tests. Starting from previously obtained simulation results [3, 4], the authors implement a combination of off-line path planning, on-line motion control and real-time tracking with a vision-based Motion Capture (MoCaP) system, to perform static and dynamic obstacle avoidance with standard and redundant manipulators. Although the algorithms have been designed to be independent on the particular obstacle detection sensor, the use of such system provides fast, redundant and highly precise tracking data.

The content of this paper provides an extensive and accurate review of the literature, the mathematical description of the developed strategy based on the Artificial Potential Field (APF) technique [5] and the results of several experiments. With respect to currently existing solutions, this work deals with both static and dynamic obstalces. Furthermore, standard algorithms are improved with the use of Bézier curves to avoid abrupt accelerations and vibrations of the robot joints in the event of sudden trajectory changes. Additionally, a technique based on the least-square damped method for inverting Jacobian matrices is introduced to prevent the robot from passing through points that are too close to singular configurations.

Finally, two different operation modes are described and validated. The first mode exploits the full Degrees

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Of Freedom (DOF) of the robots, while the second mode maintains the robot's end-effector orientation fixed. The research shows the possibility of changing the robot's operating mode independently of the real-time recognition of the obstacle motion. As a result, the system proves to be dynamic, efficient, and adaptive to any sensor that provides obstacle-tracking measurements. The remainder of the paper is organized as follows: Section 2 provides an accurate analysis of related works, both for obstacle avoidance strategies and for vision and MoCaP systems; Section 3 discusses the obstacle avoidance strategy implemented, in particular the trajectory path planning algorithm based on the potential field, the on-line motion control and the explanation of the two possible operating modalities implemented for the UR5; Section 4 describes the system architecture for the acquisition of the obstacle data and reports the performance of the strategy applied on real systems, both with a 6-Degrees Of Freedom (DOF) Universal Robots UR5 and a redundant 7-DOF KUKA LBR iiwa; Section 6 reports concluding remarks and future works.

2 Related Work

2.1 Obstacle Avoidance

Obstacle avoidance and obstacle detection have been widely researched topics for decades in the robotics community. Classical methods involving the optimization of pre-defined goal functions [5, 6] have been followed over the years by alternative strategies. In [7, 8], for example, the authors perform online scaling of dynamic safety zones to avoid potential collision between a human operator and the robot in motion. Authors in [9] propose a cell decomposition method involving probabilistic sampling for path planning. Cell decomposition requires dividing the environment into free and occupied sections so that an avoidance trajectory within the free regions is computed [10]. Other probabilistic solutions have also been proposed: in [11], a collision-free path is generated using an RRT (Rapid-exploring Random Trees) algorithm, whereas a predictive model based on a BIT method (Batch Informed Trees) is presented in [12], which improves the planned path's points connection.

Strategies based on machine learning and neural networks have also been extensively investigated. In [13], the authors propose an improvement of the Mamdani fuzzy controller [14], by increasing the number of sensors to detect the environment and reducing the fuzzy rules. In [15] and [16], Deep Reinforcement Learning (DRL) approaches for vision-based obstacle avoidance are presented, processing data obtained from the raw sensor data of the obstacle and training specific neural networks to plan optimal trajectories. However, the smoothness within the avoiding motion and return to the original path is not always guaranteed.

Another interesting technique in [17] combines DRL with dynamic reward to improve the efficiency of route planning and obstacle avoidance. This makes it possible to arrive at an optimal solution faster and have a good strategy. However, the effectiveness of DRL may not be valid in the case of complex scenarios and also, the validation of the algorithm was done only in simulation and may not represent the real environment.

The use of machine learning techniques or neural networks could lead to limitations, including the requirement for a substantial amount of training data that may not be readily obtainable. Processing all this data could demand significant computational time, making the operation complex. Additionally, neural network-based systems may struggle to adapt if they were trained on static datasets. More recent and promising work [18] relies on a search algorithm based on a relative entropy policy to avoid obstacles. With respect to the work proposed in this paper, there are still some limitations and challenges. First, the algorithm was designed under the assumption that skills are learned in static environments. This may cause harm if it is applied to dynamic environments peculiar to real scenarios. Furthermore, it involves the solution of an optimization problem that minimizes the difference between the optimal trajectory distribution and the experimental distribution, while avoiding obstacles. The complexity of this optimization process, performed offline without real-time feedback from the environment, may lead to sub-optimal solutions and even require significant computational resources, making it less suitable for real HRC applications.

2.1.1 Obstacle Avoidance in Mobile Robotics

Mobile robot navigation also benefits from obstacle avoidance algorithms, both with neural networks [19] and with model-based approaches. An overview of the control strategies and algorithms for obstacle avoidance in mobile robots can be found in [20]. In the following, some notable solutions are reported and discussed, as the limitations of current algorithms are extremely similar in the case of robot manipulator arms, subject of this paper.

In [21], a Nonlinear Model Predictive Control (NMPC) is presented in order to solve point-stabilization problems with static and dynamic obstacles. It is based on the minimization of an error cost function that incorporates the obstacle as a time-varying constraint. The NMPC involves solving online optimization problems at each time step, but the computation time could be high if the system is complex. In addition, a prediction-based technique considers only short time horizons since, in the case of long-term predictions,

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

the computational demand is high, which would reduce the ability of the control to respond quickly.

Authors in [22] introduce a motion planning algorithm designed for mobile robots navigating in known environments with static obstacles. Leveraging Particle Swarm Optimization (PSO), the algorithm converges to a global minimum, utilizing a customized approach to generate search space coordinates. These coordinates are then employed by the PSO algorithm to establish the most efficient path between two specified end positions. In [23], the DAYKUN-BIP method is introduced as an innovative navigation strategy to ensure seamless and obstacle-free movement for mobile robots. The DAYKUN-BIP virtual target displacement (DVTD) method comes into play when a robot encounters difficulties in finding a suitable path. Upon sensor detection of potential barriers, multiple virtual goals are generated around the actual goal. The safest route is determined by assigning appropriate weights to each goal line based on selected factors, thereby visualizing the most optimal path.

Coverage Path Planning (CPP) techniques, typically used for service mobile robots, are able to generate smooth paths and avoid sharp turns, thus reducing accelerations and decelerations [24, 25]. The complexity and dynamism of the environment may not be handled optimally so real-time trajectory adaptation may not be respected. In addition, the effectiveness of CPP is often dependent on the accuracy and range of sensors used by the robotic system. The CPP technique may not handle all types of mobile robots also considering that, since these robots have limited energy resources, optimizing the path may require excessive energy consumption.

2.1.2 Obstacle Avoidance with Artificial Potential Field

Generating smooth trajectories is a requirement that translates in robot manipulators as well, in terms of position and velocity of the robot's joints. Motion smoothness is crucial for the implementation in a real HRC system.

The main challenges that researchers encounter are related to proper sensing of the environment, efficiency of the path planning algorithms and safety for the human operators. The most used techniques for dealing with these issues are based on the APF [5], which drives the robot toward the target position inside the workspace. Its implementation consists in building a set of forces able to drive the manipulator's end-effector to the goal. The potential field is composed of an attractive force (towards the goal) and a repulsive one (from the region of influence of the obstacle). These forces are associated with the velocities of the manipulator computed at the end-effector; the optimal trajectory can then be obtained by numerical integration. Strengths of this technique are its simplicity and intuitiveness, which make APF-based obstacle avoidance tasks easy to implement. In addition, this technique is computationally more efficient than other mentioned solutions in the literature, allowing for real-time responses. Such efficiency makes it extremely suitable for applications where the environment is dynamic and decisions must be made quickly. Finally, this technique allows the robot to navigate within the working environment without knowing it entirely. The APF suffers from local minimum problems, which are addressed in several works [26–28]. On the one hand the repulsive forces generated by obstacles help the robot navigate around them and escape from trapped situations; on the other, the robot may get stuck in local minima, especially for complex paths, and therefore not reach the highest possible accuracy. In addition, the overall performance could heavily depend on parameter tuning, requiring careful calibration.

The method proposed in [26] deals with the velocity control of an Unmanned Aerial Vehicle (UAV) to avoid the local minimum trap in a dynamic environment. It assumes knowledge of the relative motion states of surrounding obstacles, but the robustness of the algorithm is based only on simulations, not considering the variability of scenarios and not extending the demonstration to a real case. Puriyanto et al. [27] presents an Improved APF (IAPF) algorithm to deal with the local minimum issue, relying on selected tolerance values and pre-defined thresholds. The complexity of the environment is not taken into account, since it is assumed to be known and static, therefore the scalability of the IAPF algorithm to handle larger environments or scenarios with multiple robots is not addressed.

In several cases, the presence of noise and uncertainties can impact the reliability of path planning algorithms. Another example of path planning is proposed in [29], where the authors use a saturation function for the attractive velocity in order to avoid oscillations around the goal, while for the repulsive velocity a spring-damper system is used to eliminate noise in the proximity of the obstacles.

The APF technique is the basis for the presented work and is considered as the origin of attractive and repulsive velocities, used to compute the end-effector trajectory, as will be explained in Section 3. The method outlined in the paper offers a solution to overcome certain challenges encountered by the previously described approaches. To begin with, this strategy was initially validated through simulation and subsequently tested in the laboratory, thereby confirming its efficacy. The workspace does not require any cell decomposition for trajectory definition or predictive methods for obstacle-free trajectory generation. The proposed method considers both static obstacles already present in the workspace and real-time updates of the position of moving obstacles as input, generating, a smooth end-effector trajectory without accelerations or vibrations, thanks to the introduction of Bezier curves. However, the main disadvantage is the necessity of using multiple cameras

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

to cover the workspace adequately and prevent occlusion issues. In fact, at least two cameras must monitor the object to ensure accurate obstacle detection. Moreover, while using many control parameters in the obstacle avoidance algorithm can lead to an effective strategy, these parameters also necessitate a well-established baseline setting that could vary from one application to another.

2.2 Vision Sensors and Motion Tracking

HRC applications require a safety system in order to avoid harm to the operators. In this context, obstacle avoidance becomes human avoidance, as the robot must continue its motion while avoiding intrusion from parts of the operator's body, i.e., hands, arms, head etc. It is therefore crucial to be aware of surroundings, knowing the presence and location of people, objects and robots.

Vision and MoCaP systems have gained popularity as human tracking devices, thanks in particular to their easy installation and usability. The main vision systems used in robotics are monocular and stereo cameras, RGBD cameras and Time-of-Flight (ToF). The reason for this popularity is based on multiple aspects. First, vision systems provide extensive information on the work environment and its features, and can be used in almost any task, from object recognition to localization and navigation. On the other hand, these systems suffer from occlusion and light conditions issues, and require high computational effort in data processing in the case of real-time applications. Motion capture systems, based on camera sensors, are extremely accurate in tracking objects or human body movements. Furthermore, by working with very high frame rates (up to 360 FPS) they offer the possibility of an immediate feedback. However, these systems are extremely expensive and require an environment free of other reflective objects that could cause noise. Both of these systems have strengths and weaknesses and the choice depends on the type of application to be performed.

Halme et al. [30] provide a general review of vision-based safety technologies and methods to detect humans and objects, illustrating the standards and the guidelines for a safe HRC in practical applications. They propose five categorizations for vision-based monitoring and safety systems: distance between points and obstacles, collision avoidance, human interaction recognition, multi-sensor data fusion and visualization/monitoring of safety zones. The main objective of [30] is to provide a technology analysis of vision-based security systems for industrial environments. The work suggests that there may be gaps in the current state of these systems, in terms of robustness, reliability, or other critical factors for industrial applications. For this reason, vision systems are not yet certified, leading to limitations in their use.

Schmidt et al. [31] propose an adaptive control for obstacle avoidance based on a depth camera. The 3D point cloud obtained is used for the detection of potential obstacles within the monitored area. The authors emphasize the potential of depth images in ensuring effectiveness for safety monitoring in HRC tasks, and the low cost of the sensors. The proposed strategy relies on the removal of stationary and known objects making it less suitable in terms of adaptability for complex scenarios. There is also no information regarding the robustness of the system or the possibility of scalability. Other works also exploit depth cameras for the implementation of distance control algorithms, in order to generate repulsive vectors that push away the robot from the obstacle area [32, 33].

Another system that is widely used is the Kinect. Among its advantages is its low cost, which makes it more attractive than other more expensive industrial systems. In addition to RGB data, it can also provide distance estimates allowing to recognize objects for obstacle avoidance. The ease of use and the possibility to rely on available pre-built libraries and tools also makes it a valuable sensor. However, Kinect devices have a limited range of action and field of vision making them unsuitable for very large environments. The accuracy and precision of the depth camera may not be sufficient for applications requiring extreme precision. Their use is exclusively indoor. In [34], data from multiple Kinect images are used to generate dynamic safety volumes around a robot and humans. If safety volumes collide, the robot decreases its velocity. Other works include the Kinect sensor to implement similar frameworks of human detection and motion tracking within the robot area [35, 36].

Many authors employ vision systems and machine learning, particularly deep learning, to recognize objects [37]. In [38], the authors propose an approach for video object detection and tracking. The methodology is structured into three distinct phases: detection, tracking, and evaluation. The detection phase encompasses foreground segmentation and noise reduction, employing a proposed Mixture of Adaptive Gaussian models to attain efficient foreground segmentation. Additionally, a fuzzy morphological filter is implemented to eliminate noise from the segmented frames. A technique for robot vision localization is suggested in [39], utilizing an iterative Kalman particle filter, enabling the global positioning of the robot. An interesting method that could be extended in HRC applications is proposed by the authors in [40], in which gesture patterns based on body-proportion characteristics around the shoulders are extracted. The authors introduce a 3D human-gesture interface for fighting games using a motion recognition sensor, which effectively models and analyzes motion through mathematical representation and principal component analysis. A novel pattern matching algorithm is proposed to minimize motion constraints in the recognition system. The results showcase a high-quality 3D motion interface for realistic gaming experiences, demonstrating the potential of real-time processing technology.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Occlusion plays a crucial role in vision sensors and is one of the open issues for such systems. Cameras are often affected by light conditions, dust, smoke and other environmental factors which limit the correct identification of objects. To obtain accurate measurements, depth cameras often require precise calibration, which can be a complex and time-consuming process. Furthermore, people and robots may occlude each other for an unknown amount of time during image acquisition. For this reason, multiple sensors are used in fusion frameworks. Redundant sensors are also fundamental in terms of safety in HRC. An example of sensor fusion in the presence of occlusion is with the use of Inertial Measurement Unit (IMU) sensors as wearable devices, as presented in [41], where the fusion of Optitrack cameras and an IMU ensures robust human tracking and localization. MoCaP systems like the Optitrack also provide redundant information, as many cameras are mounted in the scene to track special infrared reflective markers from multiple directions, thus improving the coverage and reducing occlusion possibilities. The MoCaPsystems are extremely accurate and precise in tracking objects and their movement in real-time. They are very suitable for HRC applications, ensuring both human and robot tracking. These systems are also extremely used for biomechanical motion analysis, animation and gesture recognition. Some systems, such as the OptiTrack PrimeX, allow multiple objects to be tracked simultaneously and offer the possibility of fusing information from other sensors. Other works make use of such systems: in [42], the authors provide an APF collision avoidance method for a Universal Robot UR10 manipulator, tracking the human arm with two wearable MoCaP systems, the HTC VIVE and the wearable AntiLatency. It requires two base stations as a reference, 18 infrared receiver sensors and a 6 DOF wearable optical camera attached to the tracked object.

In summary, the combination of the close distance measurements from the vision or the MoCaP sensors allows the robot to perform its task while keeping safe and collision-free motions and sharing the workspace with operators. Vision systems, such as Intel RealSense or Kinect, are cheaper and require no markers. These systems are highly dependent on lighting conditions and suffer from measurement noise . Some surfaces can reflect and create challenges for obstacle detection or are subject to occlusions, making it difficult to detect objects covered by other objects. Their calibration is time-consuming and they have a limited field of view. For this reason, they can be inaccurate or even fail to acquire particular movements. MoCaP systems, on the other hand, offer a high accuracy and inherently increased coverage providing exceptional precision in position data, rapid detection and high sampling frequency.

3 Materials and Methods

3.1 Obstacle Avoidance Strategy

This study describes the experimental implementation of a control strategy for obstacle avoidance, which uses a combination of off-line path planning and on-line motion control algorithms. The algorithms, introduced in [3, 4, 43], are here applied to both a standard and a redundant manipulator operating in a three-dimensional workspace. The strategy is validated with a Universal Robot UR5 and a 7-DOF KUKA LBR iiwa 14, and with the use of the Optitrack PrimeX 22 MoCaP system for obstacle tracking. The overall proposed control strategy for a manipulator operating in a dynamic environment can be formulated by combining:

1. An *Off-line path planning* able to plan the trajectory of the robot's end-effector considering the possible presence of obstacles along the path and consequently adjusting the trajectory based on the position of the obstacle before the motion starts.
2. An *on-line motion planning* that controls the robot with joint velocity commands to enable the robot to avoid moving obstacles.
3. A *redundancy control strategy* to avoid collisions between the elements of the kinematic chain of the manipulator with obstacles, and to try to keep the trajectory of the end-effector unchanged.

Two different operation modes for collision avoidance are introduced. The first one, referred to as "Full Mode" allows for 6-DOF avoidance motions of the end-effector. The second mode, referred to as "Reduced Mode", leaves the orientation of the end-effector unchanged during the perturbation from the obstacle.

Note that the overall control algorithm assumes fully functional materials involved, and does not generally consider uncertain systems and fault-tolerance. Recent work [44] considers a fault-tolerant nonlinear observer-based control framework to cope with uncertainties in the motors of a UAV. However, it is assumed that in the proposed experiments the electric motors of the robot manipulator work normally and that the commands sent via TCP/IP do not present communication errors.

3.2 Off-line Path Planning

The path planning algorithm utilized in order to define the cartesian motion $\mathbf{x}_e(t)$ of the end-effector, considering the initial presence of static obstacles, is applied as follows.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

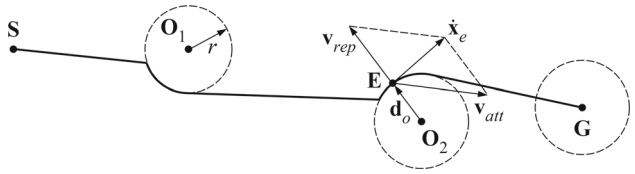

Fig. 3 Potential Fields for trajectory planning

A repulsive and an attractive velocity, respectively $\mathbf{v}_{rep}$ and $\mathbf{v}_{att}$ components originated by an APF, are used to drive the end-effector **E** following the minimum potential path towards the goal. As shown in Fig. 3, **S** is the initial position of the end-effector, **G** is the goal and $\mathbf{O}_i$ are the obstacles, with their region of influence outlined by their radius r. Considering the distances $\mathbf{d}_O = \mathbf{E} - \mathbf{O}_i$ and $\mathbf{d}_G = \mathbf{G} - \mathbf{E}$, the equations that define attractive and repulsive velocities can be formulated as in [3]:

$$\mathbf{v}_{att} = \begin{cases} v_{att}\dfrac{\mathbf{d}_G}{r} & d_G < r \\ v_{att}\mathbf{d}_G & d_G \geq r \end{cases} \tag{1}$$

$$\mathbf{v}_{rep} = \begin{cases} \dfrac{v_{rep}}{d_O}\left(\dfrac{1}{d_O} - \dfrac{1}{r}\right)\nabla d_O & d_O < r \\ \mathbf{0} & d_O \geq r \end{cases} \tag{2}$$

where the symbol ∇ indicates the gradient operator. Consequently, the goal is to guide the robot to move in a way that minimizes a potential function. The potential function, or APF, is constructed from the balance of attractive and repulsive velocities, so that the robot aims to navigate through the environment by following the path of least resistance and least cost according to the APF:

$$\dot{\mathbf{x}}_e = \mathbf{v}_{rep} + \mathbf{v}_{att} \tag{3}$$

This strategy implicitly imposes a speed constraint, with higher velocities when the end-effector is further from the goal, gradually decreasing as it approaches. Naturally, the greater the distance from the desired point, the stronger the attractive field it experiences, resulting in a higher traveling speed.

The reference end-effector trajectory at each time step can be found by integrating the resulting velocity $\dot{\mathbf{x}}_e$:

$$\mathbf{x}_e(t + \mathrm{d}t) = \mathbf{x}_e(t) + \dot{\mathbf{x}}_e(t)\mathrm{d}t \tag{4}$$

which is updated in accordance with the velocity imposed at each step $\mathrm{d}t$.

Ultimately, the algorithm will allow the robot to follow this given reference trajectory while avoiding collisions with known obstacles or other agents that might appear along the path. The process concludes when the distance between the end-effector and the target falls below a predetermined threshold.

Unfortunately, the trajectory resulting from Eqs. (1)-(4) is characterized by a short radius and sharp corners resulting in high accelerations and vibration problems. An example is illustrated in Fig. 4, where two obstacles situated between the starting and goal points are disturbing the motion along the ideal linear trajectory. The planning algorithm generates the black curve that consistently stays outside the influence region of the obstacles at all points. Nevertheless, this curve undergoes rapid directional changes at two specific points, namely where the trajectory intersects with the influence spheres of the obstacles. In order to avoid this effect, an interpolation procedure has been applied to generate a smoother trajectory based on a third-order Bezier curve.

Having $n + 1$ 3D points $\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_n$, where n is the order of the Bezier curve, the latter is defined as:

$$\mathbf{B}(s) = \sum_{i=0}^{n} \binom{n}{i} \mathbf{C}_i (1-s)^{n-i} s^i, \quad s \in [0, 1] \tag{5}$$

where s is the curvilinear abscissa of the curve. Considering a third order Bezier curve, Eq. (5) becomes:

$$\mathbf{B} = \mathbf{C}_0(1-s)^3 + 3\mathbf{C}_1 s(1-s)^2 + 3\mathbf{C}_2 s^2(1-s) + \mathbf{C}_3 s^3 \tag{6}$$

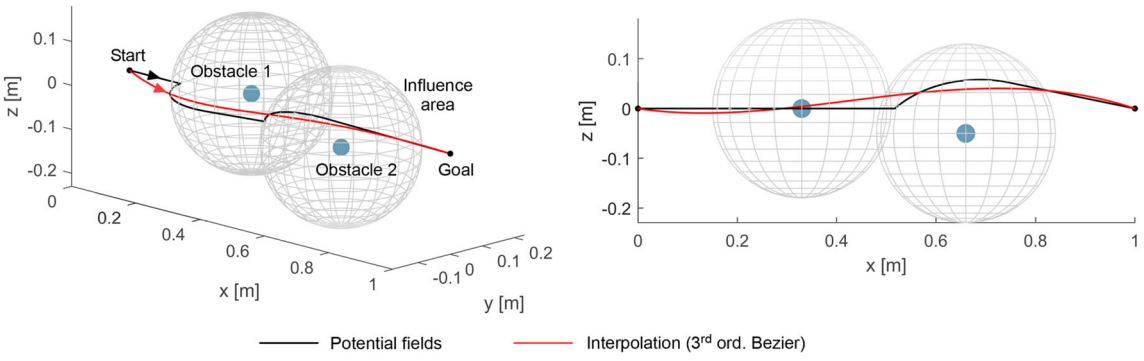

Fig. 4 Example of path planning in the case of potential fields (black trajectory) and with third-order Bezier curves (red trajectory)

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Here, $\mathbf{C}_0$ and $\mathbf{C}_3$ are respectively the known initial and the final point of the overall trajectory $\mathbf{x}_e(t)$, while $\mathbf{C}_1$ and $\mathbf{C}_2$ are the intermediate points that allow to generate the curve $\mathbf{B}$ as the best approximation of the original one. To determine the six variables that define points $\mathbf{C}_1$ and $\mathbf{C}_2$ and minimize the quadratic error between the original path and the Bézier curve, an optimization problem needs to be addressed. Equation 6 can be re-written in matrix form as:

$$\begin{aligned}\mathbf{B}^T = &\left[(1-s)^3,\ s^3\right]\left[\mathbf{C}_0,\ \mathbf{C}_3\right]^T \\ &+\left[3s(1-s^2),\ 3s^2(1-s)\right]\left[\mathbf{C}_1,\ \mathbf{C}_2\right]^T\end{aligned} \tag{7}$$

When the curvilinear abscissa s is discretized into m samples, the trajectory $\mathbf{x}_e$ transforms into a collection of m points. Consequently, Eq. (7) can be expressed m times for the points $\mathbf{B}_j$, where j takes values from 1 to m, assuming the following form:

$$\mathbf{N} = \mathbf{S}_1\mathbf{D}_1 + \mathbf{S}_2\mathbf{D}_2 \tag{8}$$

where:

$$\begin{aligned}&\mathbf{N} = \left[\mathbf{B}_1\ \ldots\ \mathbf{B}_m\right]^T \quad \mathbf{S}_1 = \begin{bmatrix} 1-s_1^3 & s_1^3 \\ \vdots & \vdots \\ 1-s_m^3 & s_m^3 \end{bmatrix} \\ &\mathbf{S}_2 = \begin{bmatrix} 3s_1(1-s_1^2) & 3s_1^2(1-s_1) \\ \vdots & \vdots \\ 3s_m(1-s_m^2) & 3s_m^2(1-s_m) \end{bmatrix} \\ &\mathbf{D}_1 = \left[\mathbf{C}_0\ \mathbf{C}_3\right]^T \quad \mathbf{D}_2 = \left[\mathbf{C}_1\ \mathbf{C}_2\right]^T\end{aligned} \tag{9, 10}$$

By manipulating Eq. (8) and substituting the matrix $\mathbf{N}$ with the corresponding matrix $\mathbf{X}$, which is constructed using the points from the trajectory $\mathbf{x}_e$ obtained through the APF algorithm, a straightforward closed-form solution for the optimal set of coefficients $\mathbf{D}_2$ can be readily determined:

$$\mathbf{D}_2 = \mathbf{S}_2^{\dagger}(\mathbf{X} - \mathbf{S}_1\mathbf{D}_1) \qquad \mathbf{X} = \left[\mathbf{x}_{e,1}\ \ldots\ \mathbf{x}_{e,m}\right]^T \tag{11}$$

where † denotes the pseudo-inverse operator based on the Moore-Penrose definition. This operator inherently yields the coefficients of the curve that optimally matches the original trajectory by minimizing the least squares error. These coefficients can be input to Eq. (6) in order to generate the smoothed reference trajectory.

3.3 On-line Motion Control

The goal of the on-line motion control is to compute the joint velocity commands that are being continuously sent to the robot in order to avoid moving obstacles. For this reason, several control points are defined on the manipulators, and an inverse kinematics problem is performed. The control point closest to the obstacle at each step is then considered for the avoidance.

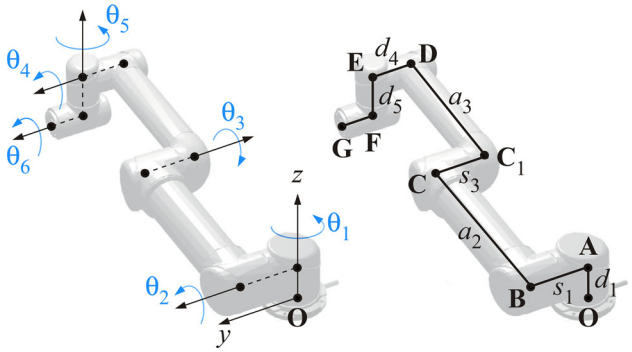

Fig. 5 Kinematic chain and control points for the Universal Robot UR5

The kinematic schemes of the Universal Robot UR5 and the KUKA LBR iiwa 14, are shown in Figs. 5 and 6, respectively.

For the UR5, the kinematic chain is characterized by eight control points (**A**, **B**, **C**, $\mathbf{C_1}$, **D**, **E**, **F**, **G**). The components of the joint space position vector are: $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$, respectively the base, the shoulder, the elbow and the three wrist joints. In the case of the KUKA LBR, instead, seven joint positions $\mathbf{q} = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6\ \theta_7]^T$ and a total of 14 control points ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{A_3}$, $\mathbf{A_4}$, $\mathbf{B_1}$, $\mathbf{B_2}$, $\mathbf{B_3}$, $\mathbf{B_4}$ $\mathbf{C_1}$, $\mathbf{C_2}$, $\mathbf{E}$) are defined, considering intermediate distances between the various segments, as shown in Fig. 6. A velocity is assigned to the control point of the robot closest to the obstacles present in the workspace; thereby, the control point is pushed away from the obstacle region, while the original goal trajectory for the end-effector is maintained.

A state vector $\mathbf{x} = [x\ y\ z\ \alpha\ \beta\ \gamma]^T$ is defined, to indicate the estimated pose of the end-effector in the Cartesian

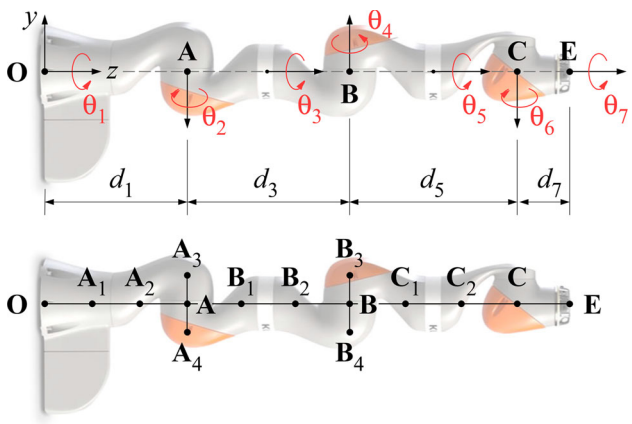

Fig. 6 Kinematic chain (top) and control points (bottom) for the KUKA LBR iiwa robot

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

space. The last three components represent the Euler angles according to the ZYZ convention. The forward kinematics of the manipulators is then described by:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \tag{12}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{x}}_p \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_o \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{13}$$

where $\mathbf{f}$ is the position forward kinematics, $\dot{\mathbf{x}}$ is the velocity vector composed by the linear velocity $\dot{\mathbf{x}}_p$ and the angular velocity $\boldsymbol{\omega}$, $\dot{\mathbf{q}}$ is the joint velocities vector and $\mathbf{J}$ is the analytic Jacobian, composed by $\mathbf{J}_p$ and $\mathbf{J}_o$, which are the position and orientation Jacobian matrices:

1. for the UR5: $\mathbf{J}$ is the (6×6) Jacobian matrix, $\mathbf{J}_p$ and $\mathbf{J}_o$ have a dimension of (3×6).
2. for the KUKA: $\mathbf{J}$ is the (6×7) Jacobian matrix, $\mathbf{J}_p$ and $\mathbf{J}_o$, respectively the position and the orientation Jacobian matrices, have a dimensions of (3×7).

As mentioned before, a velocity vector is assigned to the control point of the robot which is closest to one of the obstacles in the workspace. The aim is to leave the motion of the end-effector unchanged, while the control point is pushed away from the obstacle. Referring to Figs. 7 and 8, a pair of points $\mathbf{P}_r$ and $\mathbf{P}_o$ (respectively belonging to the robot and the obstacle) at the minimum distance d_O is identified at each time step. The region of influence of each control point is delimited by the radius r. If the conditions $d_O < r$ is verified, a repulsive velocity $\dot{\mathbf{x}}_0 = a_v v_{rep} \mathbf{d}_O$ along the direction of $\mathbf{d}_O$ is assigned to the relative control point, where a_v is an activation factor, function of d_O, r and other parameters that are used to define a critical minimum safety distance. The control point changes according to the criterion of minimum distance from the obstacle, and could also be the end-effector $\mathbf{E}$.

The safety regions that enclose the kinematic structure of the KUKA LBR have been defined as control spheres that have equidistant centers, as shown in Fig. 7. For the UR5 a different solution has been adopted. Only one safety region

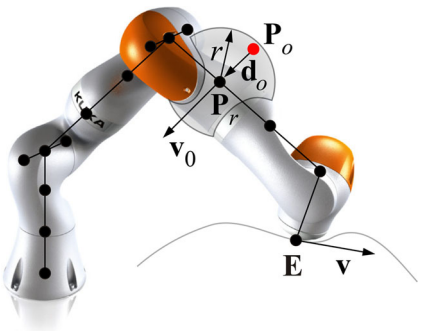

Fig. 7 Linear velocity of the end-effector $\mathbf{E}$ and of the control point $\mathbf{P}_r$ closest to the obstacle $\mathbf{P}_o$

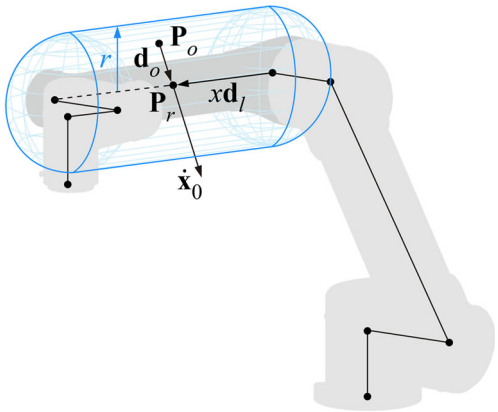

Fig. 8 Region of influence of a link and repulsive velocity

per each link has been defined as in Fig. 8: the cylindrical region has a height equal to the length of the link and two hemispheres at the extremities of the link. As for the KUKA, the radius of the cylindrical/hemispherical safety volume is variable as a function of distance and velocity of the obstacles. The reader can refer to [3, 4, 43] for more information regarding the construction of these regions.

The following expressions must be imposed in order to assign the two velocity tasks in the joint space, i.e., following the reference end-effector trajectory and avoiding the obstacle:

$$\mathbf{J}\dot{\mathbf{q}} = \dot{\mathbf{x}}_e \tag{14}$$

$$\mathbf{J}_{0p}\dot{\mathbf{q}} = \dot{\mathbf{x}}_0 \tag{15}$$

where $\mathbf{J}_{0p}$ represents the (3×7) upper part of the Jacobian matrix $\mathbf{J}_0$ associated to the velocity of the point $\mathbf{P}_r$ and $\dot{\mathbf{x}}_0$ is the imposed repulsive velocity.

Additionally, in order to prevent singularity for both the robots considered, a damped least-square strategy is used [45, 46]. The damped inverse matrix can be obtained as:

$$\mathbf{J}^* = \mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{J}^{\mathbf{T}} + \lambda^2 \mathbf{I})^{-1} \tag{16}$$

where λ is a damping factor, which is a function of the smallest singular value of the Jacobian matrix. However, this approximation could introduce a position error that must be recovered by a proportional term in the control law, as typically used in Closed-Loop Inverse Kinematic (CLIK) control laws [47, 48], that will be addressed in the following.

3.3.1 Redundancy Control and Operation Modes

Since the KUKA LBR is a redundant robot, the inverse position kinematics problem in not uniquely defined. Therefore, the obstacle avoidance strategy is based on the null space control for redundant manipulators, a technique used to exploit

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

the redundancy of the robot for the avoidance motion while simultaneously accomplishing a primary task in the task space. A numerical solution based on the inverse kinematics velocity problem is required in order to write the control law [5, 49, 50]:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \mathbf{N}\dot{\mathbf{q}}_0 \tag{17}$$

$$\mathbf{q}(t + \mathrm{d}t) = \mathbf{q}(t) + \dot{\mathbf{q}}(t)\mathrm{d}t \tag{18}$$

The terms in Eq. (17) are defined as follows:

1. $\dot{\mathbf{q}}_0$ is the joint null-space velocity;
2. $\mathbf{J}^{\dagger} = \mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{J}^{\mathbf{T}})^{-1}$ is the pseudoinverse of the Jacobian matrix $\mathbf{J}$;
3. $\mathbf{N} = \mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}$ is the projection into the null space of $\mathbf{J}$.

Equation (17) can be modified with the addition of the damped Jacobian from Eq. (16) and the corresponding corrective term:

$$\dot{\mathbf{q}} = \mathbf{J}^{*}\dot{\mathbf{x}}_c + (\mathbf{J}_{0p}\mathbf{N}^{*})(\dot{\mathbf{x}}_0 - \mathbf{J}_{0p}\mathbf{J}^{*}\dot{\mathbf{x}}_e) \tag{19}$$

where $\mathbf{N}^{*} = \mathbf{I} - \mathbf{J}^{*}\mathbf{J}$, $\dot{\mathbf{x}}_c = \dot{\mathbf{x}_e} + \mathbf{K}\mathbf{e}$ is the corrected end-effector velocity, $\mathbf{K}$ a positive-defined gain matrix and $\mathbf{e}$ is the position error between the desired position $\mathbf{x}_e$ and the actual position $\mathbf{x}$. The gain matrix $\mathbf{K}$ is defined as:

$$\mathbf{K} = k_e\mathbf{I} \tag{20}$$

where k_e is a scalar used as a tuning parameter. The effect of changing this parameter will be described in Section 4.

The first term of Eq. (19) guarantees the exact velocity of the end effector with minimum joints speed. The second term drives the motion of the point $\mathbf{P}_r$ of the robot, satisfying the collision avoidance additional task.

As already anticipated, two different operation modalities are presented:

1. **Full Mode: 6 and 7 DOF perturbation**: movements of the joints exploiting all DOFs are permitted. The joint velocity control law for the KUKA LBR is the one given by Eq. (19). For the UR5, instead, there is no null space and the two velocity tasks defined in Eq. (14) and (15) result in:

$$\dot{\mathbf{q}} = \mathbf{J}^{*}\left(\dot{\mathbf{x}}_e + \mathbf{K}\mathbf{e}\right) + \mathbf{J}^{*}{}_{0p}\dot{\mathbf{x}}_0 \tag{21}$$

2. **Reduced Mode: perturbation with fixed orientation**: only translations are admitted, whereas the orientation is fixed during the motion. The control law is:

$$\dot{\mathbf{q}} = \mathbf{J}^{*}\left(\dot{\mathbf{x}}_e + \mathbf{K}\mathbf{e}\right) + \mathbf{J}_R^{*}\begin{bmatrix}\dot{\mathbf{x}}_0 \\ \mathbf{0}_{3\times 1}\end{bmatrix} \tag{22}$$

The Jacobian matrix of the second term has a dimension of (6×6) and it is composed by the translation part of $\mathbf{J}_0$ and the orientation part of $\mathbf{J}$:

$$\mathbf{J}_R = \begin{bmatrix}\mathbf{J}_{0p} \\ \mathbf{J}_o\end{bmatrix} \tag{23}$$

The reduced mode is valid only for the UR5.

4 Experiments and Results

Several experimental tests have been conducted for both Universal Robot UR5 (Full Mode and Reduced Mode) and KUKA LBR iiwa 14 (Full Mode only) in order to verify the applicability of the strategy.

The sensor system used to track the obstacle's position is the OptiTrack PrimeX 22. In the following, the word "obstacle" will be used to indicate a human hand equipped with Optitrack markers. A fixed obstacle will be created by placing the hand statically in the scene before the robot's motion starts. A dynamic obstacle will be created by disturbing the robot's motion from different directions, moving toward its various joints.

4.1 Marker-based Obstacle Motion Tracking

The Optitrack is a high-performance MoCaP system based on infrared cameras detecting reflective markers, which can be tracked with sub-millimetric precision. The 360 FPS frame rate and the 2.2 MP resolution make the system efficient, ensuring robustness and accuracy. Nevertheless, proper placement of the cameras is needed to ensure the correct estimation of the 3D position of the markers. At least three cameras should always frame the markers in order to avoid low-quality estimation. In this work, 8 cameras have been placed to cover the area in which the UR5 robot is mounted, and 4 cameras have been used for the LBR iiwa. The cameras have been mounted on the ceiling to better capture the scene around the robots. For both scenarios, the calibration of the vision system has been performed. The calibration procedure involves the collection of thousands of samples from each camera at 360 FPS. An L-shape body with 3 markers is used to set the origin of the coordinate system. The point clouds are then aligned in OptiTrack's software Motive and the position of each camera is computed with respect to the origin. Once the calibration is complete, it is possible to create rigid bodies or skeletons and retrieve their position and orientation via software. For the proposed experiments, a rigid body has been created to fit on a glove worn by the operator. This allows both fixed and dynamic disturbance of the robot's motion with the hand as an obstacle. The homogeneous transformation T_O^B

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

from the robot's Base frame to the Optitrack origin frame can be easily measured. With this, the position of the obstacle can be expressed with respect to the robot's Base frame by:

$$\mathbf{P}_{ob}^{B} = \mathbf{T}_{O}^{B}\mathbf{P}_{ob}^{O} \tag{24}$$

The $\mathbf{P}_{ob}^{B}$ is used as the obstacle' position $\mathbf{P}_o$ by the Motion Planning and Redundancy Control algorithms: the closest point of the robot P_r to the obstacle $\mathbf{P}_o$ is computed and the distance $\mathbf{d}_O$ is calculated. The corresponding evasive motion in terms of joint velocities is then computed as explained in Section 3. For the experiments, Matlab 2021b software and the NatNet library provided by OptiTrack have been used. Firstly, the Optitrack system acquires the coordinates of the rigid body in real-time for each time step and sends them to the Matlab PC. The Optitrack software provides a client that allows communication with Matlab in order to acquire the coordinates of the rigid body. The acquired data are reported to the UR5 base reference system and used by the obstacle avoidance algorithm to identify the location of the obstacle in real-time. The last part of the communication chain, namely with the real robot, uses the TCP/IP protocol through an interface with Matlab. The socket connection is able to transmit the data at a frequency higher than 100 Hz. Thus, the two-way communication allows to read the joints' positions at each time step and to send the speed control command to the robot in real-time. The highest possible frequency is required in order to guarantee real-time control, fast signal acquisition and optimal communication with the sensors.

4.2 UR5 Test Cases

In this section, different examples with a moving obstacle interfering the trajectory of the manipulator are shown. The aim is to simulate a collaborative scene, therefore the robot is positioned on a table at a 0.7 m height from the ground, in a working area free of additional obstacles. In absence of disturbances, the robot is initially programmed to move from a initial point A and a final point B in a straight line, in a setup very similar to Fig. 1. In Fig. 9, the joint positions and velocities for this motion are plotted for reference. Additionally, a 3D-printed passive gripper with a length of 210 mm has been attached to the robot's flange, exploiting the easy adaptation of the avoidance algorithms, which only requires modification of the **G** control point coordinates for the UR5 (or the **E** control point for the KUKA).

4.2.1 Test 1 - Moving obstacle, Full Mode

In this example, a moving obstacle interferes with the end-effector and other parts of the manipulator. Figure 10 shows the obstacle coming from different directions (front, under and side) and the respective reaction of the manipulator during the evasive motion. The test is executed using the Full Mode, which allows the manipulator to exploit its full kinematic chain to avoid the obstacle.

Figure 11 shows the robot's behavior in terms of its joint positions and velocities in time. In particular, it can be noticed from the velocities plot how the robot quickly reacts and triggers all the joints. The magnitude of the induced joint velocity change is fairly low (maximum of 0.6 rad/s), resulting in a smooth evasive motion, as confirmed by the joint positions plot. Figure 12 instead shows the robot's trajectory in terms of the cartesian position and orientation of the Tool Center Point (TCP), compared to the trajectory that the robot would have without any obstacle. Figure 12 also reports the angle relative to the orientation given by the fifth joint (which corresponds to the β Euler angle) and the euclidean distance between obstacle and TCP. The former is used to confirm the most visible difference between Full Mode and Reduced Mode. Refer to the figure's caption for more comments on the analyzed data.

The safety region below which no action is possible, introduced in Section 3.3, is illustrated in orange. Referring to Eq. (21), the activation factor a_v will depend on the tunable radius r, which defines the region of influence of a control point, in this case the TCP. A small radius limits the influence area, making it challenging for the algorithm to handle cases where obstacles might interfere. On the other hand, a larger radius is better for swiftly moving obstacles, while small

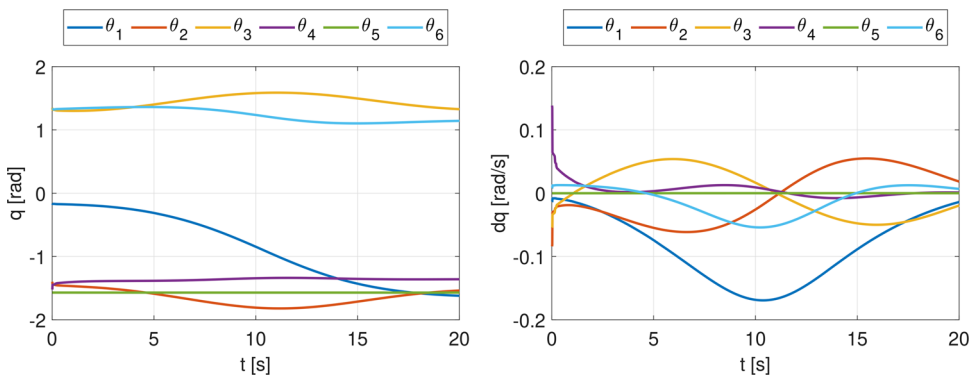

Fig. 9 Joint angles and velocities during the reference trajectory without obstacle. For the programmed motion, joint 5 does not need to move

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

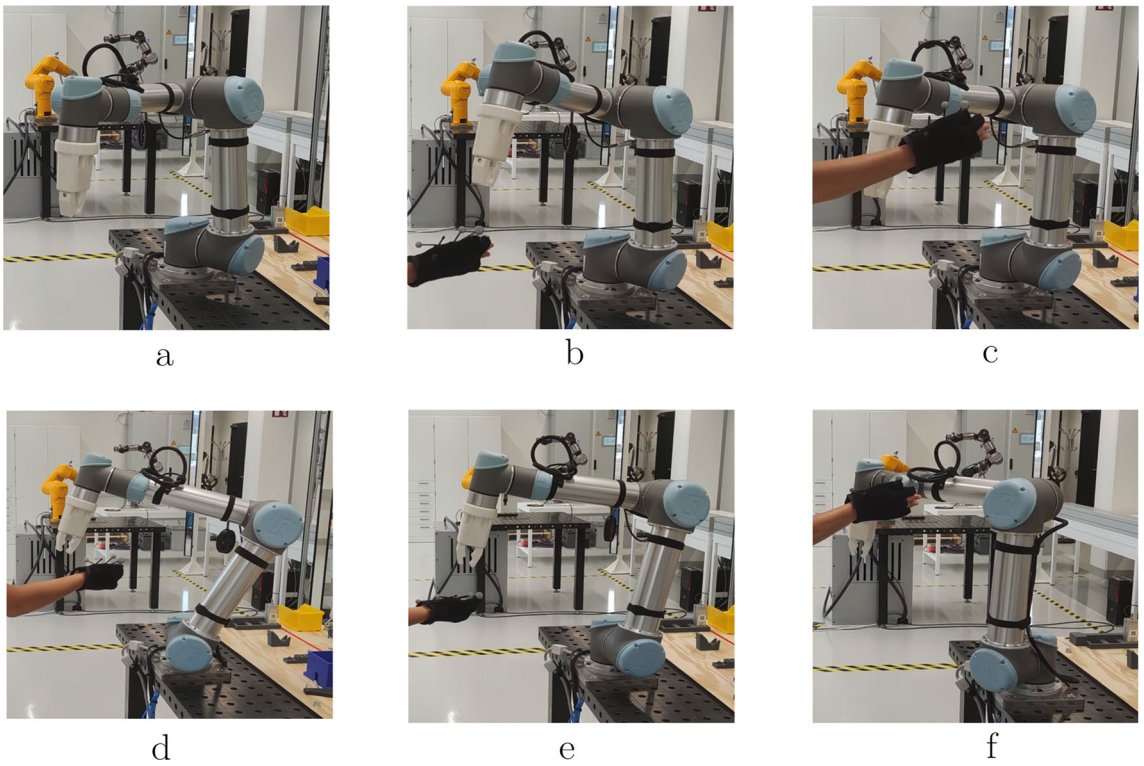

Fig. 10 **Test 1.** Avoidance of a moving obstacle in Full Mode: the end-effector orientation is not preserved during the evasive motion

regions suffice for fixed obstacles [3, 4, 43]. For these tests, the computation of the maximum allowed distance between TCP and obstacle, resulted in a value of roughly 16 cm.

4.2.2 Test 2 - Moving Obstacle, Reduced Mode, Gripper

In this example, the robot's motion is again disturbed with a moving obstacle. This time, however, the Reduced Mode is used. With respect to the previous case, it can be seen that Joint 5 is kept at a constant angle, i.e., the end-effector does not tilt with respect to the vertical direction, as visible in Fig. 13. This type of motion keeps the vertical alignment of the TCP, which could be beneficial for some operations, but it could require higher joint velocities. This can be seen in Fig. 14, which shows the corresponding joint rotations and velocities. Both graph correctly shows that Joint 5 never moves and has therefore null velocity. The same behavior is clearly visible in Fig. 15 which reports the cartesian coordinates, the obstacle motion and the distance TCP-Obstacle, which is again kept above the safety region.

4.2.3 Test 3 - Simple Motion, Reduced Mode

In this test, a simple vertical motion that resembles a pick-and-place task is programmed on the robot. The aim of this experiment is to visualize the response to one-directional disturbances, especially from the joint velocity point of view. The TCP is disturbed with an obstacle moving along the

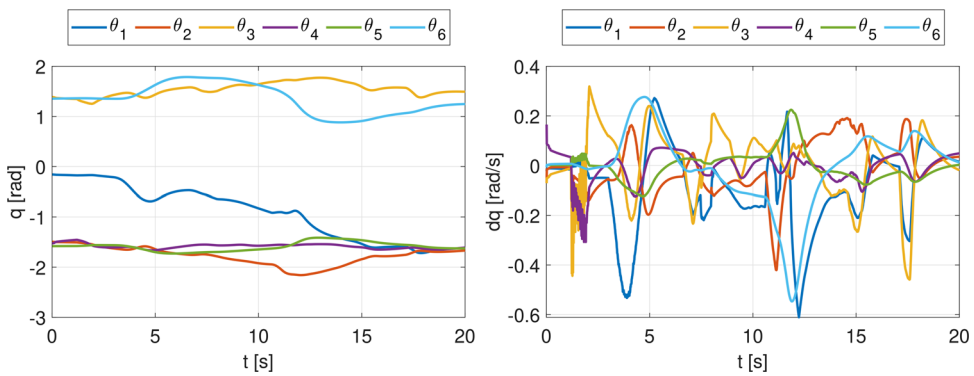

Fig. 11 **Test 1.** Joint rotations (left) and joint velocities (right) with a moving obstacle in Full Mode. All joints are triggered in order to perform the evasive motion efficiently

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

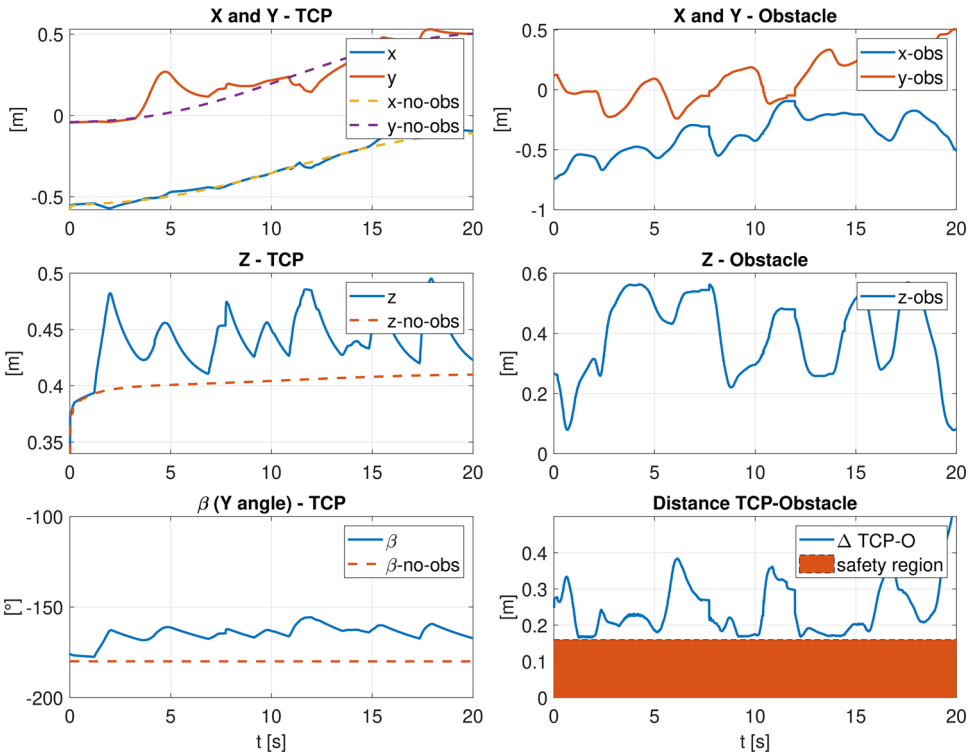

Fig. 12 **Test 1.** Cartesian coordinates and distance obstacle-flange with a moving obstacle in Full Mode. The obstacle is moved in such a way that most evasions happen along the vertical direction, as clearly visible from the Z graphs. The reader should remember that the proposed avoidance algorithm considers the closest point of the kinematic chain to the obstacle's position, which means that the avoidance is not limited to the end-effector. However, both the cartesian representation and the TCP-Obstacle distance information can help in visualizing the evasive motion with respect to the obstacle-free trajectory, and can be analyzed in parallel to the joint space information in Fig. 11. Furthermore, it is noticeable how the obstacle never enters the safety region of the manipulator

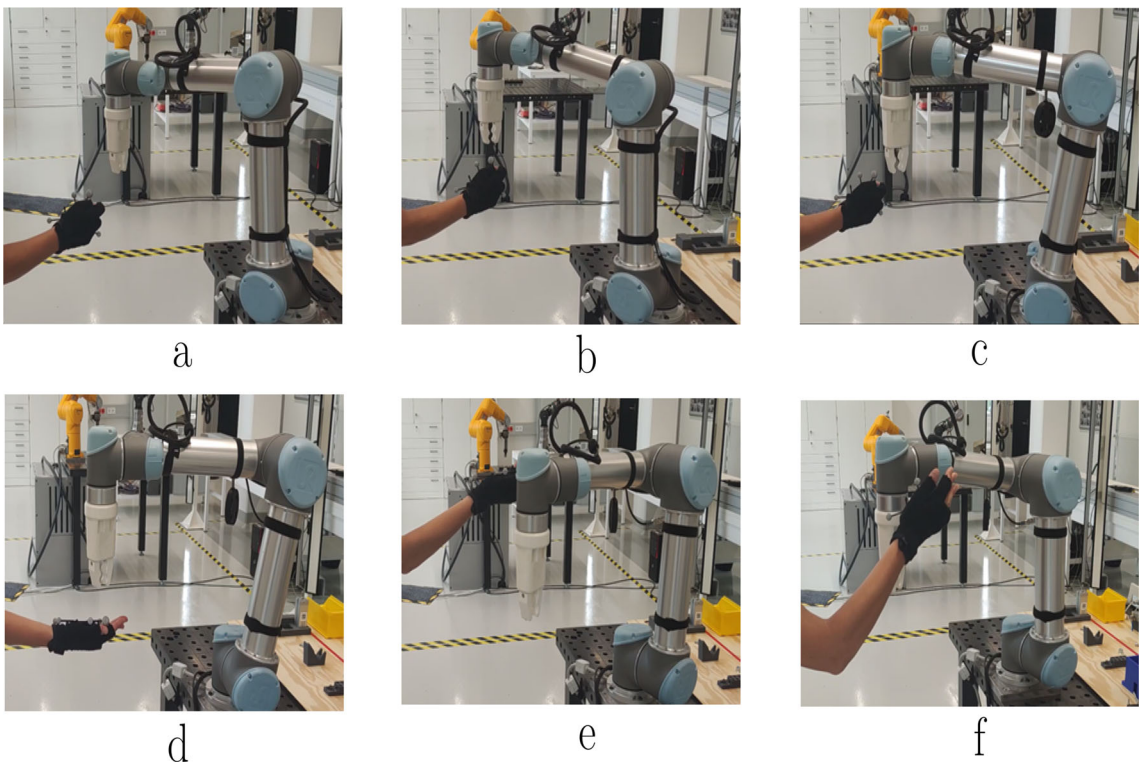

Fig. 13 **Test 2.** Avoidance of a moving obstacle with the tool in Reduced Mode. The end-effector keeps a straight vertical orientation

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

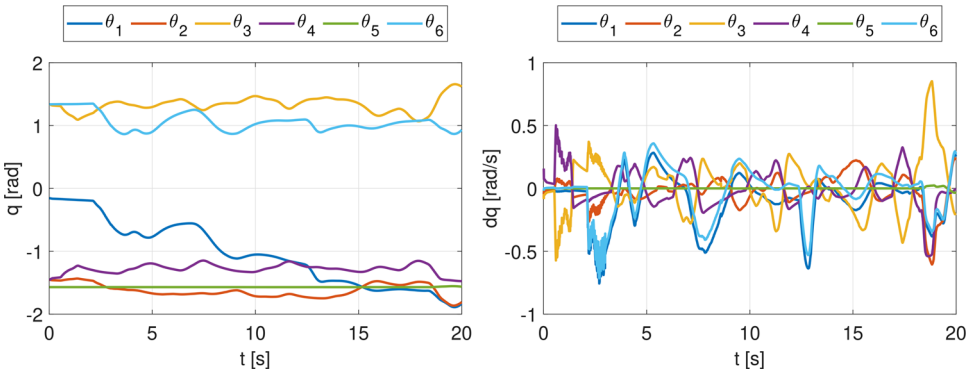

Fig. 14 **Test 2.** Joint rotations and joint velocities in Reduced Mode: avoidance of a moving obstacle. Since the orientation is constrained, the fifth joint does not move, as in the obstacle-free trajectory. As a result, the vertical angle is maintained, but the rest of the joints have to compensate with more motion

vertical direction, from below and above the end-effector. In Fig. 16, the graph of dq shows periodic accelerations, which are still low in magnitude, but are clearly differentiable with respect to previous cases. The evasive motion is more visible in the cartesian plots of Fig. 17. The X and Y coordinates are left almost unaltered, as the obstacle mainly moves up and down, disturbing the TCP. In particular, it can be seen how the TCP trajectory moves above and below the corresponding obstacle-free trajectory, according to the direction of the obstacle motion within the Z axis.

4.3 KUKA LBR iiwa 14 Test Cases

Several tests were also conducted for the KUKA LBR iiwa 14.The examples help to understand the behavior of the previously presented control law under different conditions. A fixed-obstacle case is also described. Finally, three examples are discussed to demonstrate the behavior of the algorithm as the k_e parameter, introduced in Eq. 20, change.

The robot is programmed to move linearly in the horizontal plane, with a speed of 0.25 m/s. Figure 18 shows the joint angles and the joint velocities in the absence of any obstacle. Note that, for this motion, the joint velocities for Joints 1, 2 and 7 are generally higher than in the UR5 case.

4.3.1 Test 4 - Fixed Obstacle, Full Mode

In this test, there is a fixed obstacle in the middle of the end-effector's path. Figure 19 shows six steps of the motion: when the obstacle reaches the region of influence of the end-effector of the robot, the control reacts by keeping the obstacle out of the safety sphere with radius r (compare

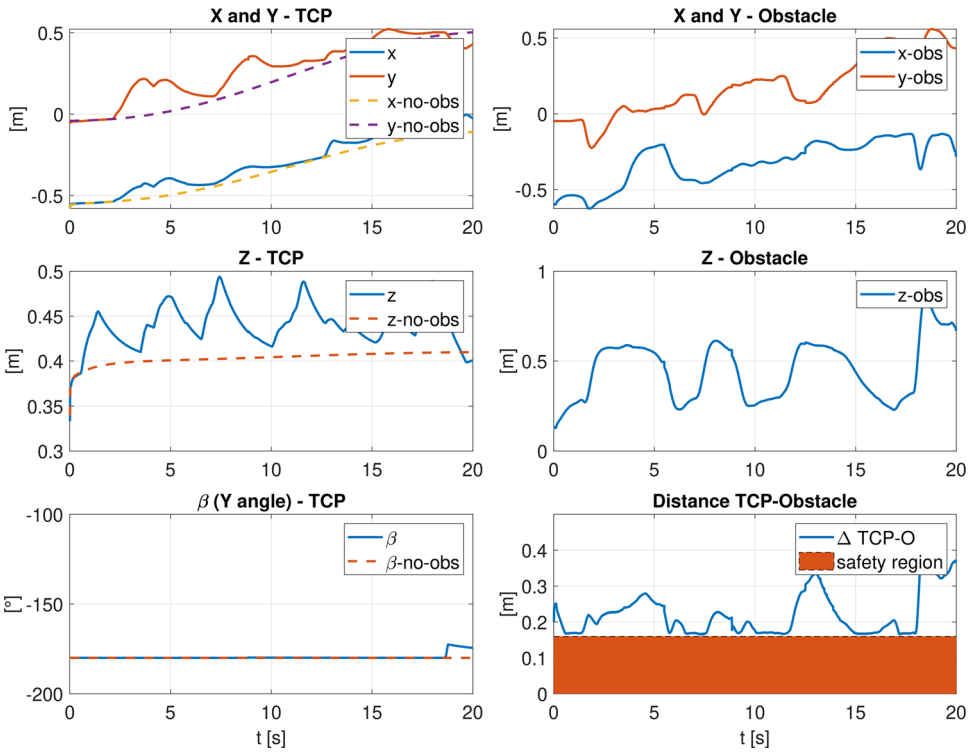

Fig. 15 **Test 2.** Cartesian coordinates and distance TCP-obstacle with a moving obstacle in Reduced Mode. As expected, the β angle is constant. The maximum allowed distance ΔTCP-Obstacle is unchanged with respect to the Full Mode case

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

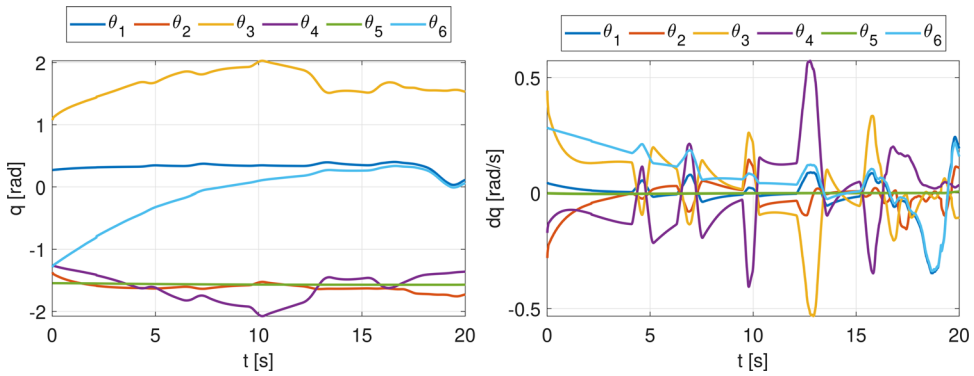

Fig. 16 **Test 3.** Joint rotations and joint velocities in Reduced Mode: avoidance of a moving obstacle

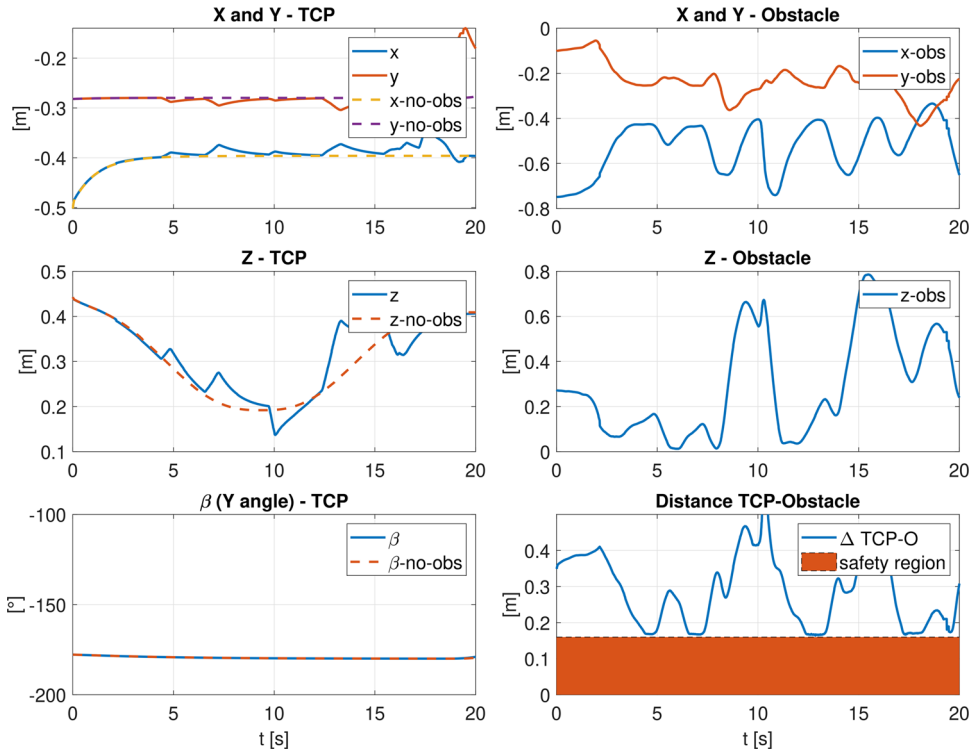

Fig. 17 **Test 3.** Cartesian coordinates and distance obstacle-flange with a moving obstacle in Reduced Mode for the simple motion test. Most of the change happens along the Z direction, while the β angle is again kept constant

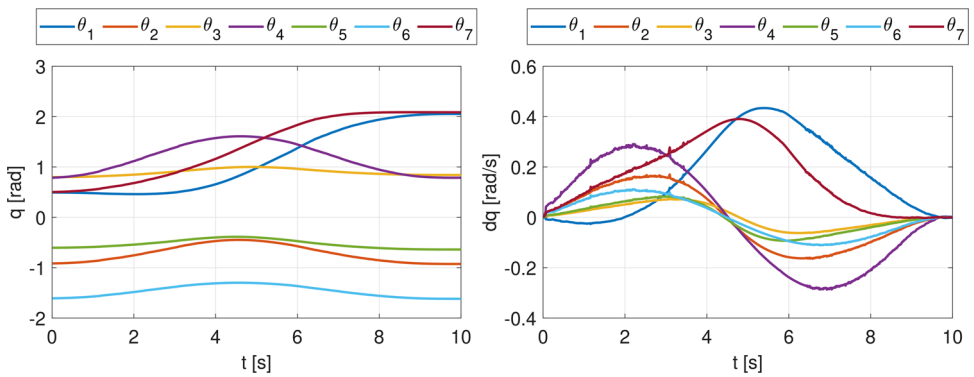

Fig. 18 **KUKA LBR.** Joint rotations and joint velocities without obstacle

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

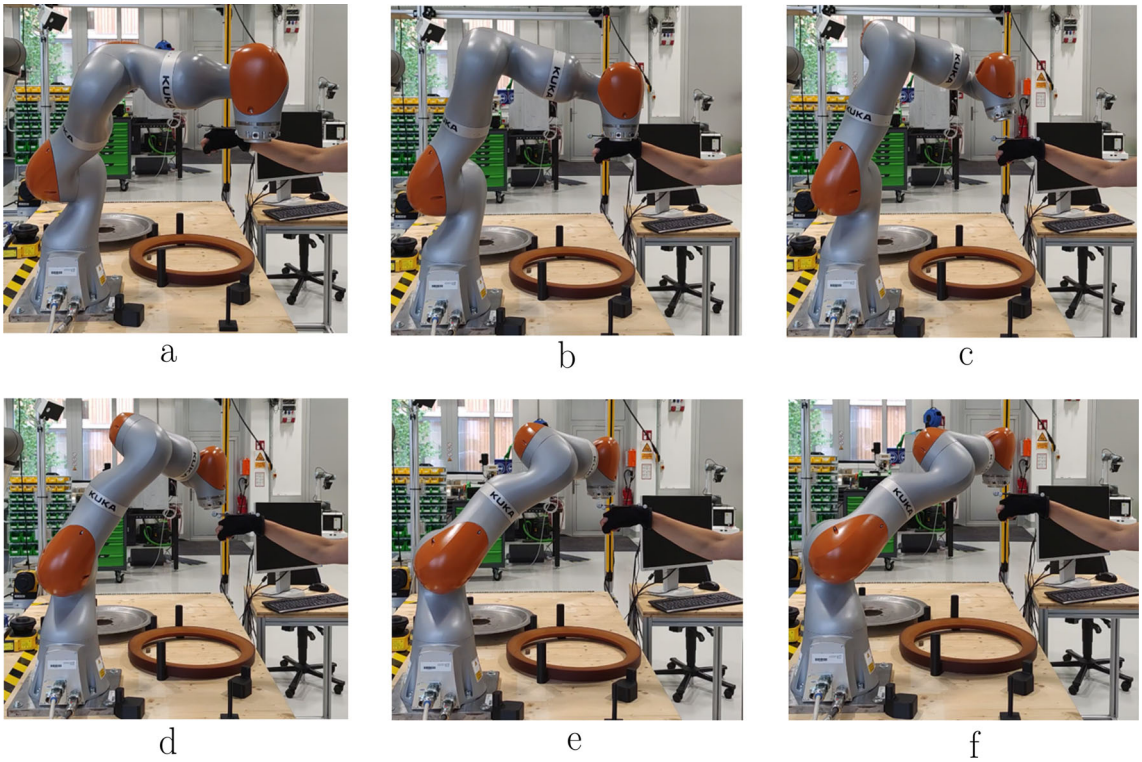

Fig. 19 **Test 4**. Avoidance of a fixed obstacle

Fig. 7). In Fig. 20 the joint rotations and velocities are reported. It can be noticed that, despite some discontinuities also due to the sampling rate, the velocity trajectories are generally smoother than in the experiments with the UR5, as no spikes are visible. The absolute values of the velocities are also kept within safe limits ($< 50°/s$). As a result, also thanks to the robot's redundancy, the joint angles experience overall small changes.

4.3.2 Test 5 - Moving Obstacle - Full Mode

In this example a moving obstacle interferes with the end-effector and the internal points of the manipulator. Figure 21 shows six steps of motion. The risk of a collision can occur when the obstacle interferes with an intermediate point in the link of the manipulator, or if the manipulator itself obstructs the vision system. For this reason, the cameras were arranged in such a way that at least three cameras always track the obstacle. Thanks to the redundancy of the kinematic chain, the control can reconfigure the manipulator without changing too much the end-effector pose, one of the intended goals explained in Section 3.

4.3.3 Test 6 - Moving Obstacle - Varying k_e

The last three examples deal with the variations of the term k_e. It is interesting to note how, even for small variations of the term with the same conditions, the control algorithm

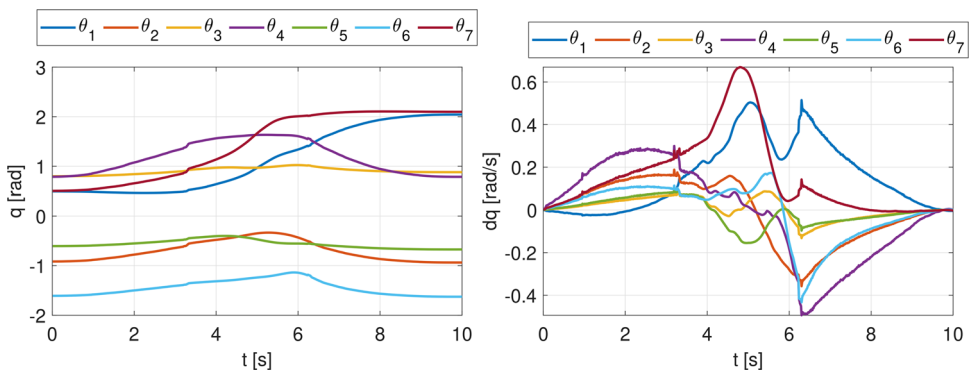

Fig. 20 **Test 4.** KUKA Joint rotations and joint speeds with a fixed obstacle

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

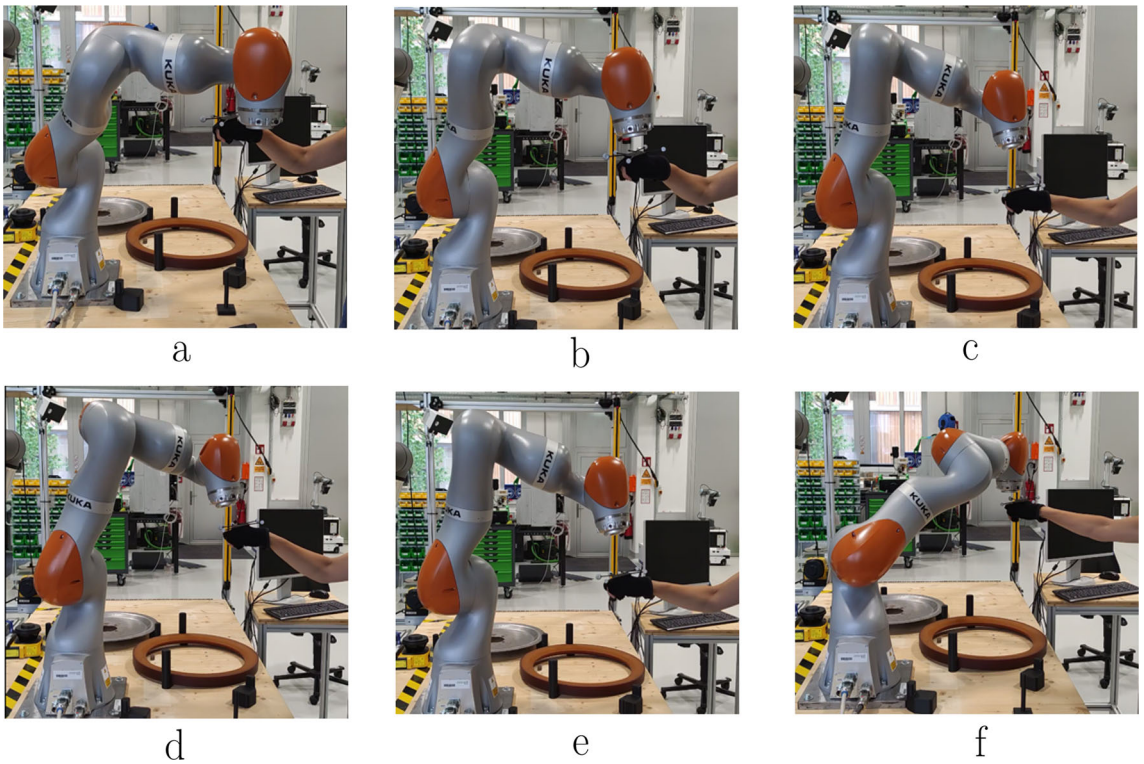

Fig. 21 **Test 5**. Avoidance of a moving obstacles with the end-effector and the kinematic chain of the manipulator

has different behavior. Clearly, the position and movement of the obstacle may be slightly different in the three cases. Howvever, small increases of the k_e, result in more jerky movements of the manipulator. This result is visible from the graphs in Figs. 23, 24 and 25.

5 Discussion

The experimental tests presented in this paper successfully demonstrates the correctness of the method and its applicability in a real system. The combination of off-line path planning and on-line motion control was previously tested in simulation. With this work, real use cases, including more realistic and complete obstacle motions have been analyzed. Similarly, it was possible to assess that the joint velocities don't increase close to potentially dangerous values. In particular, the KUKA LBR iiwa 14 presents a better response in terms of the smoothness of the joint velocities. Although both modes used in the UR 5 show similar acceleration and computational times, there are no significant differences between them. The peculiarity of the UR5 modalities lies in the possibility to select the appropriate control mode based on the specific application to be executed. If there is a possibility of a dangerous instrument being moved, it is advisable to maintain a fixed orientation to avoid collisions or exposure to the human operator's face. Conversely, if the orientation of the end-effector is not crucial, using the Full Mode can

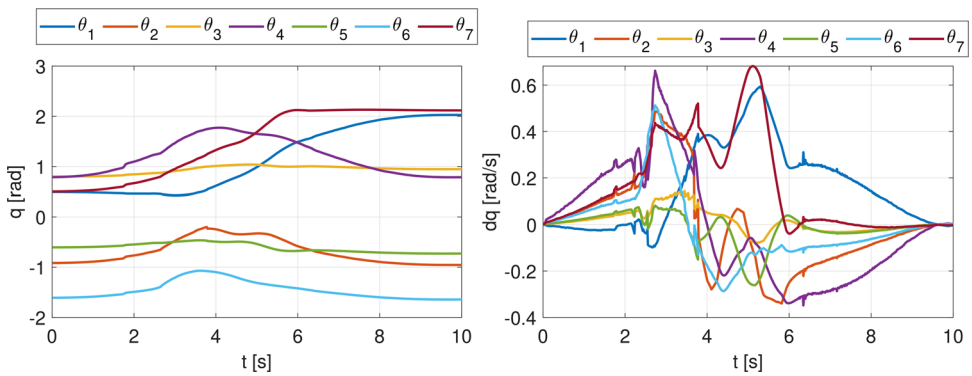

Fig. 22 **Test 5**. Joint rotations and joint speeds with a mobile obstacle

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

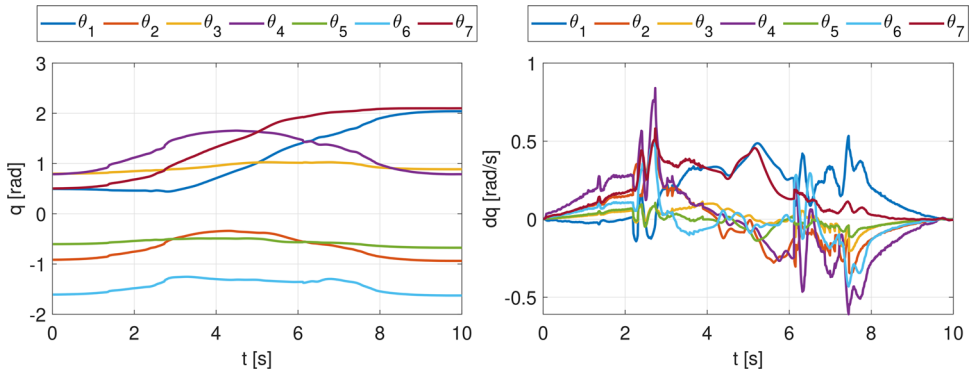

Fig. 23 **Test 6**. Joint rotations and joint speeds with $k_e = 10$

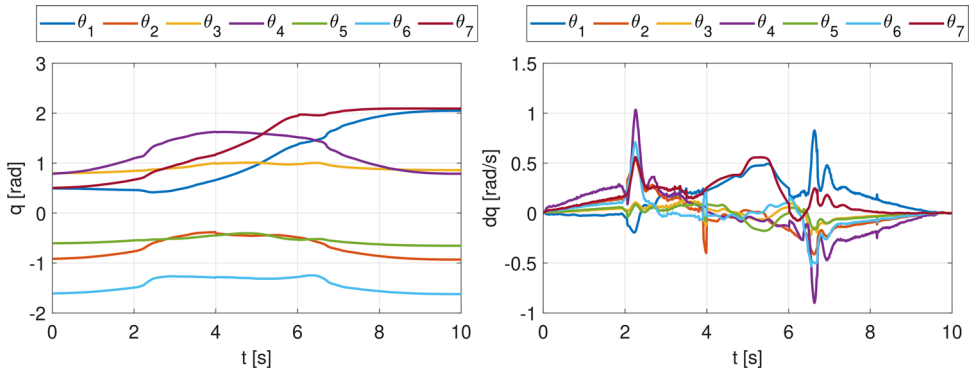

Fig. 24 **Test 6**. Joint rotations and joint speeds with $k_e = 12$

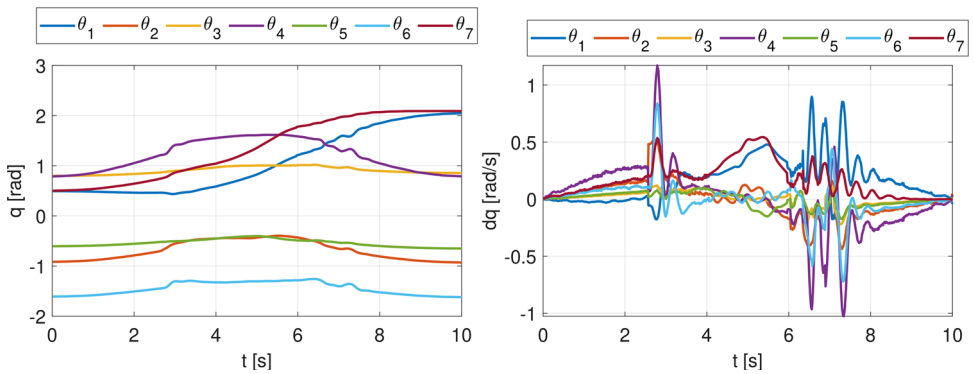

Fig. 25 **Test 6**. Joint rotations and joint speeds with $k_e = 15$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

enhance collision avoidance. This is because the additional DOF available for creating repulsive movements help avoid obstacles in general.

The MoCaP system can accurately acquire the position of objects or humans in the workspace at a high frame rate. However, the presented obstacle avoidance strategy can work given any tracking estimator, although precise measurements with errors in the order of a few centimeters are preferable. The robot can be disturbed from any direction, provided that the position estimation of the obstacle doesn't fail. The proposed obstacle avoidance strategy is a preferred solution in HRC for use cases where the standard safety measures would disrupt the robot's operation. The Reduced Mode can be exploited for line-following or reduced pick-and-place tasks; the Full Mode can overcome the limitations due to the robot's dimension in case of excessive avoidance needed in certain cases.

6 Conclusions

In this paper, an obstacle avoidance strategy both for UR manipulators and KUKA LBR iiwa 14 moving in dynamically varying environments is presented and verified through experimental testing. The experimental tests presented in thiw work demonstrates the correctness of the method and its applicability in a real system. In prior work, the interplay between offline path planning and online motion control was examined through a simulation environment. This study advances the research by delving into practical applications, which involve more intricate and authentic obstacle dynamics and tracking sensors. Furthermore, it has been confirmed that joint velocities remain at safe levels, even in the vicinity of potentially perilous obstacles. Two different operating modes, Full Mode and Reduced Mode, are proposed and validated for the UR manipulator. The single mode presented for the KUKA LBR iiwa gives in general better responses in terms of the smoothness of the joint velocities. A motion capture system Optitrack Prime X22 is used for the obstacle detection with extreme accuracy. The adaptive control approach allows the robot to dynamically adjust control parameters in response to changes in the environment, e.g., the obstacle's trajectory. This addresses challenges related to uncertainties in the environment and variations in the robot's dynamics.

7 Future work

Analyzing the stability and uncertainty of systems in collision avoidance applications necessitates a comprehensive strategy. This involves integrating advanced control strategies, probabilistic reasoning, fault tolerance, and adaptive mechanisms. Such an interdisciplinary perspective is vital for the development of collision avoidance systems proposed in this work that can operate securely and efficiently within the inherently uncertain and dynamic environments encountered by robots. Addressing human-robot interaction is crucial. Specifically, the goal is to ensure that the strategy is adaptable to accommodate human behavior in shared spaces, enhancing the robot's capacity to engage with humans in a safe and predictable manner. For this reason, among future developments, an analysis of the uncertainty and stability of the system will be considered. Sensor fusion techniques can improve the accuracy and reliability of the robot's perception of its environment. An additional future development involves implementing the strategy on different manipulators using both motion capture systems and vision systems to obtain real-time data. Integrating information from multiple sensors will allow obtaining a higher accuracy of the surrounding environment.

Author Contributions Conceptualization, C.S and B.U.; methodology, C.S., B.U. and G.P.; software, C.S and B.U.; validation, C.S. and B.U.; formal analysis, C.S.; investigation, G.P; resources, M.H.; data curation, C.S.; writing-original draft preparation, C.S. and B.U.; writing-review and editing, G.P.; visualization, M.R.; supervision, G.P. and H.R.; project administration, M.H. All authors have read and agreed to the published version of the manuscript.

Funding Open access funding provided by Universitá Politecnica delle Marche within the CRUI-CARE Agreement. This research has received funding from the "Karntner Wirtschaftsf orderung Fonds" (KWF) and the "European Regional Development Fund" (EFRE) within the PATTERN-Skin project 26616/ 34294/49769.

Availability of data and materials No publicly archived dataset has been used or generated during the research.

Code availability Not applicable.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication The authors declare their consent for publication.

Conflict of interest/Competing interests Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. ISO 10218-2:2011-07 Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration, Geneva, Switzerland (2011)
2. ISO/TS 15066:2016-02 Robots and robotic devices – Collaborative robots, Geneva, Switzerland (2016)
3. Palmieri, G., Scoccia, C.: Motion planning and control of redundant manipulators for dynamical obstacle avoidance. Machines **9**(6), 121 (2021)
4. Chiriatti, Giorgia, Palmieri, Giacomo, Scoccia, Cecilia, Palpacelli, Matteo Claudio, Callegari, Massimo: Adaptive obstacle avoidance for a class of collaborative robots. Machines **9**(6), 113 (2021)
5. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Autonomous robot vehicles, pp. 396–404. Springer, ??? (1986)
6. Lozano-Perez, Tomas: A simple motion-planning algorithm for general robot manipulators. IEEE J. Robot. Autom. **3**(3), 224–238 (1987)
7. Scalera, L., Giusti, A., Vidoni, R., Gasparetto, A.: Enhancing fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones. Int. J. Adv. Manuf. Tech. **121**(9), 6783–6798 (2022)
8. Scalera, L., Vidoni, R., Giusti, A.: Optimal scaling of dynamic safety zones for collaborative robotics. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 3822–3828 (2021). IEEE
9. Lingelbach, F.: Path planning using probabilistic cell decomposition. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 1, pp. 467–472 (2004). IEEE
10. Gonzalez, R., Kloetzer, M., Mahulea, C.: Comparative study of trajectories resulted from cell decomposition path planning approaches. In: 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), pp. 49–54 (2017). IEEE
11. Wang, J., Liu, S., Zhang, B., Yu, C.: Manipulation planning with soft constraints by randomized exploration of the composite configuration space. Int. J. Control Autom. Syst. **19**(3), 1340–1351 (2021)
12. Xu, P., Wang, N., Dai, S.L., Zuo, L.: Motion planning for mobile robot with modified bit* and mpc. Appl. Sci. **11**(1), 426 (2021)
13. Shitsukane, A.S.: Fuzzy Logic Model for Obstacles Avoidance Mobile Robot in Static Unknown Environment. PhD thesis, JKUAT-COETEC (2022)
14. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Mach. Stud. **7**(1), 1–13 (1975). https://doi.org/10.1016/S0020-7373(75)80002-2
15. Wenzel, P., Schön, T., Leal-Taixé, L., Cremers, D.: Vision-Based Mobile Robotics Obstacle Avoidance With Deep Reinforcement Learning. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 14360–14366 (2021) https://doi.org/10.1109/ICRA48506.2021.9560787
16. Zindler, F., Lucchi, M., Wohlhart, L., Pichler, H., Hofbaur, M.: Towards Dynamic Obstacle Avoidance for Robot Manipulators with Deep Reinforcement Learning. In: Müller, A., Brandstötter, M. (eds.) Advances in Service and Industrial Robotics, pp. 89–96. Springer, Cham (2022)
17. Gharbi, A.: A dynamic reward-enhanced Q-learning approach for efficient path planning and obstacle avoidance in mobile robotics. Appl. Comput. Inform. (2024)
18. Liu, A., Fu, J., Zhan, S., Jin, Z., Zhang, W.: A Policy Searched-Based Optimization Algorithm for Obstacle Avoidance in Robot Manipulators. IEEE Trans. Ind. Electron. 1–10 (2024) https://doi.org/10.1109/TIE.2023.3344831
19. Farag, K.K.A., Shehata, H.H., El-Batsh, H.M.: Mobile robot obstacle avoidance based on neural network with a standardization technique. J. Robot. **2021**, (2021)
20. Zohaib, M., Pasha, M., Riaz, R.A., Javaid, N., Ilahi, M., Khan, R.: Control Strategies for Mobile Robot With Obstacle Avoidance **3**, 1027–1036 (2013)
21. Sani, M., Robu, B., Hably, A.: Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control. In: IECON 2021 - 47th Annual Conference of the IEEE Industrial Electronics Society, 1–6 (2021) https://doi.org/10.1109/IECON48115.2021.9589658
22. Adamu, P.I., Okagbue, H.I., Oguntunde, P.E.: Fast and optimal path planning algorithm (FAOPPA) for a mobile robot. Wirel. Pers. Commun. **106**, 577–592 (2019)
23. Kumar, S., Dadas, S.S., Parhi, D.R.: Path planning of mobile robot using modified DAYKUN-BIP virtual target displacement method in static environments. Wirel. Pers. Commun. **128**(3), 2287–2305 (2023)
24. Gasparetto, A., Zanotto, V.: A new method for smooth trajectory planning of robot manipulators. Mech. Mach. Theory **42**(4), 455–471 (2007)
25. Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R.: Path planning and trajectory planning algorithms: A general overview. Motion and operation planning of robotic systems 3–27 (2015)
26. Du, Y., Zhang, X., Nie, Z.: A Real-Time Collision Avoidance Strategy in Dynamic Airspace Based on Dynamic Artificial Potential Field Algorithm. IEEE Access **7**, 169469–169479 (2019). https://doi.org/10.1109/ACCESS.2019.2953946
27. Puriyanto, R.D., Wahyunggoro, O., Cahyadi, A.I.: Implementation of Improved Artificial Potential Field Path Planning Algorithm in Differential Drive Mobile Robot. In: 2022 14th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 18–23 (2022) https://doi.org/10.1109/ICITEE56407.2022.9954079
28. Feng, S., Qian, Y., Wang, Y.: Collision avoidance method of autonomous vehicle based on improved artificial potential field algorithm. Proc. Inst. Mech. Eng. Pt. D J. Automobile Eng. **235**(14), 3416–3430 (2021)
29. Xu, X., Hu, Y., Zhai, J., Li, L., Guo, P.: A novel non-collision trajectory planning algorithm based on velocity potential field for robotic manipulator. Int. J. Adv. Robot. Syst. **15**(4), 1729881418787075 (2018)
30. Halme, R.J., Lanz, M., Kämäräinen, J., Pieters, R., Latokartano, J., Hietanen, A.: Review of vision-based safety systems for human-robot collaboration. Procedia CIRP **72**, 111–116 (2018)
31. Schmidt, B., Wang, L.: Depth camera based collision avoidance via active robot control. J. Manuf. Syst. **33**(4), 711–718 (2014)
32. Fabrizio, F., De Luca, A.: Real-time computation of distance to dynamic obstacles with multiple depth sensors. IEEE Robot. Autom. Lett. **2**(1), 56–63 (2016)
33. Flacco, F., Kröger, T., De Luca, A., Khatib, O.: A depth space approach to human-robot collision avoidance. In: 2012 IEEE international conference on robotics and automation, pp. 338–345. (2012). IEEE
34. Rybski, P., Anderson-Sprecher, P., Huber, D., Niessl, C., Simmons, R.: Sensor fusion for human safety in industrial workcells. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Sys-

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

tems, pp. 3612–3619 (2012) https://doi.org/10.1109/IROS.2012.6386034
35. Morato, C., Kaipa, K.N., Zhao, B., Gupta, S.K.: Toward safe human robot collaboration by using multiple kinects based real-time human tracking. J. Comput. Inf. Sci. Eng. **14**(1) (2014)
36. Wang, L., Schmidt, B., Nee, A.Y.: Vision-guided active collision avoidance for human-robot collaborations. Manuf. Lett. **1**(1), 5–8 (2013)
37. Jiang, D., Li, G., Sun, Y., Hu, J., Yun, J., Liu, Y.: Manipulator grabbing position detection with information fusion of color image and depth image using deep learning. J. Ambient Intell. Humaniz. Comput. **12**, 10809–10822 (2021)
38. Thangaraj, M., Monikavasagom, S.: A competent frame work for efficient object detection, tracking and classification. Wirel. Pers. Commun. **107**, 939–957 (2019)
39. Li, X.: Robot target localization and interactive multi-mode motion trajectory tracking based on adaptive iterative learning. J. Ambient Intell. Humaniz. Comput. **11**, 6271–6282 (2020)
40. Kim, J., Jung, H., Kang, M., Chung, K.: 3D human-gesture interface for fighting games using motion recognition sensor. Wirel. Pers. Commun. **89**, 927–940 (2016)
41. Amorim, A., Guimares, D., Mendona, T., Neto, P., Costa, P., Moreira, A.P.: Robust human position estimation in cooperative robotic cells. Robot. Comput.-Integr. Manuf. **67**, 102035 (2021)
42. Heredia, J., Cabrera, M.A., Tirado, J., Panov, V., Tsetserukou, D.: Cobotgear: Interaction with collaborative robots using wearable optical motion capturing systems. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pp. 1584–1589 (2020). IEEE
43. Scoccia, C., Palmieri, G., Palpacelli, M.C., Callegari, M.: A collision avoidance strategy for redundant manipulators in dynamically variable environments: on-line perturbations of off-line generated trajectories. Machines **9**(2), 30 (2021)
44. Ahmadi, K., Asadi, D., Merheb, A., Nabavi-Chashmi, S.Y., Tutsoy, O.: Active fault-tolerant control of quadrotor UAVs with nonlinear observer-based sliding mode control validated through hardware in the loop experiments. Control Eng. Pract. **137**, 105557 (2023). https://doi.org/10.1016/j.conengprac.2023.105557
45. Moe, S., Antonelli, G., Teel, A.R., Pettersen, K.Y., Schrimpf, J.: Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results. Front. Robot. AI. **3**, 16 (2016)
46. Chiaverini, S., Siciliano, B., Egeland, O.: Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. IEEE Trans. Control Syst. Technol. **2**(2), 123–134 (1994)
47. Melchiorre, M., Scimmi, L.S., Pastorelli, S.P., Mauro, S.: Collison avoidance using point cloud data fusion from multiple depth sensors: a practical approach. In: 2019 23rd International Conference on Mechatronics Technology (ICMT), pp. 1–6 (2019). IEEE
48. Lee, K.K., Buss, M.: Obstacle avoidance for redundant robots using Jacobian transpose method. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3509–3514 (2007). IEEE
49. Zlajpah, L., Nemec, B.: Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.2, pp. 1898–1903 (2002) https://doi.org/10.1109/IRDS.2002.1044033
50. Swarup, A., Gopal, M.: Control strategies for robot manipulators-a review. IETE J. Res. **35**(4), 198–207 (1989)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Cecilia Scoccia graduated in A.A. 2017/2018 in Mechanical Engineering at Polytechnic University of Marche, Ancona, Italy. From July 2018 to February 2019, she spent a period abroad at Siemens PLM Software, Belgium, in the automotive research department. Since 2019, she is Research Fellow at the Department of Industrial Engineering and Mathematical Sciences (DIISM). From January 2022 to June 2022, she worked with the JOANNEUM Research in Austria as a visiting scientist. In March 2023 she received the PhD degree in Industrial Engineering at DIISM and she currently holds the position of researcher in the same department. Her research interests include collaborative robotics and human-robot collaboration, mechatronics, and biomechanics.

Barnaba Ubezio received the bachelor's degree in electronic engineering and the Master's Degree in Mechatronic Engineering in 2016 and 2018, respectively, both at Politecnico di Torino, Italy. After three years as a Robotics Research Engineer at JOANNEUM Research, Klagenfurt, Austria, he is currently pursuing a PhD in the topic of Radar Sensors for Robotics applications at the University of Klagenfurt, Austria. His research interest are mainly directed at radar and other proximity sensors for signal processing and simulation.

Giacomo Palmieri graduated in 2006 in Mechanical Engineering at Polytechnic University of Marche, Ancona, Italy. In 2010 he received the PhD degree in Mechanical Engineering at Polytechnic University of Marche. From 2010 to 2016, he was a researcher at eCampus University, Novedrate (CO), Italy. From 2016 to 2019, he was a researcher at the Department of Industrial Engineering and Mathematical Sciences (DIISM) of the Polytechnic University of Marche, where he currently holds the position of associate professor. He is scientific head of the robotics section of the i-Labs Industry Laboratory, Jesi (AN), operating in the field of Industry 4.0 technologies.

Michael Rathmair studied electrical engineering and information technology at the Technical University of Vienna (TU Wien) and received his Ph.D. from the Institute of Computer Engineering in 2018. In addition, he worked as a research associate in various national and international research projects and university teaching. Since March 2019, he has been working as a post-doc and technical experiment manager of the ROBOTICS Evaluation Lab (REL) at the Institute of ROBOTICS of JOANNEUM RESEARCH Forschungsgesellschaft GmbH. In October 2021, he took over the leadership of the Industrial Robotics System Technologies research group. Since April 2023, he has additionally assumed the position of Deputy Director of the Institute ROBOTICS. His professional interests lie primarily in researching and developing robotic system technologies for applications in a flexible cyber-physical system. In order to be able to integrate these processes into modern everyday production optimally. He is also involved in the national and international committees for the robot safety standard ISO 10218.

Michael Hofbaur is professor for Modular Robotics at University of Klagenfurt and founder / CEO of riskaware-X, a deeptech startup providing solutions and services for performance enabling robot safety. Prior to this activity he was CTO and Director of the applied research institute JOANNEUM RESEARCH ROBOTICS and held various faculty positions in the field of Automation, Control and Robotics in Austria. In addition, he was visiting professor and visiting scholar at M.I.T./USA working at the interface between Systems Engineering, Systems Safety, Robotics and Artificial Intelligence and worked as Chartered Engineer for Systems Engineering Functional Safety.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com