Experiments of Visual Servoing in Presence of Humans based on Task Priority

Giovanni Francesco Comune, Luca Bianco, Gianmarco Di Puorto, Marco Costanzo, and Ciro Natale

***Abstract*— This paper presents a visual servoing system for human-robot interaction that emulates mutual gaze and joint attention. A YOLOv8n module detects human faces or objects of a given category, and the image-plane error is regulated to zero via an Image-Based Visual Servoing (IBVS) law implemented within a Singularity-Robust Task-Priority framework. Four tasks are arranged in a strict hierarchy: joint limit avoidance, collision avoidance, camera roll stabilisation and IBVS tracking. This hierarchy guarantees that safety constraints are never sacrificed for tracking performance, while redundant degrees of freedom are used to maintain the image horizon level and keep the detected entity to the center of the image plane. Experimental results demonstrate robust tracking, strict enforcement of all constraints and the qualitative benefits of roll stabilization for achieving natural gaze behavior.**

I. INTRODUCTION

Stemming from the increasing integration of robots in domestic environments –not only for assisting elderly people or individuals with disabilities– Human-Robot Interaction (HRI) has become a topic of growing interest. In social settings, robots are expected to engage with humans and display intentionality through natural communication. Modern socially interactive robots pursue this goal using a series of non-verbal, *attention-based* mechanisms [1] that reinforce verbal communication. Among these, *eye gaze* is a key mechanism for social engagement, typically categorized into *mutual gaze* (i.e., eye contact), *referential or deictic gaze*, *joint attention*, and *gaze aversions* [2]. In particular, reciprocal gaze has a crucial effect on attribution of human-likeness [3], which motivates the need for technical solutions to ensure the robot consistently orients toward its interlocutor, involving a dedicated vision system and a specifically designed control strategy. Traditionally, these abilities have been reserved for humanoid robots; however, there is the risk that such robots can trigger the uncanny valley phenomenon [4], hence this work focuses on a standard robotic arm that moves a camera emulating robot eyes. Although many works address eye gaze in HRI, they primarily focus on the detection of mutual gaze [5], [6], the classification of gaze types [7], [8], or their effects on humans [9], with low attention dedicated to achieving social skills while respecting robots' inherent kinematic and control limitations. This work proposes an approach based on visual servoing for mutual gaze and explicit joint attention that accounts for robot-related constraints. The core of the proposed approach is to command a robot manipulator equipped with an eye-in-hand camera that serves as the robot's visual apparatus, thereby directing the "robot gaze" to consistently center the human interlocutor's face within the image plane. Although this appears to be the highest-priority objective, it is nonetheless subordinated to joint-limit compliance and collision avoidance. Consequently, tracking of the human face is achieved through a multi-task kinematic control strategy with task hierarchy, wherein the gaze objective remains subordinate to safety constraints. Control algorithms that pursue this goal have been investigated since the origins of robotics, initially considering only the execution of two tasks with relative priority [10] and finally generalized to an arbitrary number of tasks [11]. To cope with the kinematic and algorithmic singularities of this generalized scheme, [12] proposed the Singularity-Robust Task-Priority (SRTP) formulation. A rigorous stability analysis of task-priority closed-loop inverse kinematics (CLIK) was later provided by [13]. A scheme to compute a control law for a set of bilateral tasks and unilateral constraints, typical of visual servoing, is then proposed within the Stack-of-Tasks framework [14]. More recent work [15] proposes the Saturation in the Null Space (SNS) approach to account for hard joint constraints. This formulation was then extended in [16], which proposes a unified framework to resolve redundancy at the velocity, acceleration, and torque levels, denoted eSNS in its basic form and Fast-eSNS and Opt-eSNS in its enhanced variants.

We present here a system that integrates visual servoing of a robotic arm to detect and track human faces, thereby establishing and maintaining mutual gaze with an interlocutor. This capability is further enriched with an explicit joint-attention mechanism to track an object chosen by the human. Our contributions are as follows:

- We develop a detection module to identify human faces and objects using the YOLOv8n network.
- We adopt a task-priority approach to implement Image-Based Visual Servoing (IBVS), explicitly considering the kinematic and physical constraints of the robot.
- We provide experimental validation in both simulated and real-world environments to evaluate the effectiveness of the proposed algorithms.

II. METHOD

A. Image-Based Visual Servoing

The vision system is modelled as a standard pinhole camera. A 3D point $\boldsymbol{p} = [p_x, p_y, p_z]^\top$ expressed in the camera frame maps to pixel coordinates

$$u = f_x \frac{p_x}{p_z} + c_x, \qquad v = f_y \frac{p_y}{p_z} + c_y, \tag{1}$$

All the authors are with Dipartimento di Ingegneria, Università degli Studi della Campania "Luigi Vanvitelli", 81031 Aversa, Italy, marco.costanzo@unicampania.it

where f_x, f_y are the focal lengths in pixels, (c_x, c_y) is the principal point, and $p_z \equiv Z > 0$ is the depth.

Face or object detection is handled by a YOLOv8n network, which processes each incoming colour frame and returns an axis-aligned bounding box for every detected entity. The visual feature is the centroid of the bounding box, $\boldsymbol{s} = [u,\, v]^\top \in \mathbb{R}^2$, expressed directly in pixel coordinates. The desired features $\boldsymbol{s}^* = [c_x,\, c_y]^\top$ correspond to the image centre, so that the centred coordinates $\tilde{u} = u - c_x$ and $\tilde{v} = v - c_y$ represent the alignment error between the detected centroid and the optical axis. The control error is accordingly $\boldsymbol{e} = \boldsymbol{s} - \boldsymbol{s}^*$.

The time derivative of $\boldsymbol{s}$ relates to the camera spatial velocity $\boldsymbol{v}_c \in \mathbb{R}^6$, expressed in the camera frame, through the image Jacobian [17]:

$$\dot{\boldsymbol{s}} = \boldsymbol{L}_{\mathrm{img}}\, \boldsymbol{v}_c, \tag{2}$$

where $\boldsymbol{v}_c = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^\top$. For a point at depth Z, the image Jacobian is the 2×6 matrix

$$\boldsymbol{L}_{\mathrm{img}} = \begin{pmatrix} \frac{\tilde{u}}{Z} & -\frac{f_x}{\rho_u Z} & 0 & \tilde{v} & \frac{\rho_u\, \tilde{u}\tilde{v}}{f_x} & -\frac{f_x^2 + \rho_u^2 \tilde{u}^2}{\rho_u f_x} \\ \frac{\tilde{v}}{Z} & 0 & -\frac{f_y}{\rho_v Z} & -\frac{\rho_v\, \tilde{u}\tilde{v}}{f_y} & \frac{f_y^2 + \rho_v^2 \tilde{v}^2}{\rho_v f_y} & -\tilde{u} \end{pmatrix}, \tag{3}$$

with scale factors $\rho_u = 1/\alpha_x$, $\rho_v = 1/\alpha_y$, where α_x, α_y are the pixel-to-metre conversion factors. As usual in image-based visual servoing, given the low sensitivity to the value of Z [17], this is kept constant at an average value foreseen for the specific application.

The camera is fixed to the end-effector flange, with the camera frame assumed to be aligned with the tool frame. Given that the camera velocity $\boldsymbol{v}_c$ is expressed in the camera frame, the joint velocity vector $\dot{\boldsymbol{q}}_{\mathrm{ibvs}}$ is obtained by combining the image Jacobian with the robot Jacobian expressed in the camera frame $\boldsymbol{J}_{\mathrm{cam}} \in \mathbb{R}^{6\times 6}$:

$$\dot{\boldsymbol{q}}_{\mathrm{ibvs}} = -\lambda \left(\boldsymbol{L}_{\mathrm{img}}\, \boldsymbol{J}_{\mathrm{cam}}\right)^\dagger \boldsymbol{e}, \tag{4}$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse and λ is the control gain. The camera-frame Jacobian is obtained from the base-frame Jacobian $\boldsymbol{J}$ via:

$$\boldsymbol{J}_{\mathrm{cam}} = \begin{pmatrix} \boldsymbol{R}^{\mathrm{cam}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^{\mathrm{cam}} \end{pmatrix} \boldsymbol{J}, \tag{5}$$

where $\boldsymbol{R}^{\mathrm{cam}} \in SO(3)$ is the rotation matrix from the base frame to the camera frame, obtained at each control cycle from the forward kinematics.

B. Task-Priority Framework

Considering that the robot moving the camera is assumed to have six joints, and the visual task is two-dimensional (u and v), four degrees of freedom are redundant and can be used to satisfy additional objectives. The task-priority approach [11] assigns a strict order to these objectives: lower-priority tasks can only use motions in the null space of all higher-priority ones, so they never interfere with them.

Following the Singularity-Robust Task Priority formulation [12] to mitigate the algorithmic singularities arising from [11], the contribution of task i to the joint velocity is computed as

$$\dot{\boldsymbol{q}}_i = \boldsymbol{P}_{i-1}\, \boldsymbol{J}_i^\dagger\, \boldsymbol{v}_i, \tag{6}$$

where $\boldsymbol{J}_i^\dagger$ is the Moore-Penrose pseudo-inverse of the task Jacobian $\boldsymbol{J}_i$, and $\boldsymbol{P}_{i-1}$ is the null-space projector of all higher-priority tasks. The projector is updated recursively:

$$\boldsymbol{P}_i = \boldsymbol{P}_{i-1}\big(\boldsymbol{I} - \boldsymbol{J}_i^\dagger\, \boldsymbol{J}_i\big), \qquad \boldsymbol{P}_0 = \boldsymbol{I}. \tag{7}$$

In the classical closed-loop fashion, each task acts as a proportional regulator on the task error, $\boldsymbol{v}_i = \boldsymbol{K}_i\, \boldsymbol{e}_i$, and the total joint command is computed by cumulating all n_t task contributions, i.e.,

$$\dot{\boldsymbol{q}}_{\mathrm{tot}} = \sum_{i=1}^{n_t} \dot{\boldsymbol{q}}_i. \tag{8}$$

When a task is inactive, its Jacobian is set to a 0×6 matrix; consequently, $\boldsymbol{P}_i = \boldsymbol{P}_{i-1}$ and the entire null space are passed to the next task. This allows constraints to be activated and deactivated dynamically without any discontinuity in the velocity command.

As mentioned in the introduction, although the main task is the face tracking of the human partner, the robot has to accomplish it properly; namely, it must respect its joint position and velocity limits, avoid collisions with other objects in the scene, and try to optimize the camera orientation to avoid configurations that might be experienced as weird by the human. To achieve this objective, four tasks are assigned to the robot and arranged in the hierarchy listed below, from safety-critical tasks (1 and 2) to functional tasks (3 and 4).

Task 1 – Joint Limit Avoidance: For joint j, let $d^j_{\mathrm{low}} = q_j - q_{j,\min}$ and $d^j_{\mathrm{up}} = q_{j,\max} - q_j$ be the distances from the lower $q_{j,\min}$ and upper $q_{j,\max}$ hard limits, respectively. The following cubic Hermite smoothstep function

$$\alpha(d) = 1 - \left(\frac{d}{\bar{d}}\right)^2 \left(3 - \frac{2d}{\bar{d}}\right), \quad d < \bar{d}, \tag{9}$$

with $\bar{d} = 10°$ is defined to modulate the repulsive velocity

$$\dot{\boldsymbol{q}}_1 = \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_6 \end{pmatrix}, \ \dot{q}_j = \begin{cases} \alpha(d^j_{\mathrm{low}}) & \text{if } d^j_{\mathrm{low}} < \bar{d} \\ -\alpha(d^j_{\mathrm{up}}) & \text{if } d^j_{\mathrm{up}} < \bar{d} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The corresponding task Jacobian $\boldsymbol{J}_{\mathrm{lim}} \in \mathbb{R}^{m\times 6}$ selects the rows of $\boldsymbol{I}_6$ for the $m \geq 0$ currently active joints, i.e., when $q_j \in [q_{j,\min} + \bar{d}, q_{j,\max} - \bar{d}]$. The smoothstep function guarantees that $\alpha(\bar{d}) = 0$ and $\alpha'(\bar{d}) = 0$, ensuring C^1 continuity at the activation boundary and eliminating velocity discontinuities when a joint enters or leaves the repulsive zone.

Task 2 – Collision Avoidance: The objective of the second task is to keep the end effector above the support surface of the arm, modelled as a horizontal plane at z-coordinate $z_{\min}$ in the base frame. Let $d = z_{\mathrm{ee}} - z_{\min}$ be the signed distance from the hard limit, the task Jacobian is obviously the third row of the base-frame robot Jacobian $\boldsymbol{J}$, which maps joint velocities into the end-effector velocity

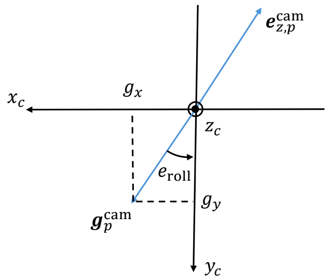

Fig. 1. Definition of the task error of the camera roll angle e_{roll}: x_c, y_c and z_c are the camera frame axes; g_p^{cam} and $\boldsymbol{e}_{z,p}^{\mathrm{cam}}$ are the projections of $\boldsymbol{g}^{\mathrm{cam}}$ and $\boldsymbol{R}^{\mathrm{cam}}\boldsymbol{e}_z$ onto x_c, y_c plane.

$\dot{z}_{\mathrm{ee}}$ along the z axis of the base frame. The joint-velocity contribution is then

$$\dot{\boldsymbol{q}}_2 = k_{\mathrm{col}}\, \boldsymbol{P}_1\, \boldsymbol{J}_{\mathrm{col}}^{\dagger}\, \alpha(d), \tag{11}$$

where $k_{\mathrm{col}} > 0$ is the task gain and $\alpha(d)$ is defined in (9) with the activation threshold set to $\bar{d} = 0.03\,\mathrm{m}$. When $d \geq \bar{d}$, the Jacobian $\boldsymbol{J}_{\mathrm{col}}$ is set to a 0×6 matrix, and the velocity $\dot{\boldsymbol{q}}_2$ is zero, passing the entire null space to lower-priority tasks.

Task 3 – Roll Stabilisation: This task is aimed at keeping, as much as possible compatibly with higher priority tasks, the roll orientation angle of the camera close to zero so that the image horizon remains level. Let $\boldsymbol{g}^{\mathrm{cam}} = (g_x\ g_y\ g_z)^{\top} = -\boldsymbol{R}^{\mathrm{cam}}\, \boldsymbol{e}_z$ be the z axis of the base frame expressed in the camera frame, $\boldsymbol{e}_z = (0\ 0\ 1)^{\top}$. As shown in Fig. 1, the roll error is the angle that the projection of $\boldsymbol{g}^{\mathrm{cam}}$ onto the camera x, y plane makes with the y axis of the camera frame, i.e.,

$$e_{\mathrm{roll}} = \mathrm{atan2}(g_x, g_y). \tag{12}$$

This error is zero when the image horizon is level. The corresponding task Jacobian $\boldsymbol{J}_{\mathrm{roll}}$ is simply the last row of the matrix $\boldsymbol{J}_{\mathrm{cam}}$ that maps joint velocities to the z component of the camera angular velocity. The control joint velocity for roll stabilisation is

$$\dot{\boldsymbol{q}}_3 = k_{\mathrm{roll}} \boldsymbol{P}_2 \boldsymbol{J}_{\mathrm{roll}}^{\dagger} e_{\mathrm{roll}}, \tag{13}$$

being $k_{\mathrm{roll}} > 0$ the task gain.

Task 4 – IBVS Tracking: The visual tracking law (4) is projected through the null space of all previous tasks as

$$\dot{\boldsymbol{q}}_4 = -\lambda\, \boldsymbol{P}_3 \left(\boldsymbol{L}_{\mathrm{img}}\, \boldsymbol{J}_{\mathrm{cam}}\right)^{\dagger} \boldsymbol{e}. \tag{14}$$

Placing the visual task at the lowest priority ensures the safety constraints are never sacrificed for tracking performance. When a safety task is active, the tracking objective is fulfilled only in the remaining null space; if the null space is exhausted, tracking degrades without violating a constraint.

Finally, to prevent commands from exceeding joint velocity limits, the total velocity is scaled uniformly by the most restrictive joint factor, i.e.,

$$\chi = \min_j \left(1,\ \frac{\dot{q}_{j,\max}}{|\dot{q}_{\mathrm{tot},j}|}\right), \qquad \dot{\boldsymbol{q}}_{\mathrm{sat}} = \chi\, \dot{\boldsymbol{q}}_{\mathrm{tot}}, \tag{15}$$

which preserves the direction of motion while keeping all joints within the limits.

TABLE I
MECA500 JOINT VELOCITY AND POSITION LIMITS

Joints	$\dot{q}_{\max}$ [°/s]	$q_{\min}, q_{\max}$ [°]
1–2	150	± 175
3	180	± 70
4–5	300	± 175
6	500	± 180

Fig. 2. Snaps of the object (top) and face (bottom) tracking experiment. Robot view and tracked object (right); external view (left).

III. EXPERIMENTAL RESULTS

A. Real-Time Software Architecture

The experimental setup is depicted in Fig. 2-top-left. A Meca500 robot is equipped with an Intel RealSense D435i camera rigidly mounted as end effector. The robot kinematic limits are reported in Tab. I. The camera streams a 640×480 RGB image at 30 fps. The image is elaborated by a YOLOv8n model that runs at 7.5 Hz. The algorithms run on a computer equipped with an Intel i7-12650H processor with 32 GB of RAM. The robot joints are velocity-controlled via EtherCAT. To ensure bounded latency, the standard Linux scheduling is insufficient, *Xenomai 3* [18] is adopted: it runs a hard-real-time co-kernel (Cobalt) alongside the Linux kernel, guaranteeing worst-case jitter below 50 µs. Everything is coordinated with ROS 2 Humble, while CycloneDDS is used as the RMW implementation. The control algorithm described in the previous sections runs in a dedicated ROS node at 100 Hz. The control parameters $z_{\min} = 0.08\,\mathrm{m}$, $k_{\mathrm{col}} = 1$, and $k_{\mathrm{roll}} = 1$ are chosen according to the guidelines in [19] as a trade-off between stability and desired performances. In IBVS, Z is set to 1 m and the gain $\lambda = 1$ is chosen to ensure stability, given the low inference rate achieved by YOLOv8n. A video reporting the real-world experiments, as well as an additional simulation experiment, is available at `https://youtu.be/lpGeq0yCTu0`.

B. Object Tracking

Two experiments were conducted to evaluate the framework. The first demonstrates the system's ability to track an object held by a human operator, namely a smartphone (Fig. 2-top). Figure 3 illustrates the image-plane error

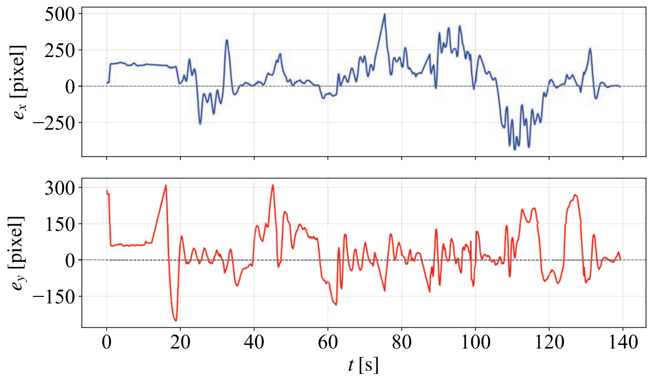

Fig. 3. Object tracking experiment: Image-plane tracking error $\boldsymbol{e} = [e_u, e_v]^\top$ over time.

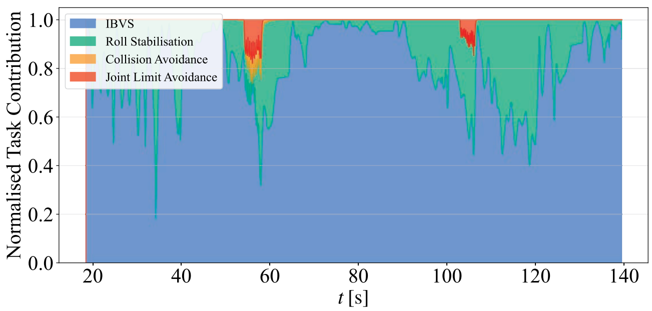

Fig. 4. Object tracking experiment: Normalised task contributions over time.

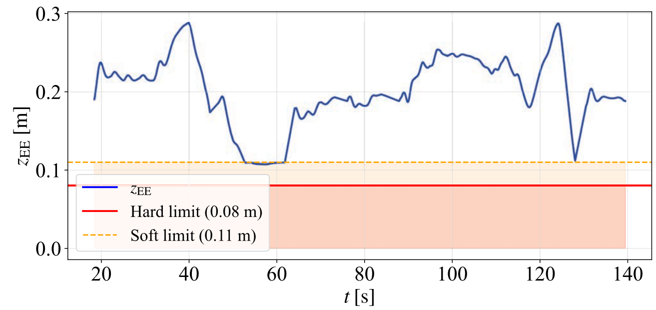

Fig. 5. Object tracking experiment: End-effector height z_{ee} over time.

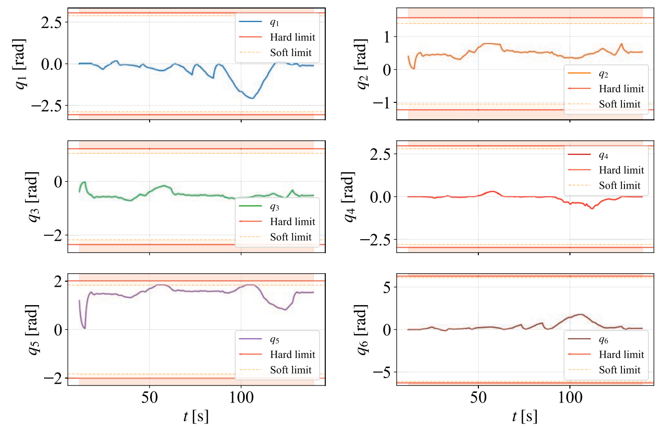

Fig. 6. Object tracking experiment: Joint angle profiles $q_{1\text{–}6}$ with safety zones. Hard limits (red) and smoothstep activation boundaries (dashed orange) are superimposed.

throughout the trial. To provide context for the tracking accuracy, a $100\,\text{px}$ error corresponds to a lateral offset of approximately $0.1\,\text{m}$ at a depth of $Z = 1\,\text{m}$. The error stays mostly within the $\pm 100\,\text{px}$ band on both axes during normal tracking. The larger transient spikes, reaching up to $\pm 450\,\text{px}$ in u and $\pm 300\,\text{px}$ in v, occur when the object moves quickly or briefly leaves the camera field of view, causing YOLO to lose the target; once detection recovers, the controller brings the centroid back to the image centre within a few seconds, demonstrating the robustness of the proportional IBVS law to re-acquisition transients.

Figure 4 reports the contribution of each task to the total joint-velocity norm, normalised to sum to 1. The IBVS task is the primary driver of motion throughout the experiment, typically accounting for more than half of the total norm during active tracking. As expected, roll stabilisation contributes continuously and significantly; as the arm pans to follow the object, joint configuration changes induce a non-negligible camera roll that must be corrected. Joint limit avoidance and collision avoidance activate only when the robot actually approaches a constraint boundary, with no spurious triggers observed throughout the experiment.

The end-effector height in Fig. 5 never drops below $0.08\,\text{m}$. The minimum height recorded during the experiment was $0.107\,\text{m}$ ($27\,\text{mm}$ above the hard limit). The collision avoidance task engages smoothly each time the end-effector descends into the activation zone and pushes it back up.

Figure 6 shows that all joints stay well within their mechanical limits throughout the run. The minimum angular margin observed was $5°$ on joint 3, confirming that the repulsive field provides an adequate safety buffer. No oscillation was observed near any limit boundary, validating the C^1 smoothness property of the activation function.

C. Face Tracking

The second experiment aims to demonstrate the capability of the system in tracking an operator's face, simulating a *mutual gaze* (Fig. 2-bottom). Given the robot's configuration and the fact that the operator's face is naturally positioned higher than the object in the previous trial, the hard safety height limit was set to $z_{\min} = 0.2\,\text{m}$ to rigorously evaluate the collision avoidance system. Additionally, because the operator cannot move behind the robot, the hard joint limit for q_1 was restricted to $\pm 50°$ to ensure the joint-limit activation could be effectively triggered and tested. Moreover, in order to show the benefit of the roll stabilization task, an ablation study has been carried out; the experiment has been repeated two times, with and without the roll task, respectively.

The image error and the joint velocities control input are shown in Fig. 7 and Fig. 8, respectively. Once again the error is quite limited and reaches larger values only during the operator's quick motions. The C^1 continuity of the smoothstep ensures no abrupt velocity changes are

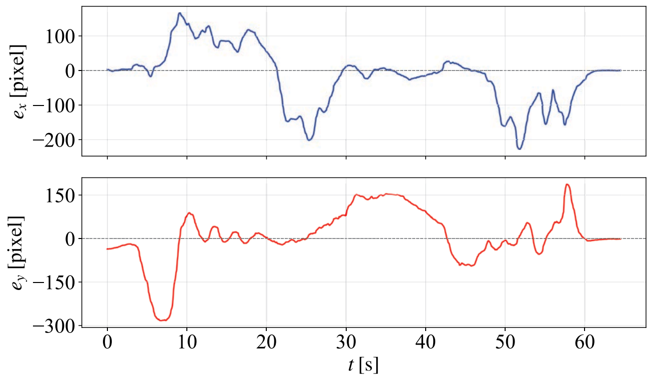

Fig. 7. Face tracking experiment: Image-plane tracking error $\boldsymbol{e} = [e_u,\ e_v]^\top$ over time.

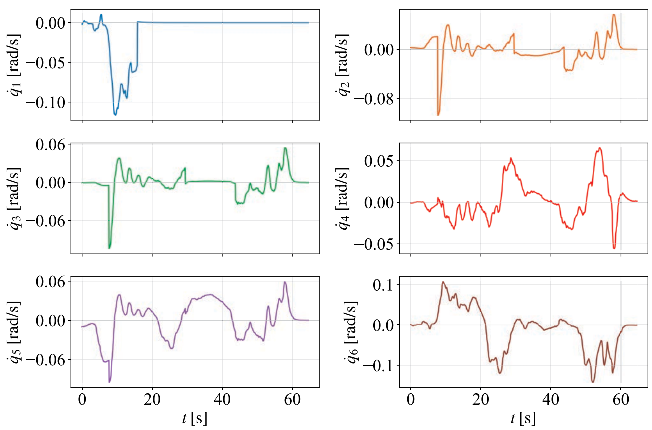

Fig. 8. Face tracking experiment: Joint velocity profiles $\dot{q}_{1\text{–}6}$. All velocities remain within the limits of Table I.

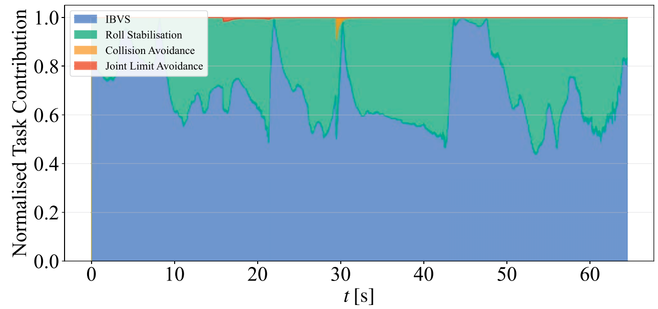

Fig. 9. Face tracking experiment: Normalised task contributions over time.

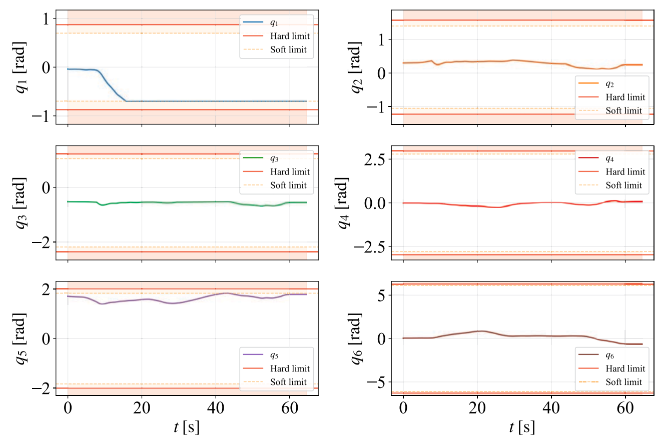

Fig. 10. Face tracking experiment: Joint angle profiles $q_{1\text{–}6}$ with safety zones.

introduced during engagement or disengagement of the tasks, as confirmed by the smooth velocity profiles throughout the experiment. Moreover, the directional scaling of (4) ensures that no limit is ever exceeded, without introducing any abrupt change in motion direction.

The contribution of each task is shown in Fig. 9. As in the previous experiment, most of the joint velocities are given by tasks 3 and 4, while tasks 1 and 2 are active only in specific time intervals. Figure 10 shows the joint positions and limits. Joint 1 reaches and keeps its soft limit at $t = 15$ s, while joint 5 briefly reaches the soft limit at $t = 42$ s. Once again, the higher priority task is able to keep the joints within the prescribed margins. Figure 11 shows the end-effector height. The system behaves as expected with performances comparable to those obtained with the object tracking.

Figure 12 reports task contributions for the additional experiment without the roll stabilization task. This time, practically all the joint velocity is given by the IBVS task with very limited (but necessary) contributions of the higher priority tasks. The reason for the task activation can also be identified in the end effector height and the joint position plots in Fig. 13 and Fig. 14, respectively. The control objective is achieved, and the limits are enforced by the high-level tasks. However, since no roll correction is applied, the final result is very different; in fact, as is shown in Fig. 15, the robot reaches a tilted gaze configuration that makes the human-robot interaction unnatural.

IV. Conclusions

This paper presented a task-priority visual servoing system for a robotic arm designed to track objects and maintain mutual gaze with a human. The described four-level hierarchy, built with the SRTP framework, subordinates IBVS tracking to joint limit avoidance, collision avoidance and camera roll stabilization, ensuring that kinematic and safety constraints are respected without a dedicated switching logic. The smoothstep activation functions guarantee C^1 continuity at constraint boundaries, preventing discontinuities when tasks enter or leave their activation regions. Real-robot experiments on the Meca500 robotic arm confirmed that all joint position limits and the end-effector height constraint were always respected, with the controller recovering gracefully from target loss due to fast motion or occlusions. An ablation study demonstrated that omitting the roll stabilisation task, while still achieving the tracking objective, leads to a tilted camera configuration that makes the resulting gaze unnatural and unsuitable for human-robot interactions.

References

[1] J. F. Ferreira and J. Dias, “Attentional mechanisms for socially interactive robots–a survey,” *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 2, pp. 110–125, 2014.

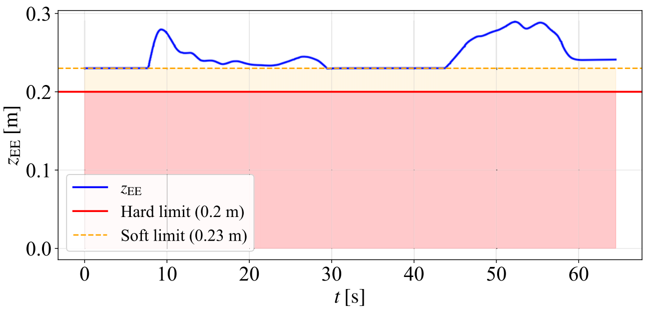

Fig. 11. Face tracking experiment: End-effector height z_{ee} over time.

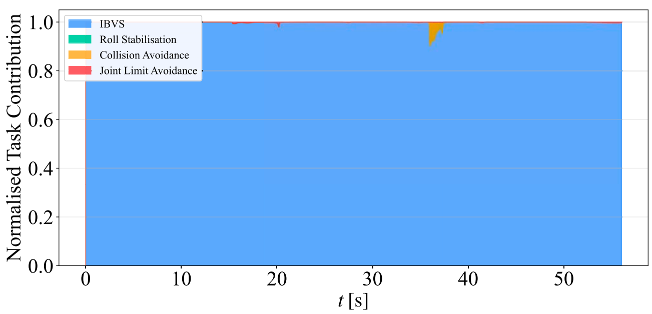

Fig. 12. Face tracking without roll stabilisation experiment: Normalised task contributions over time.

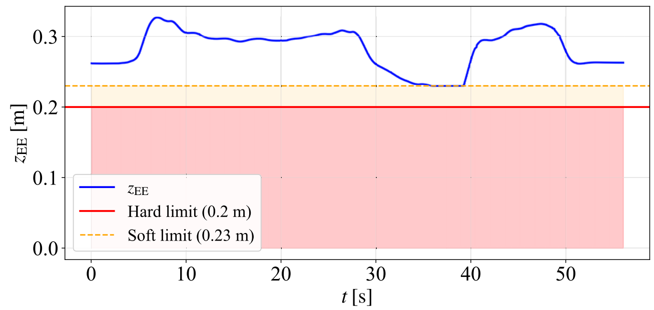

Fig. 13. Face tracking without roll stabilisation experiment: End-effector height z_{ee} over time.

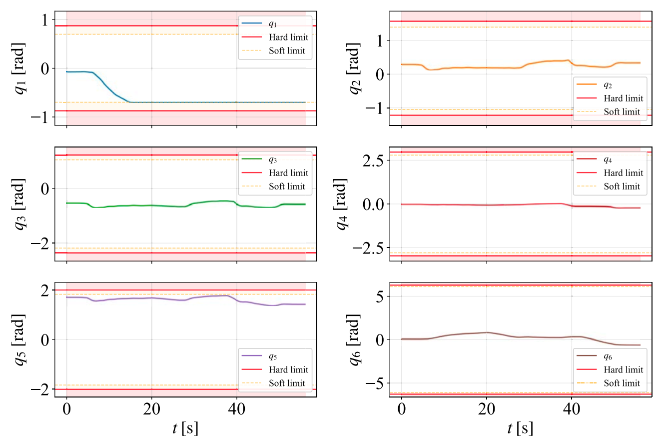

Fig. 14. Face tracking without roll stabilisation experiment: Joint angle profiles q_{1-6} with safety zones.

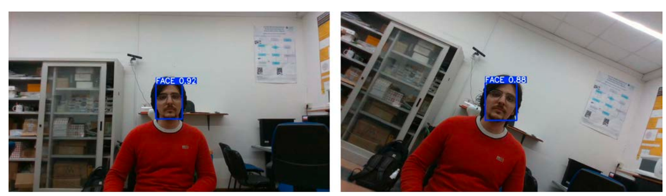

Fig. 15. Comparison of the results with (left) and without (right) the roll stabilisation task.

[2] H. Admoni and B. Scassellati, "Social eye gaze in human-robot interaction: a review," *Journal of Human-Robot Interaction*, vol. 6, no. 1, pp. 25–63, 2017.

[3] K. Kompatsiari, V. Tikhanoff, F. Ciardo, G. Metta, and A. Wykowska, "The importance of mutual gaze in human-robot interaction," in *Social Robotics*, A. Kheddar, E. Yoshida, S. S. Ge, K. Suzuki, J.-J. Cabibihan, F. Eyssel, and H. He, Eds. Cham: Springer International Publishing, 2017, pp. 443–452.

[4] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," *IEEE Robotics & automation magazine*, vol. 19, no. 2, pp. 98–100, 2012.

[5] S. Arreghini, G. Abbate, A. Giusti, and A. Paolillo, "A long-range mutual gaze detector for hri," in *Proceedings of the 2024 ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2024, pp. 870–874.

[6] M. Lombardi, E. Maiettini, D. De Tommaso, A. Wykowska, and L. Natale, "Toward an attentive robotic architecture: Learning-based mutual gaze estimation in human–robot interaction," *Frontiers in Robotics and AI*, vol. 9, p. 770165, 2022.

[7] F. Schirmer, P. Kranz, C. G. Rose, V. Willert, J. Schmitt, and T. Kaupp, "Utilizing eye gaze for human-robot collaborative assembly," in *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2025, pp. 1603–1607.

[8] L. Shi, C. Copot, and S. Vanlanduit, "Visual intention classification by deep learning for gaze-based human-robot interaction," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 750–755, 2020.

[9] E.-H. Roy and M. K. Pan, "The social life of industrial arms: How arousal and attention shape human-robot interaction," in *2025 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2025, pp. 930–935.

[10] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[11] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, 1991, pp. 1211–1216 vol.2.

[12] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[13] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, 2009.

[14] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[15] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[16] M. D. Fiore, G. Meli, A. Ziese, B. Siciliano, and C. Natale, "A general framework for hierarchical redundancy resolution under arbitrary constraints," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2468–2487, 2023.

[17] F. Chaumette and S. Hutchinson, "Visual servo control I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[18] Xenomai, "Xenomai real-time framework," 2024, https://v3.xenomai.org/overview/ [Accessed (Jan., 2026)].

[19] P. Falco and C. Natale, "On the stability of closed-loop inverse kinematics algorithms for redundant robots," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 780–784, 2011.