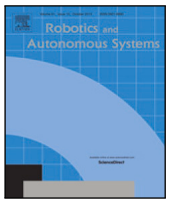
Robotics and Autonomous Systems 200 (2026) 105401

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Reactive grasp controller for dynamic grasping of moving objects

Emilio Maranci, Carlo Alberto Avizzano, Salvatore D'Avella *

Department of Excellence in Robotics & AI, Mechanical Intelligence Institute, Scuola Superiore Sant'Anna, Pisa, Italy

ARTICLE INFO

Keywords:
Reactive and sensor-based grasp planning
Dynamic grasping
Human–robot collaboration
Industry 5.0

ABSTRACT

Robots are not yet ready to handle unstructured and uncertain environments by adapting their trajectory online depending on the changes in the surroundings. Knowing how to handle moving objects is an essential skill for robots in many industrial and service robotics applications. This work presents a reactive grasp controller inspired by an established human motion model. It starts planning by generating a straight line from the current pose of the end-effector to the target object pose. Then, the line is shaped considering the new target position. A trace controller guarantees that the end-effector remains on the path, and the collision-aware velocity regulation feature ensures to safely proceed on it by modulating the advancing velocity until completely stopping the motion if an obstacle is encountered on the path toward the target pose goal. Control actions are sent to the robot controller as smooth velocity commands at 1 KHz to be reactive. The capabilities of the proposed planner are showcased on a real robot in three dynamic experimental settings relevant to industry: (i) grasping objects moving on a conveyor belt, and two human–robot collaboration tasks that are (ii) grasping objects that can move during the robot's motion toward the target due to the interaction of a human operation, and (iii) human-to-robot handover. A comparison with state-of-the-art methods is also discussed. Demonstrations in the three different contexts are showcased in the supplementary material video.

1. Introduction

In recent years, with the advances of collaborative robots, the barriers between robots and humans have been removed with the aim of sharing the same environment, offering opportunities and challenges. To achieve an efficient and safe collaboration, robots should exhibit the ability to react to unstructured and uncertain environments, adapting their trajectory online depending on the changes in the surrounding environments. This is also in line with the concept of flexible manufacturing introduced by Industry 4.0 [1] and now 5.0 for industrial tasks or service robotics [2], in which rapid and customized differentiation of products requires human intervention and interaction. In order to pursue such a goal, robots should exhibit autonomous and intelligent functions to adapt their behaviors to the complexity and unpredictability introduced by the environment and target changes.

Nowadays, many industrial processes involve the collaboration of two operators [3]. For example, when two people perform a shared task like assembling together the same piece formed by several parts, one person may move the part that the other one was going to grasp during their motion. Typically, this situation is not a problem for humans since we are able to react to the changes on the fly. The same is not always true for robots, as they usually detect the target object, compute the grasping configuration, and then execute a trajectory for that location. Therefore, if the object moves during the execution of the trajectory, the robot might not be able to grasp it unless it updates the trajectory in a reactive way. This skill, other than for human–robot collaboration, is also essential for other tasks, like grasping objects that move on conveyor belts. Indeed, conveyor systems are widely used in industry, and robotic manipulators are being deployed extensively at conveyor belts for automation and faster operations [4]. However, when a robot needs to manipulate the object, the object reaches the photocell, stopping the conveyor with a consistent waste of time, or alternatively, the robot and the speed of the conveyor are synchronized in such a way that the robot performs seamlessly the same motion within the same time without being able to handle uncertainty or perturbation in the process. To increase throughput and reliability, robots should be able to grasp the objects while moving to avoid stopping the conveyor for every grasp. But, handling moving objects in uncontrolled environments is a more complex problem with respect to static conditions [5]: the perception module needs to continuously detect the object, track it, and generate feasible grasping configurations; the planning or low-level control module has to react to the changes in real time and on the fly, preserving a smooth and secure trajectory. Early works focused on tracking the poses of the objects when they move [6,7], and generating grasps based on the current object poses [8], or predicting future object poses based on

* Corresponding author.
E-mail addresses: emilio.maranci@santannapisa.it (E. Maranci), carlo.avizzano@santannapisa.it (C.A. Avizzano), salvatore.davella@santannapisa.it (S. D'Avella).

https://doi.org/10.1016/j.robot.2026.105401
Received 1 June 2025; Received in revised form 22 December 2025; Accepted 21 February 2026
Available online 24 February 2026

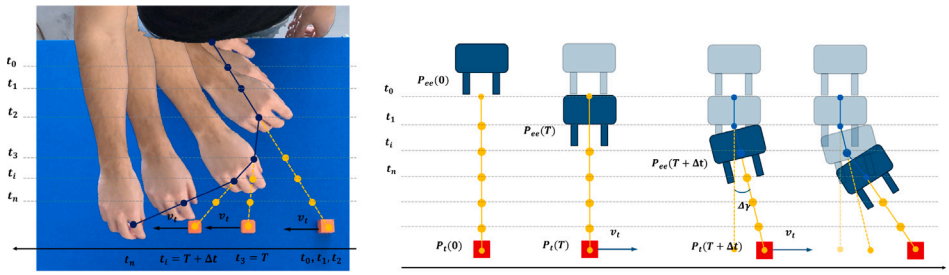


Fig. 1. Evolution of trajectory generation over time according to the human motion model of [12,13] (left) and performed by the robot under the proposed approach (right). The blue line is the path EE^- executed until that timestamp, while the orange line represents the remaining path EE^+ to be executed. Dots are the waypoints. At each instant of time, the orange line is straight toward the object, which is static at the beginning (until t_3 for the human motion), and then starts moving with constant velocity v_t, generating a heading variation $\Delta\gamma$.

the tracking results [9]. Recent approaches, instead, try to keep similar semantic features in subsequent grasps within a time horizon [10,11]. However, in the end, even if the high-level modules of the task are becoming increasingly complex, the actual grasping demonstrations involve object moving with very low speed primarily because existing approaches lack a sufficiently robust, reactive low-level controller to enable fluent tracking of target motion.

This work, indeed, focuses on the low-level reactive control aspect of the problem, and since people are very good at grasping objects even when they move, it finds its foundation in the human motion models studied in [12,13]. When a person has to grasp an object or, in general, has to reach a goal configuration, the person tries to reach the goal with a straight line in the joint space, and, if the object moves, the person tries to adapt the final configuration accordingly, and scales the velocity depending on the relative distance between the hand and the target.

The proposed approach is inspired by the aforementioned human motion model and implemented in the robot's operation space. Actually, the reactive grasp controller starts generating a straight line from the current position of the end-effector to the final pose configuration. Then, the line is shaped considering the new target position at each timestamp, and the velocity is scaled depending on the relative distance between the end-effector and the target, and eventually, other factors in the surroundings, like the presence of humans. Fig. 1 shows the similarities graphically. A trace controller, supported by the Lyapunov stability theory, guarantees that the end-effector will remain on the path and safely proceed on it whenever the trajectory is feasible. The control actions are sent to the robot controller as smooth velocity commands at 1 KHz to be reactive. To show the capabilities of the proposed planner, several experiments on a real robot have been performed in three different contexts resembling generic industrial and service robotics tasks: (i) grasping objects moving on a conveyor belt, and two human–robot collaboration tasks that are (ii) grasping objects that can move during the robot's motion toward the target due to the interaction of a human operation, and (iii) human-to-robot handover. It is worth noticing that the aforementioned tasks are open problems in the literature of robotic grasping and manipulation, and usually require a specialized solution to be managed, while the proposed method is able to solve them without applying any changes to its structure. Moreover, our method does not use any pre-computed set of paths or have any prior knowledge of the object's model, and runs the control loop faster than other solutions, being more reactive.

Summarizing, the contributions of this work are

- a novel reactive grasp controller inspired by a human motion model for industrial tasks that:
  - can grasp moving objects
  - is stable according to Lyapunov stability
  - provides a safety mechanism with collision awareness velocity regulation, decreasing the velocity for obstacles along the planned trajectory until completely stopping the robot
  - shows a versatile usability, being effective in multiple contexts
  - is robust to external disturbances and uncertainties in nominal conditions
  - is predictable since the generated trajectory is deterministic according to the used human motion model
  - is flexible as the velocity of the end-effector on the trajectory can be changed and adapted on the fly according to the external conditions
- a comparison with different state-of-the-art methods, showing the superior performance of our method.

The rest of the paper is organized as follows: Section 2 gives an overview of related works; Section 3 introduces the proposed approach describing each aspect of the planner; Section 4 presents the implementation details and the performed experiments in simulation and on the real robot; Section 5 discusses the achieved results and the comparison with two state-of-the-art methods; Section 6 concludes the essay and draws future directions.

2. Related works

Grasping static objects is a very explored field [14], while grasping moving objects, i.e., dynamic grasping, is a relatively recent and less researched field. The task can be decomposed into different submodules: object detection, pose estimation, grasp synthesis, object or grasp tracking, prediction in a future time horizon, all demanded by the vision system, and fast motion planning or reactive robot control to reach the target in the final configuration.

Initial works concentrated on realizing control algorithms to approach the moving objects with the end effector [8,15,16], limited to smoothing the trajectory of the end-effector when approaching the target, but leveraged many assumptions and considered very low target motion. In recent years, the literature has focused much more on vision systems, and the trend then moved to track the 6-DoF pose of the object while moving and generate the grasp based on the current object pose [7,17], or its future prediction based on the results of the tracking method [9,18]. However, such methods can still handle very low object velocities due to the computational demand of the 6-DoF tracking or uncertainty in the prediction. Recent methods, instead, concentrate on maintaining grasp consistency across subsequent frames, exploiting similar semantic features in an arbitrary time horizon [10,11,19]. However, even if the vision modules of the task are becoming increasingly complex, the actual grasping demonstrations are still limited to objects with very low speed because all such methods lack a robust and reactive low-level controller that makes the robot fluently behave according to the target motion. The proposed study aims to bridge this gap by introducing a reactive grasp controller inspired by humans, who are very good at grasping moving objects, and, in particular, implements the human motion model established in [12,13].

Some other approaches exploit reinforcement learning techniques to learn a policy for grasping moving objects [20–22], but they do not scale well to unknown scenarios.

On the aspect of the motion module, sampling-based motion planning methods are widely used in robotic manipulation [23] since they are able to handle high-dimensional systems better than graph-based methods. However, they are global methods that are usually not reactive since they optimize for a trajectory using a full explicit map of the environment by randomly sampling the robot's configuration space to identify a sequence of feasible nodes from the start to the goal and might need to plan from scratch when that map changes, resulting in slow and inefficient implementations that cannot easily adapt to environments explored online.

Alternative approaches use offline trajectory planning conducted before the robot's movement to compute an optimal initial guess and then apply trajectory planning online during execution. Recently, stochastic trajectory optimization (STO) methods, e.g., Via Point-STO [24] and chance-constrained Via Point-STO [25], have been utilized to optimize over a continuous trajectory space defined by via points. Although these approaches have been successfully implemented in different robotic systems, the online capability is still challenging.

Recently, some potential-based methods, like STORM [26] and RAMP [27], have been proposed for online collision avoidance in dynamic scenes, but are not developed to reach a moving target as required by dynamic grasping.

This study tests the proposed method and compares it against some state-of-the-art baselines in three different contexts resembling generic industrial and service robotics tasks: (i) grasping objects moving on a conveyor belt, and two human–robot collaboration tasks that are (ii) grasping objects that can move during the robot's motion toward the target due to the interaction of a human operation, and (iii) human-to-robot handover. The work presented in [9], which introduces a reachability and motion-aware dynamic grasping framework predicting the grasp quality based on the current motion of the target and replanning by utilizing a solution from the previous time step, is used in the conveyor setting (i) along with [10] that, instead, tracks grasps by matching the semantic feature in the local area around the grasps across subsequent frames. In the other two human–robot collaboration tasks (ii, iii), [28] is employed, which implements a generative grasping convolutional neural network in a closed-loop visual servoing approach to react to target location changes during the trajectory execution of the robot. Furthermore, the work in [10] is also used for a comparison in the tabletop setting (ii).

Different from the previous works, the proposed approach is able to change the trajectory of the end-effector on the fly depending on the target position and velocity changes, even if the object moves very fast, being more reactive. Several experiments have been performed on the conveyor setting with different velocities up to 8 m/min, a.k.a 13,33 cm/s. In addition, the method is based on a human motion model, making it interesting for future improvement, and the decoupling between the hand and the rest of the robot body with trace control lets the approach be versatile. Indeed, our approach can work in all the settings without applying any changes to its structure, while the others, due to their pipeline, can fit with at most two settings. Moreover, our method does not use any pre-computed set of paths or have any a priori knowledge of the model of the object. It is also robust to external disturbances and uncertainties in nominal conditions, and can even handle obstacles along the planned trajectory, modulating the approach velocity till completely stopping for safety reasons, showing adaptability to dynamic environments. Furthermore, it can be proved to be stable under Lyapunov stability.

3. Proposed approach

According to the human motion model presented in [12,13], a person who wants to reach a (moving) goal tries to execute a straight line in the joint space and reactively adjusts the path whenever the conditions change. Considering that we, as humans, are very good at grasping dynamic objects, this work proposes an implementation of such a strategy for robotic manipulators where the initial straight line in the cartesian space is shaped considering every new target position in real time.

The following first formally presents the problem, and then moves to the cartesian real-time planning and the reactive grasping control for executing and updating the path.

3.1. Problem formulation

Consider a robot $\mathcal{B}$ with N joints moving in a workspace $\mathcal{W} \subset \mathbb{R}^3$, $q = (\theta_1, \dots, \theta_N)$ is the N-dimensional vector representing its configuration in $\mathcal{C} \subset \mathbb{R}^N$, and $u \in \mathcal{U} \subset \mathbb{R}^N$ is the N-dimensional vector of the control inputs. The robot has a first-order dynamics, thus the control inputs are velocities $v = \dot{q}$, and its state is defined as $\mathbf{x} = q \in \mathcal{X}$. Given an initial state $\mathbf{x}(0) = q_{start}$, and the desired pose of the end-effector $T_{goal}(t) \in SE(3)$ corresponding to a configuration for grasping a target (moving) object $P_o(t) \in SE(3)$ at time t, the objective is to find a sequence of feasible control actions $u : [0, T] \to \mathcal{U}$ that drives the robot from q_{start} to q_{goal} while ensuring $FK(q(t))$ is collision free $\forall t \in [0, T]$ for eventual objects along the generated cartesian path.

3.2. Reactive end-effector line waypoint generation

The main idea is to compute and update in real time the most direct path between the end-effector and the target in the cartesian space and use it as a reference for the gripper. The approach fits well with most industrial tasks where precise placement of the terminal tool is required, while less attention can be put on body positioning.

Given a starting configuration q_{start}, we define $T_{start} = FK(q_{start}) \in SE(3)$ the starting pose of the end-effector and $ee_{start} \in \mathbb{R}^3$ the corresponding translational component, the algorithm generates a straight line path from ee_{start} to $ee_{goal}(t)$ coinciding with the target position $p_o(t)$ with $W_n(t)$ waypoints $EE(t)_{ee_{start}}^{ee_{goal}} = \{ee_{start} = w_1, w_2, \dots, w_{W_n-1}, ee_{goal} = w_{W_n}\}$ equally spaced with step $\delta \in \mathbb{R}$. In $EE(t)_{w_i}^{w_j}$, the pedex represents the starting point of the path, while the apex represents the final point. In the following, apex and pedex are omitted if they coincide with ee_{start} and ee_{goal}. It is worth noticing that the number of waypoints can vary over time as the path changes, but the time dependency is omitted to streamline notation. The path $EE(t = t_c)$ at a specific time $t = t_c$ can be represented as $EE(t = t_c) = \{EE(t)_{w_i}^{ee_{start}} = EE^- \cup EE(t)_{ee_{goal}}^{w_i} = EE^+\}$, i.e., the union of previously executed waypoints for $t < t_c$ and the set of remaining waypoints for $t \geq t_c$, being w_i the waypoint at current time t_c. The approach can be used either if the target is still or if it moves since the upcoming path EE^+ is always updated at runtime, generating eventually a non-linear closed-loop motion, as shown in Fig. 1. However, if the target moves with continuous acceleration and without abrupt velocity changes, the angular variation of the line connecting the end-effector to the goal remains continuous, resulting in a smooth closed-loop path. Such consideration usually holds in controlled industrial scenarios. Fig. 2 graphically explains the concept and shows that the angular rate of change $\Delta\gamma$ of the line can be roughly approximated as:

$$\Delta\gamma \approx asin(\frac{v_t^{\perp}}{R}), \tag{1}$$

where $v_t^{\perp}$ is the component of the target velocity vector v_t perpendicular to the line and R is the distance from the target. This type of control is well-known in mobile robotics [29], being the foundation of pure pursuit controllers.

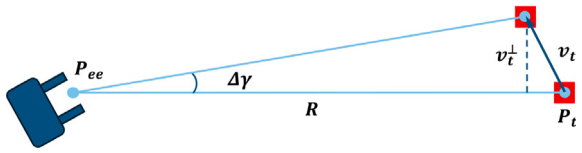


Fig. 2. Relation between heading variations $\Delta\gamma$ and target speed v_t.

3.3. Reactive grasp controller

To decouple the convergence on the path from the advanced speed, a trace control approach is used. However, this technique requires the closed-form equation of the curve $C(p) = 0$ that the end-effector should follow, where $p = [x, y, z]^T \in \mathbb{R}^3$ represents the generic point in Cartesian space. Since $EE(t)$ is a time-varying path composed of a non-constant number of waypoints, a least-squares polynomial approximation is used locally to the current position of the robot's end effector p_{ee} to generate a local approximating polynomial $\bar{C}(p)$. Being $w_{ee} = \arg\min_{w \in EE(t)} \|w - p_{ee}\|$, the waypoint closest to the current position of the robot's end effector, $\overline{EE} = \{w_{ee \pm l}\}$, $l \in [0, L]$ represents the local set of waypoint adjacent to w_{ee}. We formulate the analytic expression of the interpolating polynomial in three-dimensional space as the intersection of two surfaces. The generic univariate polynomial $F(\zeta)$, $\zeta = \{x, y, z\}$ of degree D representing a surface in the 3D space can be expressed as a function of ζ, such that $F(\zeta) = \Gamma_\zeta \bar{\zeta}$, where $\Gamma_\zeta = [\gamma_0, \dots, \gamma_D]$ is the vector of coefficients and $\bar{\zeta} = [1, \zeta, \zeta^2, \dots, \zeta^D]^T$ the basis vector.

By expressing y and z as a function of x, the local polynomial approximation $\overline{C}(p)$ is obtained by solving the intersection of two surfaces described by the following system of equations:

$$\overline{C}(p)_{|x} : \begin{cases} y = \Gamma_y \bar{x} \\ z = \Gamma_z \bar{x} \end{cases} \tag{2}$$

where the coefficients Γ_y and Γ_z are obtained by minimizing the distance between each surface and the considered waypoints in $\overline{EE}$ as:

$$\min_{\Gamma_y}(\tilde{Y} - \tilde{X}\,\Gamma_y^T)^2, \quad \min_{\Gamma_z}(\tilde{Z} - \tilde{X}\,\Gamma_z^T)^2, \tag{3}$$

in which $\tilde{Y}$ and $\tilde{Z}$ are the y, and z components of each waypoint in $\overline{EE}$, respectively, while $\tilde{X}$ is the polynomial matrix of power:

$$\tilde{X} = [\overline{x_{ee-L}}, \dots, \overline{x_{ee+L}}]^T \in \mathbb{R}^{(2L+1)\times(D+1)} \tag{4}$$

Therefore, the closed-form equation of the curve $\bar{C}(p) = 0$ can be obtained from Eq. (2) as:

$$\bar{C}(p)_{|x} = \begin{bmatrix} y - \Gamma_y \bar{x} \\ z - \Gamma_z \bar{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5}$$

The choice of the polynomial approximation is not unique since the same procedure can be repeated using the other two basis variables y and z, and obtaining the set $\{\overline{C}(p)_{|x}, \overline{C}(p)_{|y}, \overline{C}(p)_{|z}\}$. Therefore, the selected polynomial approximation is chosen according to the variable ζ that presents the polynomial matrix of power $\tilde{\zeta}$ (in Eq. (4)) with the lower conditioning number $\|\tilde{\zeta}\| \cdot \|\tilde{\zeta}^{-1}\|$. In the following, we assume to choose $\zeta = x$ without losing generality.

The formulation of trace control is derived from Lyapunov's stability theory by proposing the following Lyapunov's function candidate,

$$V(p) = \frac{1}{2}\bar{C}(p)_{|x}^T \bar{C}(p)_{|x}, \tag{6}$$

which is positive-definite everywhere except on the trace $\bar{C}(p)_{|x}$ By deriving Eq. (6) with respect to time, we get:

$$\dot{V} = \bar{C}(p)_{|x}^T \bar{C}_p(p)_{|x} \dot{p} \tag{7}$$

where $\bar{C}_p(p)_{|x} = \partial \bar{C}(p)_{|x} / \partial p$ and $\dot{p} = J(q)\dot{q}$, assuming a pure kinematic control which means directly controlling the robotic arm joints velocities $\dot{q}$ through the Jacobian. Since variables are decoupled, $\bar{C}_p(p)_{|x}$ can be easily computed in closed form as:

$$\frac{\partial \bar{C}_{|x}}{\partial x} = -\begin{bmatrix} \Gamma_y \\ \Gamma_z \end{bmatrix} \frac{\partial \bar{x}}{\partial x} = -\begin{bmatrix} \Gamma_y \\ \Gamma_z \end{bmatrix} [0, 1, x, \dots, D\,x^{D-1}]^T, \tag{8}$$

leading to the complete form of the matrix $\bar{C}_p(p)_{|x}$ as:

$$\bar{C}_p(p)_{|x} = \begin{bmatrix} -\Gamma_y \dfrac{\partial \bar{x}}{\partial x} & 1 & 0 \\ -\Gamma_z \dfrac{\partial \bar{x}}{\partial x} & 0 & 1 \end{bmatrix} \tag{9}$$

Choosing an orthonormal basis $S(p)$ of the null space of $\bar{C}_p(p)_{|x}$, i.e., $\bar{C}_p(p)_{|x} S(p) = 0, \forall p$, and a right-pseudoinverse of the robot's Jacobian $J^\dagger(q)$, we can write following stabilizing control action in closed-form:

$$\dot{q} = -J^\dagger(q)\bar{C}_p^T(p)_{|x}\bar{C}(p)_{|x}\xi + J^\dagger(q)S(p)\alpha \tag{10}$$

where ξ and α are two scalars for tuning the convergence rate on the curve $\bar{C}(p)_{|x}$ and the advancing speed, respectively, and a closed-form expression of a basis $S(p) \in \mathbb{R}^3$ of its null space is:

$$S(p) = \begin{bmatrix} 1 \\ \Gamma_y \dfrac{\partial \bar{x}}{\partial x} \\ \Gamma_z \dfrac{\partial \bar{x}}{\partial x} \end{bmatrix} \tag{11}$$

Substituting Eq. (10) in Eq. (7), we get:

$$\dot{V} = -\bar{C}(p)_{|x}^T \bar{C}_p(p)_{|x} \bar{C}_p^T(p)_{|x} \bar{C}(p)_{|x} \xi \tag{12}$$

which is semi-negative defined when $\xi > 0$, guaranteeing the stability of the curve $\bar{C}(p)_{|x}$ and the asymptotic attractiveness on its neighborhood, according to Lyapunov's stability theorem.

Therefore, a closed-form of Eq. (10) can be written having only the current end-effector position and the interpolating polynomial. Theoretically, some points could exist outside of the curve $\bar{C}(p)_{|x}$ that verify $\dot{V} = 0$ but are not attracted toward the curve. These points are the ones that belong to the kernel of $\bar{C}_p(p)_{|x}$, where the derivative of the curve loses rank. However, our closed-form of Eq. (10) ensures the stability of the approach and the attractiveness of the polynomial curve $\bar{C}(p)_{|x}$ for each point of the space since the matrix in Eq. (9) never loses rank.

3.4. Orientation control

Being $R_{ee}(q(t)) \in SO(3)$ and $R_{goal} \in SO(3)$, the rotation matrices representing the current and desired orientation of the end-effector, respectively, the orientation error $e_R(t)$ can be expressed using the Lie algebraic formulation as:

$$e_R(t) = \log(R_{goal}^T R_{ee}(q(t)))^\vee \in \mathbb{R}^3, \tag{13}$$

where $\log(\cdot)$ is the matrix logarithm from $SO(3)$ to $\mathfrak{so}(3)$, and $(\cdot)^\vee$ represents the v-operator mapping skew-symmetric matrices to $\mathbb{R}^3$. To reject the orientation error we use the following control action:

$$\dot{q}_R = K_R J_R^\dagger(q(t)) e_R(t) \tag{14}$$

where K_R is a scalar weight to tune the convergence rate and $J_R^\dagger(q(t))$ is the right-pseudoinverse of the orientation Jacobian of the robot.

3.5. Control actions

The two control actions introduced in Sections 3.3 and 3.4 are combined into a single control law to fulfill both tasks simultaneously based on Lyapunov's stability theory. The proposed control requires 6 DOFs (3 for orientation and 3 for position) and thus can be applied to any robots having at least 6 DOFs. In particular, the convergence on

the path and the orientation errors are vertically stacked into a single error vector, as in the following equation:

$$e = \begin{bmatrix} \bar{C}(p) \\ e_R \end{bmatrix} \tag{15}$$

The proposed Lyapunov's candidate function is:

$$V = \frac{1}{2} e^T e = \frac{1}{2} \bar{C}(p)^T \bar{C}(p) + \frac{1}{2} e_R^T e_R \tag{16}$$

where we can find the same Lyapunov candidate presented in Eq. (6) for trace control, and an additional term referred to the orientation. By deriving Eq. (16) w.r.t. time, we get:

$$\dot{V} = e^T \dot{e} = e^T \begin{bmatrix} \bar{C}_p(p) J(q) \\ -J_R(q) \end{bmatrix} \dot{q} \tag{17}$$

and rearranging the matrix containing position and orientation Jacobians, Eq. (17) becomes:

$$\dot{V} = e^T \begin{bmatrix} C_p(p) & O^{3x3} \\ O^{3x3} & -I^{3x3} \end{bmatrix} J(q)\, \dot{q} = e^T A\, J(q)\, \dot{q} \tag{18}$$

where J is the complete Jacobian of the manipulator. Introducing the matrix:

$$B = \begin{bmatrix} S(p) \\ 0^{3x1} \end{bmatrix} \in \mathbb{R}^6, \tag{19}$$

the stabilizing control law for the proposed Lyapunov candidate is expressed as:

$$\dot{q} = -J^{\dagger}(q) A^T K e + J^{\dagger}(q)\, B\, \alpha \tag{20}$$

where $K > 0$ is a diagonal matrix of weights for individually controlling the two error dynamics and α is a scalar term that controls the advancing speed on the trace. Neglecting the weight matrices to simplify the notation and substituting the proposed control law (Eq. (20)) in Eq. (18), we get:

$$\begin{aligned} \dot{V} &= -e^T A A^T e + e^T A B \\ &= e^T \begin{bmatrix} -\bar{C}_p(p) \bar{C}_p^T(p) & 0 \\ 0 & -I \end{bmatrix} e + e^T \begin{bmatrix} \bar{C}_p(p) S(p) \\ 0 \end{bmatrix} \end{aligned} \tag{21}$$

The second term of Eq. (21) is zero because $S(p)$ is an orthonormal basis of the null space of $\bar{C}_p(p)$ by construction, as explained in Section 3.3. Expanding the first term we get:

$$\dot{V} = -\bar{C}(p)^T \bar{C}_p(p) \bar{C}_p^T(p) \bar{C}(p) - e_R^T e_R \leq 0, \tag{22}$$

which is semi-negative defined. Therefore, the control law in Eq. (20) satisfies Lyapunov's theory for the candidate presented in Eq. (16), ensuring the stability of both tasks. However, the stability of the advancement task is not guaranteed. Nonetheless, it represents a feed-forward term that does not perturb or interfere with the asymptotic stability of positioning and orienting tasks.

3.6. Micro-interpolation

The control actions just described are generated at 60 Hz and can lead to discontinuities in the motion of the robot, potentially causing high jerk peaks. To solve the problem, a micro-interpolation step applies a sigmoid function between individual control steps, sending final smooth control velocity actions $\dot{q}$ to the robot at 1 kHz. The sigmoid function used is continuous at the switching points until the second derivative (jerk), ensuring the smooth transition.

Hereafter, a single joint is considered, but the same applies to all the others. The microinterpolation module receives the reference velocity v_r every T_p seconds at 60 Hz. Assuming to receive v_{curr} at t_{curr} and being v_{prev} the target velocity at the previous period t_{prev}, the microinterpolation generates the new velocity v_{act} weighted with a sigmoid profile σ to pass from v_{prev} to v_{curr} over the time elapsed

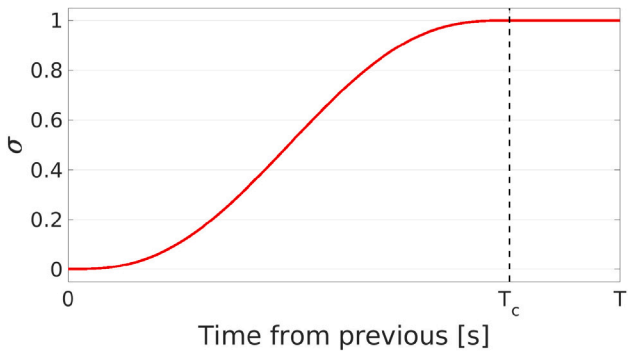


Fig. 3. Normalized sigmoid function over the nominal period T_p considering $v_{prev} = 0$ and $v_{curr} = 1$. The dashed line represents the rising time T_c to reach the new reference (80% of T_p).

from the last received update $t_d = (t - t_{prev})/T_c$, where T_c is the desired rising time that corresponds to an arbitrary percentage of T_p, as:

$$r_{act}(t) = (1 - \sigma(t))\, r_{prev} + \sigma(t)\, r_{curr} \tag{23}$$

The sigmoid function is continuous at the switching points until the second derivative (jerk), thus ensuring a smooth transition between extreme values. It is graphically depicted in Fig. 3 and analytically presented below:

$$\sigma(t_d) = \begin{cases} 0 & t_d \leq 0 \\ t_d^3\, (6t_d^2 - 15t_d + 10) & 0 < t_d \leq 1 \\ 1 & t_d > 1 \end{cases} \tag{24}$$

3.7. Collision aware velocity regulation

The control action proposed in Eq. (20) allows adjusting the advancing speed on the trace at runtime by changing the scalar value α. This is useful for two reasons: (i) increasing the speed for grasping the moving objects to avoid the tail-chase effect, and (ii) slowing down and eventually stopping the velocity to avoid potential collisions with dynamic obstacles on the path to guarantee a safe and controlled robot's behavior. To this end, an octomap-based obstacle detection module has been embedded in the framework to monitor changes in the scene without prior knowledge of the environment. Since the planner generates a deterministic linear path in real time, the proposed approach can look at the incoming path EE^+ to detect obstructed waypoints that are closer than a threshold distance d_{min} to any obstacle, which represents a potential collision for the end-effector. Therefore, the value of α is adjusted at runtime as:

$$\alpha = 1 - \frac{d_{max} - d(p_{ee}, w_o)}{d_{max} - d_{min}} \tag{25}$$

where $d(p_{ee}, w_o)$ is the distance from the current position of the end effector p_{ee} to the first blocked waypoint w_o, and d_{max} is the maximum value for which this slowing effect is considered. As a consequence, the robot linearly reduces the velocity when approaching an obstacle until reaching a complete stop when the distance reaches d_{min}. As soon as a collision-free path to the target becomes available the robot starts moving again.

4. Experimental validation

The method has been tested on a real robot in three dynamic experimental settings relevant to many industrial and manufacturing tasks:

1. grasping moving objects on a conveyor belt (one axis)
2. grasping objects that change location on a plane (two axes)

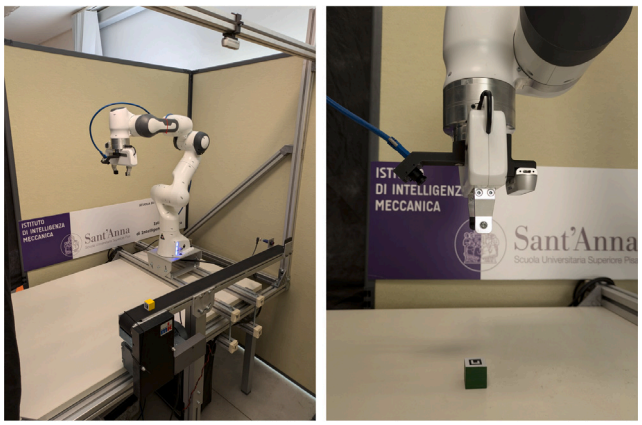


Fig. 4. Camera setup. Left: task 1; Right: tasks 2 and 3. The Realsense camera is used for long-medium range distances, while the FireFly S in-hand camera is used in the short range of the final phase grasp when the other camera cannot detect the target due to occlusions.

3. human-to-robot handover where the target object can move in space (three axes).

The experiments have an increasing level of difficulty in terms of the degrees of freedom of the target object's motion to be handled. The latter two cases involve human–robot collaboration tasks. It is worth noticing that all three settings represent open problems in the robotic grasping and manipulation research and require specialized solutions. Demonstrations in the three different contexts are showcased in the supplementary material video.

The method has been executed in a docker container with ROS Noetic on a PC with Intel Core i9 having 32 GB of RAM and Nvidia GeForce 4070 with 8 GB vRAM running Ubuntu 22.04 LTS. The experiments have been conducted on the Franka Emika Panda robot, employing a double camera setup for all the tasks: task 2 and 3 present the same setup and grasping strategy, while task 1 is a little bit different. In particular, in task 1, an external Realsense 435 camera has been used with a top-down view looking at the conveyor, in combination with an eye-in-hand RGB Firefly S camera having a 1/2,5" 3.6 mm 5MP optics. For tasks 2 and 3, the Realsense 435 camera has been utilized in combination with the Firefly S camera as a double eye-in-hand configuration on both sides of the parallel-jaw gripper (see Fig. 4). It is worth noticing that the choices of the Firefly S camera, its optics, and the rapidly prototyped printed support have been designed on purpose to guarantee that the target object remains visible until the last grasp phase, as depicted in Fig. 5. The following formulas detail the design choices:

$$h\_FoV = 2\arctan\frac{w}{2f} = 2\arctan\frac{4.968}{3.6} = 69.21° \tag{26}$$

$$v\_FoV = 2\arctan\frac{h}{2f} = 2\arctan\frac{3.726}{3.6} = 54.72° \tag{27}$$

$$L_x = 2h \cdot \tan\frac{h\_FoV}{2} = 169.6 \cdot \tan 34.6° = 11.7 \text{ cm} \tag{28}$$

$$L_y = \frac{L_x}{\rho} = 8.78 \text{ cm} \tag{29}$$

where w and h are the width and height of the camera sensor, f is the focal length, ρ is the ratio between the horizontal and vertical resolution of the image sensor, h_FoV and v_FoV are the horizontal and vertical field of view, and L_x and L_y are the horizontal and vertical space covered by the image at distance h from the sensor. The cameras have been calibrated to know the intrinsic parameters, and for the external camera in task 1 a robot–camera calibration has been performed for the extrinsic parameters.

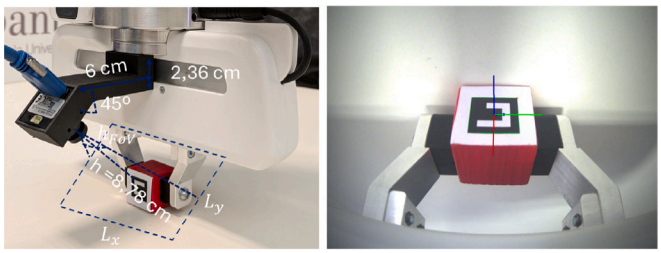


Fig. 5. FireFly camera setup for a close-up view guaranteeing object visibility in the final grasping phase.

Since the presented method does not focus on the perception part, color detection and Aruco markers are used to detect the grasping pose. Cubes from the YCB dataset [30] of size 2,5 cm are employed in tasks 1 and 2, while Mustard, Foam Brick, Plastic Banana, Hammer, Spam Meat Can, Screwdriver, and Plastic Duckie (still from the YCB dataset) are involved in the human-to-robot handover. Color detection is exploited from the external Realsense camera in task 1, while Aruco marker detection is always employed by the Firefly S and by the Realsense in the other tasks. For tasks 2 and 3, the orientation followed during the execution of the entire trajectory is not the one required to grasp the object, rather it is adapted on the fly to maximize the chances of keeping the target inside the vision cone of the Realsense camera, which is used for long/medium distance detection. The strategy aims to orient the gripper along the line connecting the Tool Center Point (TCP) to the goal position, considering that the Realsense camera is mounted parallel to the gripper. To this aim, a virtual orientation that matches the direction of the incoming straight path EE^+ is generated at runtime and sent to the controller. A weighted SLERP interpolation is then applied to obtain the actual gripper orientation by merging the virtual pose and the one detected through the camera for grasping. The weight used is proportional to the distance to the target, thus allowing a smooth switching to the orientation necessary for grasping when approaching the target, i.e., when a clear image is available from the other close-up camera.

The general strategy for the positioning of the gripper is the same for all three tasks: it updates the remaining path EE^+ in real-time based on the latest available target position detected through the cameras as explained in 3.2. Additionally, a synchronization mechanism has been developed to handle conflicts between the detection modules when both cameras detect the target simultaneously: in these cases, the RealSense camera takes the lead over the other. As soon as the Realsense is no longer able to detect the target, the system switches to the close-up camera, as depicted graphically in Fig. 5, guaranteeing a precise finger positioning. A different advancing speed on trace has been selected for each experiment: for tasks 1 and 2, the chosen speed is constant, while, in task 3, the actual speed of the gripper is changed at runtime in a way that it is inversely proportional to the distance between the gripper and the interaction area with the operator, showcasing safe behavior when approaching the handover. Finally, the three tasks have been repeated by introducing dynamic obstacles along the path to stress the robustness of the planner to perturbation.

The proposed method has been compared with three state-of-the-art approaches for the corresponding tasks: [9] for task 1, [28] for task 2 and task 3, and [10] for task 1 and task 2.

4.1. Grasping moving objects on a conveyor belt

Two conveyor belts with the following characteristics have been used:

(a) the conveyor is large 0,08 m and long 1,0 m, with an effective length between the start positioning of the objects and the photocell, which stops the movement, of 0,9 m. The nominal speed is 5 m/min.

(a) (b) (c)

Fig. 6. Task1: grasping moving objects on the conveyor belt.

(b) the conveyor is large 0,5 m and long 1,60 m. The speed can be changed on the fly from 0 to 16 m/min.

The setup with *conveyor a* is depicted in Fig. 6-a. The top-down Realsense camera runs at 60 Hz with a resolution of 640 × 480, detecting the top of the cube via color segmentation after cropping the region of interest (ROI) that contains only the conveyor belt. The robot has a home configuration, which is 90 degrees perpendicular to the longitudinal direction of the conveyor, and starts moving once the Realsense camera detects the cube for the first time. Then, the end-effector follows the cube (Fig. 6-b), and once it is in the final grasping phase, the gripper occludes the view of the external camera, but the eye-in-hand running at 60 Hz can detect the Aruco marker attached to the side of the cube (Fig. 6-c). The same applies to *conveyor b*. Nominal conditions are as follows:

- constant speed at 5 m/min, which is not known by the robot, thus being estimated from the perception module
- initial position of the target is at zero coordinates of the conveyor frame

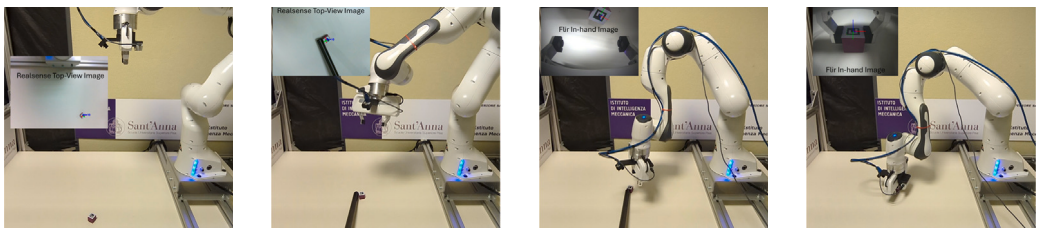
Perturbations and uncertainties have been applied by (i) pausing and restoring *conveyor a* multiple times during the target movement, (ii) using different initial positions along the direction of motion of *conveyor a*, (iii) using *conveyor b* with different constant and dynamic velocities, (iv) using different initial coordinates and (v) orientations on *conveyor b*. The task ends either successfully if the robot grasps the cube before it reaches the photocell or fails otherwise.

4.2. Grasping moving objects in human–robot collaboration

The robot should grasp a cube placed on the table in a working area of 0,6 × 0,4 m. The cube presents an Aruco marker on the top surface. The hardware configuration is the double eye-in-camera, where the Realsense running at 60 Hz is in charge of detecting the cube from the long-medium range, and the Firefly S, running at 60 Hz, is used for a close-up view in the last grasping phase. The experiment starts when a cube is placed on the working table and the vision system, pointing toward the table, detects it (Fig. 7-a). While the robot moves to grasp the target, the cube is moved two or three times on the plane with rapid touches through a tool (Fig. 7-b) that lets it move of few centimeters with respect to its previous position (Fig. 7-c). The task ends if the robot can grasp the cube (Fig. 7-d) or if it is not able anymore to track the target.

4.3. Human-to-robot handover

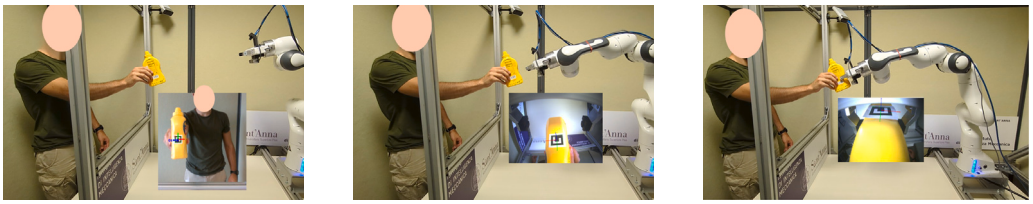
Seven objects from the YCB dataset (i.e., Mustard, Foam Brick, Plastic Banana, Hammer, Spam Meat Can, Screwdriver, and Plastic Duckie, shown in Fig. 8), are involved in this experiment. An Aruco marker is attached to each object and represents the grasping point for the end-effector, while the grasping depth is adjusted accordingly to the specific object. In a trial, an object is held in the hand of an operator to perform the handover with the robot. It is worth noticing that the operator had not performed any adjustment aimed at increasing the grasping possibilities: if a possibility of cheating was detected the trial was repeated. There is a safety distance between the operator and the robot: the interaction can happen just in a designed interaction area, where only the hand of the operator is present. The hardware configuration is the same as the previous task. The experiment starts when one of the objects enters the interaction area and the vision system, pointing toward the operator, detects it (Fig. 9-a). As the robot moves to perform the handover, the operator can adjust the pose of the target until the final grasping phase since it may be difficult for the operator to maintain the same configuration for the whole robot trajectory execution (Fig. 9-b). The speed of the robot decreases for safety reasons while approaching the area covered by the operator, who is detected through the Google MediaPipe 3D body pose estimation. Fig. 10 shows the 2D skeleton detection and the 3D area represented as a red cylinder. The task ends if the robot can grasp the object (Fig. 9-c) or if it is not able anymore to track the target. The experiments were repeated in various relative arrangements between the initial configuration of the robot and the target pose, while ensuring that the object was visible to the in-hand camera at the start of each trial. This constraint is illustrated in Fig. 11, which highlights the limited field of view of the wrist-mounted RealSense camera.



(a) (b) (c) (d)

Fig. 7. Task2: grasping moving objects in a (potential) human–robot collaboration scenario.

Fig. 8. Objects used in task 3.



(a) (b) (c)

Fig. 9. Task3: human-to-robot handover.

4.4. Collision aware velocity regulation

The previous tasks have also been repeated by introducing a dynamic obstacle in the scene, interfering with the trajectory of the end-effector, to further test the robustness of the method to external perturbations and uncertainties and prove its safety mechanisms. It is worth noticing that the other three baselines cannot handle dynamic environments.

5. Discussion

Each task has been repeated several times to be statistically significant, and Table 1 summarizes the statistics of the comparison in the three tasks under different experimental settings.

Experiments for Task 1 were conducted under three conditions: (i) 15 trials under nominal conditions on *conveyor a*; (ii) 30 trials introducing perturbations and uncertainties, including pausing and resuming the motion (10 trials), varying the initial position within 30 cm of the nominal setup on *conveyor a* (10 trials), and changing the object orientation to −45°, 30°, 60°, 90°, and 135° relative to the nominal pose on *conveyor b* (10 trials); and (iii) 15 trials on *conveyor b* with non-nominal velocities and varying initial poses.

It is worth noticing that in our approach, no velocity model has been used. However, it is important to select an advancing speed for the robot greater than the conveyor belt's nominal speed to avoid tail-chase scenarios characterizing pure tracking techniques that would have made the experiments unfair. The errors for our method (one with nominal conditions on *conveyor a*, two with perturbations and uncertainties on *conveyor a*, and two on *conveyor b*) are due to four main reasons: (i) in the final part of the conveyor near the photocell, the robot is close to a singularity, which reduces its precision, both in terms of position and orientation (ii) there is no closed loop feedback during the closure of the gripper, thus, once the closure is triggered, even if the camera detects a new change of the target in that fraction of second, there is no way to recover, (iii) if the velocity of the conveyor is set higher than the maximum velocity set for the robot it is unable to reach the target, (iv) with 135° orientation offset, the target moves in a direction opposite to the line of sight of the close-range camera, leading to a temporary loss of visual tracking and subsequent misdetection. For the latter case, it is worth noticing that when the orientation offset was −45°, the grasp was executed successfully, leveraging the symmetry of the parallel-jaw gripper. This outcome indicates that the observed limitation is primarily related to the hardware design rather than to the proposed method itself. Different velocities have been used up to 8 m/min successfully. The two baseline methods [9] and [10], instead, were found not to be able to perform dynamic grasping at a speed higher than 2 m/min. The main problem in [9] is the continuous replanning caused by unnecessary changes in the grasping configuration. The only chance [9] had to grasp the moving target was when the conveyor stopped and the object remained in its steady position on *conveyor a* or with very low speed (i.e., less than 2 m/min) on *conveyor b*. For [10], the failures were due to the fact that the low-level controller was unable to handle higher speeds, and, in addition, the tracking lost the grasp on the target object at high speed.

The experiment for task 2 was repeated 30 times, perturbing the target multiple times during the robot's motion. The motivation for the (single) error of our method is related to the second reason discussed in Task 1: the operator moved the object abruptly after the trigger to close the gripper was issued. The method presented in [28], instead,

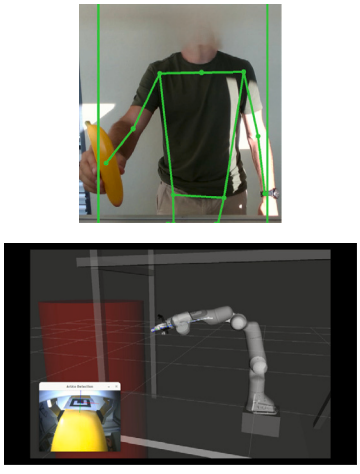


Fig. 10. MediaPipe 2D and 3D human detection.

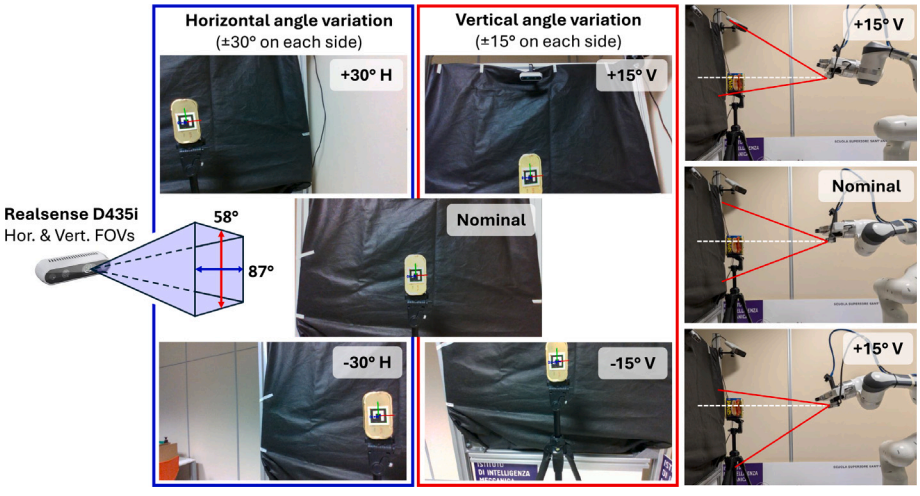


Fig. 11. Field-of-view analysis of the wrist-mounted camera in the human-to-robot handover setup. Due to the limited field of view of the RealSense camera, reliable target detection is ensured only within an angular variation of approximately ±15° in the vertical direction and ±30° in the horizontal direction around the nominal pose.

Table 1
Experimental validation and comparison results.

Methods	Task 1			Task 2	Task 3
	Nominal conditions	Perturbations and uncertainties	Different static and dynamic velocities	succ | avg time | effort	succ | avg time | effort
	succ | avg time | effort	succ | avg time | effort	succ | avg time | effort		
ours	93,33 - 4,65 - 1,57	90,00 - 5,31 - 2,01	93,33 - 3,42 - 2,25	93,33 - 8,35 - 1,30	94,28 - 12,24 - 0,13
[9]	0 - -	13,33 - 18,29 - 3,37	6,33 - 19,36 - 4,49	–	–
[28]	–	–	–	56,66 - 13,4 - 1,32	20,00 - 15,21 - 0,20
[10]	0 - -	26,67 - 7,12 - 2,32	13,33 - 8,12 - 3,80	60,00 - 11,2 - 1,42	–

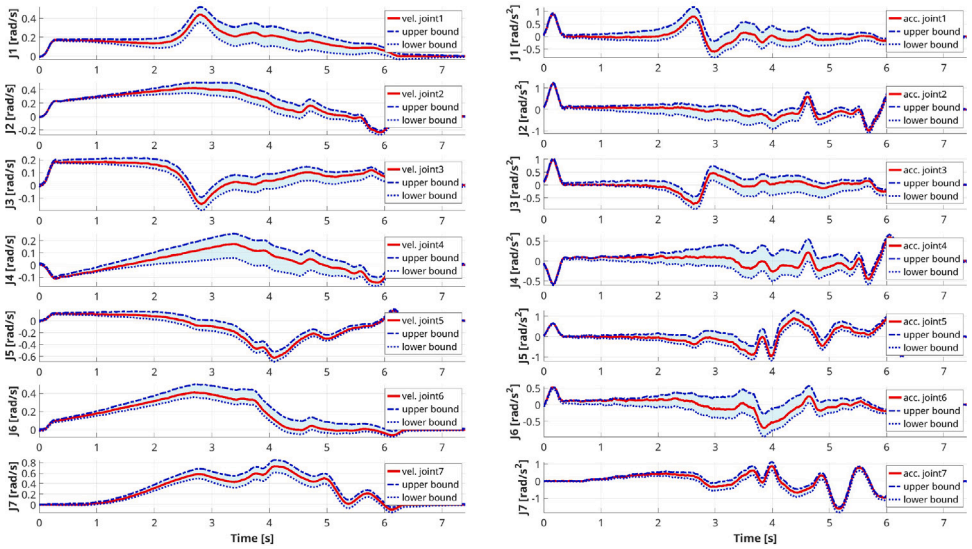


Fig. 12. Velocity (left) and acceleration (right) profiles of the seven joints of the robot captured during a set of 10 executions while performing task 1. The graphs depict the values obtained during a single execution of the task (red) and the envelope (light blue) obtained by overlapping the readings of the other executions.

failed multiple times due to two main reasons: (i) the limited camera's field of view of the depth compared to our multi-camera setup, (ii) basic Cartesian control that could end up in singularities, contrary to our method that guarantees control stability. The other baseline [10] performed slightly better than [28]. Its failures were related to tracking errors of the target grasp during the object motion, especially in the last 15 cm along the z-axis.

The experiment for task 3 was repeated five times per object, registering two errors. The first was caused by an inaccurate estimate of the target pose immediately before gripper closure. The second resulted from a temporary loss of visual contact with the object when testing extreme camera field-of-view angles; this limitation is primarily attributable to the hardware design rather than to the proposed method itself. The same considerations about the failures of [28] discussed for task 2 can be applied here.

Concerning the collision avoidance capabilities, the experiments have been repeated 10 times per task. Only one failure has been registered with the conveyor experiments since the obstacle persisted too long on the path, thus not giving the robot the possibility to reach the target.

5.1. Method analysis

Our method ensures a smooth path for the end-effector from the starting configuration to the target goal, and the micro-interpolation guarantees sending smooth control velocity actions at 1 kHz. Fig. 12 depicts the velocity and acceleration of the seven joints of a run in task 1, which is the one with the fastest dynamics, along with their envelope across 10 trials using all the same experimental settings with a conveyor speed of 5 m/min. The plots show that the method is consistent since the spread of the graphics is limited. Additionally, Fig. 13 shows the effect of the microinterpolation step, without which it is evident that the profiles are more jerky. The quantities are obtained with the same procedure that was previously used.

To further assess the benefits of the proposed approach and showcase its applicability in generic scenarios of dynamic grasping, we tested the method against two widely used closed-loop techniques to control a robot, i.e., visual servoing (VS) and pseudo inverse Jacobian algorithms (PIJ), in a controlled physical-based simulated environment (ROS Gazebo). The proposed scenarios have been designed to resemble those used for testing in real-world environments, involving a small target that moves with variable speeds on fixed trajectories, which are graphically shown in Fig. 14. To ensure the fairness of the comparison, all three methods can access the target's position directly, thus underlying perfect sensing. Additionally, since VS and PIJ cannot reject the tracking error from a moving target, they have been embedded with the same PI (proportional–integral) controller used in our approach to avoid the tail-chase effect when traveling at a speed closer to the target one. Finally, all three methods have been properly tuned to guarantee the same maximum controlled velocity of the end effector. It is worth noting that the closing of the gripper has not been considered during these experiments, as the execution is considered successful if the robot can reach the target pose with the end effector with a maximum error norm of 2.5 cm (the object's size) and 0.3 radians. Multiple trials have been repeated, changing the target speed in the range [2, 5, 8, and 11 m/min] on linear trajectories (resembling task 1) and with the nominal speed of 5 m/min on curved paths. Table 2 summarizes the achieved results in terms of success rate, average completion time, cartesian path length, and overall effort, intended as the amount of path traveled in configuration space before reaching the target. The results reported in Table 2 are obtained by averaging the outcomes of all the executions, considering the actual value for successful executions,

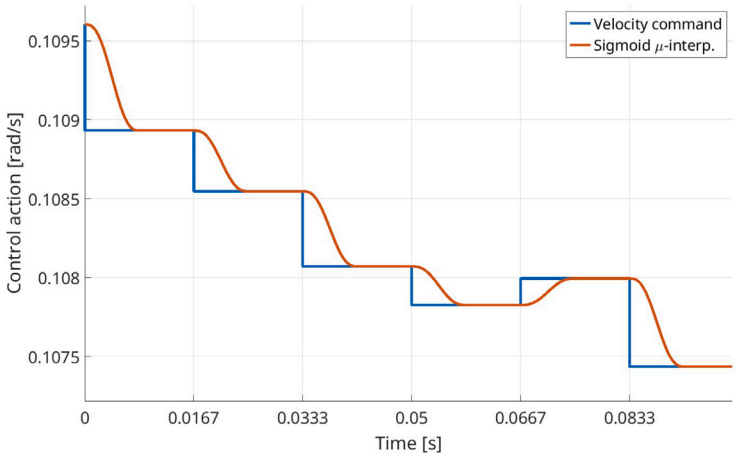


Fig. 13. Sigmoid microinterpolation function applied to a portion of a velocity command. The velocity command (60 Hz) is interpolated, and a smooth velocity reference for the robot is generated at 1 kHz.

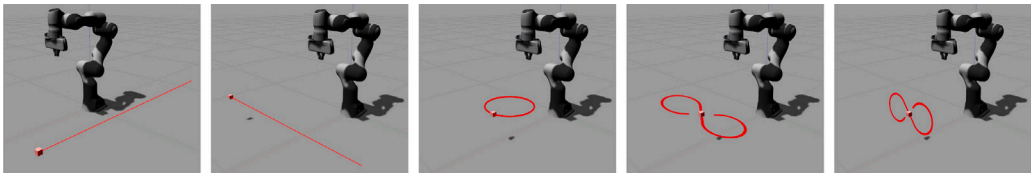


Fig. 14. Scenarios used to compare the proposed reactive grasp controller against visual servoing and pseudo inverse Jacobian algorithms in the ROS Gazebo physics-based simulation. Multiple trials have been repeated, changing the target speed in the range [2, 5, 8, and 11 m/min] on linear trajectories (resembling task 1) and with the nominal speed of 5 m/min on curved paths.

and the worst value among all methods for failures. The proposed approach obtained better performance in all the tasks and metrics. In particular, other than achieving the highest success rate, our method could perform the tasks within the shortest time and cover the shortest distance among the three, both in task space (which was expected as we follow the shortest path to the goal as a design choice), and in configuration space, thus achieving the smallest effort. The failures observed in the competing methods mainly arise from their limited ability to rapidly track and follow the moving object. Visual servoing approaches generate cartesian velocities for the end effector that are typically capped and require careful tuning of the maximum speed along each axis. Consequently, their performance is highly sensitive to the target pose and tends to be overly conservative when handling fast motions. Pseudoinverse Jacobian control, on the other hand, produces joint commands that minimize the norm of the required joint velocities to reduce the error. However, since these actions are applied in open loop, the resulting motion does not necessarily follow a straight path in Cartesian space, which can lead to inefficient trajectories. By contrast, our planner explicitly enforces motion along the straight line toward the goal through the trace controller, guaranteeing predictable and consistent trajectories. This approach directly contributed to improved overall performance, as demonstrated by our experimental results.

Current grasp synthesis methods can hardly generate or track grasps at high frequency. Therefore, we analyzed the effect of different response times of the vision module mimicking diverse grasp synthesis speeds (30 Hz, 15 Hz, 10 Hz, and 5 Hz). Even in this case, task 1 has been selected because the target moves faster than the other two settings, making it more challenging. 10 trials per frequency have been performed for different conveyor speeds (3m/min, 5m/min, 8m/min). Table 3 summarizes the achieved results in terms of success rate. The values show that higher conveyor speeds require higher vision frequency. However, until nominal speed (5m/min), which is already quite high compared to other state-of-the-art methods, the proposed approach appears to be robust even to a slow perception rate.

The approach has been proven to be usable in different contexts and robust to nominal conditions and obstacles along the path, modulating the advanced velocity accordingly. The model assumption is that the target moves with smooth velocity, which is, though, plausible in many industrial service robotics applications. The failures noted in the aforementioned tests are mostly related to the lack of feedback between the last grasping instants and the reactive grasping controller. Therefore, once the grasping is issued, the action continues even if the target moves without any recovery possibilities. The integration of an in-hand sensor, either tactile or visual, will be considered in future works to further improve the performance of the method. Additionally, the presented solution at the moment can only handle obstacles along the path modulating the advancing velocity, but does not present a proper collision avoidance mechanism that inspects the environment and shapes the line accordingly to avoid obstacles while still going

Table 2
Results of the low-level controller comparison.

Method	Succ. [%]	Fails [%]	Cart.PathLen. [m]	Effort [rad]	Time [s]
OURS	72.727	27.273	1.446	6.450	7.339
PIJ	45.455	54.545	1.599	7.131	9.321
VS	45.455	54.545	1.595	7.100	8.800

Table 3
Sensitivity analysis on the effect of vision frequency rate.

speed	30 Hz	15 Hz	10 Hz	5 Hz
3m/min	100%	100%	100%	70%
5m/min	100%	100%	30%	0
8m/min	100%	100%	0	0

toward the forecasted object position. This aspect will also be explored in a future extension of the work.

6. Conclusions

Robots should exhibit the ability to react to unstructured and uncertain environments, adapting their trajectory online depending on the changes in the surroundings to achieve an efficient and safe collaboration.

Other works in the field of dynamic grasping concentrated on the perception aspect of the problem, proposing always more complex solutions to track and forecast the object or the candidate grasp across frames, trying to ensure consistency. However, the results are limited to low target motion due to the lack of a dynamic control approach. Therefore, this work presents a reactive grasp controller inspired by an established human motion model. The method generates at each time step a straight line in the cartesian space from the current position of the end-effector to the target pose that is then shaped considering the movements of the target object over time. A trace controller makes the end-effector remain on the path and allows scaling the velocity depending on the relative distance between the end-effector and the target, and eventually other factors in the surrounding environment, like the presence of humans. Control actions are sent to the robot controller as smooth velocity commands at 1 KHz to be reactive. The method has been tested and compared with three state-of-the-art methods in three different contexts resembling generic industrial and service robotics tasks: (i) grasping moving objects on a conveyor belt, (ii) grasping objects that can move during the robot's motion toward the target due to human operation actions, and (iii) human-to-robot handover, achieving almost perfect success rate in all of them. Comparisons show that the proposed approach outperforms the baselines.

Future directions involve further development of the collision avoidance mechanism to make the reactive grasp controller even more generally applicable and the integration of a grasping feedback to connect the closure of the gripper with the motion of the robot during the last instances of the grasping.

CRediT authorship contribution statement

Emilio Maranci: Writing – original draft, Visualization, Validation, Software, Formal analysis, Data curation. **Carlo Alberto Avizzano:** Writing – original draft, Resources, Funding acquisition. **Salvatore D'Avella:** Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2026.105401.

Data availability

No data was used for the research described in the article.

References

[1] Salvatore D'Avella, Carlo Alberto Avizzano, Paolo Tripicchio, ROS-industrial based robotic cell for industry 4.0: Eye-in-hand stereo camera and visual servoing for flexible, fast, and accurate picking and hooking in the production line, Robot. Comput.-Integr. Manuf. 80 (2023) 102453.

[2] Muhammad Hamza Zafar, Even Falkenberg Langås, Filippo Sanfilippo, Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review, Robot. Comput.-Integr. Manuf. 89 (2024) 102769.

[3] Luca Gualtieri, Erwin Rauch, Renato Vidoni, Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review, Robot. Comput.-Integr. Manuf. 67 (2021) 101998.

[4] Yizhe Zhang, Lianjun Li, Michael Ripperger, Jorge Nicho, Malathi Veeraraghavan, Andrea Fumagalli, Gilbreth: A conveyor-belt based pick-and-sort industrial robotics application, in: 2018 Second IEEE International Conference on Robotic Computing, IRC, IEEE, 2018, pp. 17–24.

[5] Salvatore D'Avella, Paolo Tripicchio, Carlo Alberto Avizzano, A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper, Robot. Comput.-Integr. Manuf. 63 (2020) 101888.

[6] Peter K Allen, Aleksandar Timcenko, Billibon Yoshimi, Paul Michelman, Automated tracking and grasping of a moving object with a robotic hand-eye system, IEEE Trans. Robot. Autom. 9 (2) (1993) 152–165.

[7] Marc Tuscher, Julian Hörz, Danny Driess, Marc Toussaint, Deep 6-dof tracking of unknown objects for reactive grasping, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 14185–14191.

[8] Farzad Husain, Adria Colomé, Babette Dellen, Guillem Alenya, Carme Torras, Realtime tracking and grasping of a moving object from range video, in: 2014 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2014, pp. 2617–2622.

[9] Iretiayo Akinola, Jingxi Xu, Shuran Song, Peter K. Allen, Dynamic grasping with reachability and motion awareness, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 9422–9429.

[10] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, Cewu Lu, Anygrasp: Robust and efficient grasp perception in spatial and temporal domains, IEEE Trans. Robot. 39 (5) (2023) 3929–3945.

[11] Nuo Chen, Xiao-Ming Wu, Guohao Xu, Jian-Jian Jiang, Zibo Chen, Wei-Shi Zheng, Motiongrasp: Long-term grasp motion tracking for dynamic grasping, IEEE Robot. Autom. Lett. (2024).

[12] David A. Rosenbaum, Ruud G.J. Meulenbroek, Jonathan Vaughan, Planning reaching and grasping movements: theoretical premises and practical implications, Mot. Control. 5 (2) (2001) 99–115.

[13] David A Rosenbaum, Ruud J Meulenbroek, Jonathan Vaughan, Chris Jansen, Posture-based motion planning: applications to grasping., Psychol Rev 108 (4) (2001) 709.

[14] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, et al., Deep learning approaches to grasp synthesis: A review, IEEE Trans. Robot. 39 (5) (2023) 3994–4015.

[15] Arjun Menon, Benjamin Cohen, Maxim Likhachev, Motion planning for smooth pickup of moving objects, in: 2014 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2014, pp. 453–460.

[16] Peter K Allen, Aleksandar Timcenko, Billibon Yoshimi, Paul Michelman, Automated tracking and grasping of a moving object with a robotic hand-eye system, IEEE Trans. Robot. Autom. 9 (2) (2002) 152–165.

[17] Naresh Marturi, Marek Kopicki, Alireza Rastegarpanah, Vijaykumar Rajasekaran, Maxime Adjigble, Rustam Stolkin, Aleš Leonardis, Yasemin Bekiroglu, Dynamic grasp and trajectory planning for moving objects, Auton. Robots 43 (2019) 1241–1256.

[18] Ching-Chang Wong, Ming-Yi Chien, Ren-Jie Chen, Hisayuki Aoyama, Kai-Yi Wong, Moving object prediction and grasping system of robot manipulator, IEEE Access 10 (2022) 20159–20172.

[19] Jirong Liu, Ruo Zhang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Chenxi Wang, Sheng Xu, Hengxu Yan, Cewu Lu, Target-referenced reactive grasping for dynamic objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 8824–8833.

[20] Yu Huang, Daxin Liu, Zhenyu Liu, Ke Wang, Qide Wang, Jianrong Tan, A novel robotic grasping method for moving objects based on multi-agent deep reinforcement learning, Robot. Comput.-Integr. Manuf. 86 (2024) 102644.

[21] Zhangpeng Tu, Canjun Yang, Xin Wu, Yuanchao Zhu, Weitao Wu, Ningbo Jia, Moving object flexible grasping based on deep reinforcement learning, in: 2022 8th International Conference on Control, Automation and Robotics, ICCAR, IEEE, 2022, pp. 34–39.

[22] Baichuan Huang, Jingjin Yu, Siddarth Jain, EARL: Eye-on-hand reinforcement learner for dynamic grasping with active pose estimation, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2023, pp. 2963–2970.

[23] Gitae Kang, Yong Bum Kim, Young Hun Lee, Hyun Seok Oh, Won Suk You, Hyouk Ryeol Choi, Sampling-based motion planning of manipulator with goal-oriented sampling, Intell. Serv. Robot. 12 (2019) 265–273.

[24] Julius Jankowski, Lara Brudermüller, Nick Hawes, Sylvain Calinon, Vp-sto: Via-point-based stochastic trajectory optimization for reactive robot behavior, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 10125–10131.

[25] Lara Brudermüller, Guillaume Berger, Julius Jankowski, Raunak Bhattacharyya, Raphaël Jungers, Nick Hawes, CC-VPSTO: Chance-constrained via-point-based stochastic trajectory optimisation for safe and efficient online robot motion planning, 2024, arXiv preprint arXiv:2402.01370.

[26] Mohak Bhardwaj, Balakumar Sundaralingam, Arsalan Mousavian, Nathan D Ratliff, Dieter Fox, Fabio Ramos, Byron Boots, Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation, in: Conference on Robot Learning, PMLR, 2022, pp. 750–759.

[27] Vasileios Vasilopoulos, Suveer Garg, Pedro Piacenza, Jinwook Huh, Volkan Isler, Ramp: Hierarchical reactive motion planning for manipulation tasks using implicit signed distance functions, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2023, pp. 10551–10558.

[28] Douglas Morrison, Peter Corke, Jürgen Leitner, Learning robust, real-time, reactive robotic grasping, Int. J. Robot. Res. 39 (2–3) (2020) 183–201.

[29] Stefan Forrest Campbell, Steering Control of an Autonomous Ground Vehicle with Application to the DARPA Urban Challenge (Ph.D. thesis), Massachusetts Institute of Technology, 2007.

[30] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, Aaron M Dollar, Yale-CMU-berkeley dataset for robotic manipulation research, Int. J. Robot. Res. 36 (3) (2017) 261–268.