A Simple and Effective Human-Robot Shared Control Approach for Agricultural Robots Employed in Table-Grape Vineyards

Jozsef Palmieri[1], Paolo Di Lillo[1], Stefano Chiaverini[1] , Alessandro Marino[1]

***Abstract*— This paper introduces a simple yet effective shared control architecture to enable a human operator to enhance robot autonomy in complex scenarios, such as those encountered in agriculture. Specifically, the proposed approach investigates dynamically exchanging control between the robotic system and a human operator depending on the uncertainty in the environment perception. This is achieved by envisioning three control modes that exhibit varying levels of autonomy for the robot: *i*) autonomous, *ii*) teleoperation and *iii*) hand-guiding control modes. The approach is embedded into a Hierarchical Quadratic Programming (HQP) control framework, that allows the robot to simultaneously address different control objectives with different priorities. Experimental results conducted as part of the EU-funded CANOPIES project, involving a mobile robot equipped with a mobile base and a dual-arm torso engaged in harvesting operations, demonstrate the effectiveness of the proposed method in real-world conditions.**

I. Introduction

Precision agriculture has emerged as a promising approach that incorporates robotics and automation into farming practices. However, achieving complete autonomy for such systems is often impractical or economically inefficient for many tasks in real-world scenarios including the agricultural domain, but also more widely. In recent years, there has been a growing interest in integrating human-robot collaboration in various industries and even domestic environments. This shift is mainly motivated by the fact that humans and robots possess unique and complementary capabilities. By combining these capabilities, the performance in task execution can be significantly improved, leading to enhanced productivity. This collaboration can take different forms, such as *i*) sharing a workspace where humans and robots work together on different tasks or *ii*) engaging in collaborative tasks where humans contribute with their cognitive skills and possibly exchange forces with robots.

In the first scenario, the main objective of the robot control strategy is to ensure the safety of humans by preventing any potential harm that may arise from collisions between the robot and the human [1]. This can be accomplished through various methods, such as implementing evasive actions to create distance between the robot and the person [2] or utilizing dynamic trajectory scaling techniques such as in [3].

[1]University of Cassino and Southern Lazio, Italy
Corresponding author: Jozsef Palmieri. Email: jozsef.palmieri@unicas.it

This work was supported by H2020-ICT project CANOPIES-A Collaborative Paradigm for Human Workers and Multi-Robot Teams in Precision Agriculture Systems (Grant Agreement N. 101016906) and has received funding from Project COM^3 CUP H53D23000610006 funded by EU in NextGenerationEU plan through the Italian "Bando Prin 2022 - D.D. 104 del 02-02-2022" by MUR.

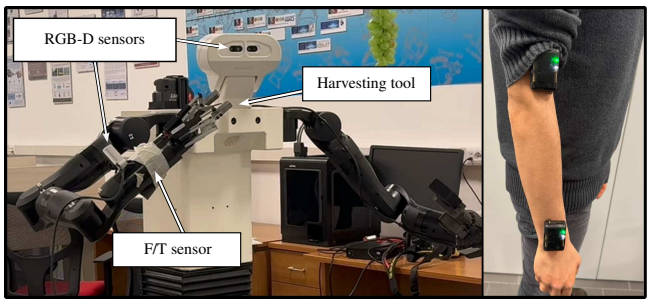

Fig. 1: On the left, the robot platform. On the right, the IMU sensors mounted on the human operator's harm.

More interesting is the case where the synergy between humans and robots is fully exploited. In this regard, there is a growing emphasis on *shared control* scenarios. These scenarios resemble *human-human* interactions, where robot autonomy is retained to a certain extent, and equal roles might be assigned to both robotic and human counterparts. The shared control paradigm is adopted, for example, in teleoperation scenarios [4], where the human operator typically inputs control commands via a haptic interface. Concurrently, the robotic system maintains autonomous behaviors, such as, for example, collision avoidance, by relying on its perception capabilities.

Several shared control approaches in the context of physical human-robot interaction have also been proposed in the last decades [5]; for example, a game theory-based solution is introduced in [6], where it is assumed that both the human and the robot aim to optimize an identical cost function. Within this proposed approach, the roles of both the human and the robot evolve dynamically in response to the force exerted by the human operator. More precisely, as the force applied by the human increases, so does the degree of influence exerted by the intended human motion on the system. In the work outlined in [7], a heuristic agreement index is introduced. This index governs the robot's role based on the alignment of forces between the robot and the human; when their forces are in agreement, the robot assumes a leadership role; otherwise, it follows the human's lead. Furthermore, in [8], a solution based on data-driven stochastic modeling is devised. It addresses uncertainties in the human behavior model through the formulation of a risk-sensitive optimization problem. More recently, authors in [9] propose achieving a similar behavior by adjusting the robot's trajectory in response to the exerted human forces. To accomplish this, a constrained optimization problem is formulated with this specific objective in mind. In a complementary approach, the authors in [10] propose the development of a systematic approach for impedance

parameter adaption in physical human interaction. In detail, the goal is the seamless and intuitive transition of control authority between the two agents, and to achieve this goal, a cost function is defined. This cost function, which is designed in such a way as to ensure the safety of collaborative tasks and to consider the assistive behavior of the automation system, is employed within a predictive controller based on a modified least squares method to modulate the automation system's impedance, optimizing the defined cost function. A different approach for a similar problem is presented in [11]. In detail, the control allocation is determined using a metric derived from a Bayesian filter, which is continuously adapted based on online sensor measurements, accounting also for the variability of measurement noise characteristics. A stability proof for the proposed shared control architecture is devised, even when there are time delays in communication between the operator and the robot. In [12], the case of tasks involving interactions with the environment is tackled. The concept of *corrective shared autonomy* is introduced, meaning that users provide corrections to critical robot state variables, such as positions, forces, and execution rate, starting from an autonomous task model. The authors show the feasibility and advantages of the approach, including reduced user effort and physical demands.

In this paper, we propose a practical approach to perform complex harvesting tasks in vineyards via human-manipulator interaction. Due to the inherent complexity of this task, caused by imperfect perception and lack of dexterity, the assistance of human operators is often necessary. Therefore, we envisage the case of a mobile manipulator robot that can adapt the level of autonomy depending on the situation. In detail, the robot might ask for different levels of human intervention in case of perception or goal-reaching failures. In this framework, the human may either teleoperate the robot by using wearable IMU sensors or ultimately physically grab the robot end-effector and move it to the right location.

II. Mathematical Background

In this section, we provide the basic theory about the kinematics of the robotic system taken into consideration and the Hierarchical Quadratic Programming (HQP) control framework, which is used as low-level controller for the shared control architecture proposed in this paper.

A. Robot Kinematics

Let us consider a robot made of a t-DoFs movable torso and a a-DoFs serial manipulator. The end-effector configuration is denoted by $\boldsymbol{\rho} = [\boldsymbol{p}^{\mathrm{T}} \ \boldsymbol{o}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^7$, where $\boldsymbol{p} \in \mathbb{R}^3$, is the positional part, while $\boldsymbol{o} \in \mathbb{R}^4$ represents the unit quaternion expressing the orientation. The vector expressing the configuration of the robot is defined as:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_T^{\mathrm{T}} & \boldsymbol{q}_A^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n \ , \tag{1}$$

where $\boldsymbol{q}_T \in \mathbb{R}^t$ gathers the joint variables of the torso, $\boldsymbol{q}_A \in \mathbb{R}^a$ are the vectors of arm joint variables and $n = t + a$.

The relationship between the end-effector generalized velocity and the joint velocity can be expressed as:

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \tag{2}$$

where $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ is the Jacobian matrix of the robot, while $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$ are the linear and angular velocities, respectively.

B. Hierarchical Quadratic Programming

The HQP formulation [13], [14] allows to seamlessly handle a task hierarchy. Each task that the robot has to perform is encoded into proper equality or inequality constraints, and strict priority among them is enforced by solving a cascade of Quadratic Programming (QP) problems in which the solution of a task with a given priority is obtained by considering the solutions of all higher-priority tasks as additional constraints. The outcome is that the solution of the lower-priority tasks does not affect the execution of the higher-priority ones.

Let us consider h arbitrary task functions $\{\boldsymbol{\sigma}_1 \in \mathbb{R}^{m_1}, \boldsymbol{\sigma}_2 \in \mathbb{R}^{m_2}, \cdots, \boldsymbol{\sigma}_h \in \mathbb{R}^{m_h}\}$ with corresponding Jacobian matrices $\{\boldsymbol{J}_1, \boldsymbol{J}_2, \cdots, \boldsymbol{J}_h\}$ and corresponding slack variables $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_h\}$. In detail, the input velocity that fulfills the i th task is the result of the following QP problem:

$$\begin{aligned} \min_{\boldsymbol{w}_i, \dot{\boldsymbol{q}}} \quad & \frac{1}{2}\boldsymbol{w}_i^T \boldsymbol{Q}_{w,i} \boldsymbol{w}_i \\ \text{s.t.} \quad & \underline{\boldsymbol{b}}_k \leq \boldsymbol{J}_k(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{w}_k^\star \leq \bar{\boldsymbol{b}}_k, \qquad \forall k \in 1, \ldots, i-1 \\ & \underline{\boldsymbol{b}}_i \leq \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{w}_i \leq \bar{\boldsymbol{b}}_i, \end{aligned} \tag{3}$$

where $\boldsymbol{Q}_{w,i} \in \mathbb{R}^{m_i \times m_i}$ and $\boldsymbol{w}_i \in \mathbb{R}^{m_i}$ are the weighting matrix and slack variables relative to the i th task, respectively, $\underline{\boldsymbol{b}}_{(\cdot)} \in \mathbb{R}^{m_k}$ and $\bar{\boldsymbol{b}}_{(\cdot)} \in \mathbb{R}^{m_{(\cdot)}}$ are the minimum and maximum desired task velocities and $\boldsymbol{w}_k^\star$ is the solution of the k QP problem. The slack variables are needed to relax the constraints in case they are conflicting with each other, and their minimization is aimed at minimizing the error in the execution of the task. If $\boldsymbol{w}_i = \boldsymbol{0}$ the constraint would be hard, and if $\underline{\boldsymbol{b}}_k = \bar{\boldsymbol{b}}_k = \boldsymbol{b}_k$ the constraint would be an equality one.

In this work, the first level of the hierarchy is devoted to the fulfillment of the joint position and velocity limits, which are handled by exploiting the Control Barrier Function (CBF) theory [15] as detailed in [16]. The second level of the hierarchy, instead, is switched among different kinds of constraints, as described in the following sections.

III. System and Problem Descriptions

We exploit a practical use case in the context of precision agriculture, borrowed from the European project CANOPIES[1], for presenting the shared control strategy devised in this paper; nevertheless, it is important to remark that it can be generalized to other contexts as well.

A. Setup description

The robotic system used to perform the experiments described in the following sections is shown in Figure 1 and has been specifically developed for the project CANOPIES. The robot is made up of a tracked mobile base and a bimanual robotic system. The tracked mobile base is an Alitrak DCT-350P equipped with a sensorial system and a Next Unit of Computing (NUC). The sensorial system consists of several navigation and perception sensors: two Lidars, an Inertial

[1] www.canopies-project.eu

Measurement Unit (IMU), and a Global Positioning System (GPS). The bimanual robotic system, instead, is a dual-arm PAL-Robotics system with a 2-DoFs torso, a 2-DoFs head and two 7-DoFs manipulator. The torso consists of a prismatic and a revolute joint, which allows changing its height and orientation, respectively, and is provided with a stereo microphone and a speaker. The head consists of a pan and a tilt joint and is equipped with a RealSense D435 RGB-D sensor. The manipulators consist of seven revolute joints, a Force/Torque (F/T) sensor, and a support for RGB-D sensors. Moreover, since the robot is designed to work in table-grape vineyards, the left arm is provided with a pruning tool. The right arm, instead, is provided with a harvesting tool capable of grasping and cutting the grapes' peduncle. It is worth noting that the proposed shared control architecture has been devised for a harvesting operation; thus, the system modeling in Section II-A considers the presence of only one manipulator.

B. Problem description

CANOPIES project tasks include harvesting grapes in a pergola system. To harvest grapes efficiently, the robot must first identify the grapes in its surrounding environment and accurately estimate the 3D location of the peduncle to be cut. Once an estimate of the peduncle position is obtained, to begin the harvesting process, a dedicated module generates the required end-effector position and orientation trajectories, connecting a series of appropriate waypoints that enable the robot to: i) reach a pre-grasp position at a predefined distance from the detected peduncle; ii) reach the grasp position on the peduncle; iii) close the gripper to hold the bunch; iv) close the scissors to cut the peduncle; v) place the bunch into a designated container. A module developed by a project's partner is used to estimate the position of the peduncle by using RGB-D cameras [17]. Due to potential obstructions from leaves or other bunches, it is often necessary to conduct the estimation procedure for grapes and peduncle position from different perspectives. To this purpose, the proposed approach involves leveraging the RGB-D sensors on the wrists. In more detail, the perception software first uses RGB-D data from the head-mounted sensor to provide an initial estimate of the peduncle position. If this is obstructed by leaves or other grapes, or if the grape is too far away to be accurately estimated by the head camera, one of the end effectors moves closer to the detected cluster. The aim was to retry and improve the accuracy of the position estimation using a wrist-mounted camera. If successful, the harvesting process can continue.

However, due to the inherent complexity of the task, imperfect perception, and lack of flexibility due to obstacles and robot limitations, the task may not be autonomously completed even by using the wrist cameras. Therefore, assistance from a human operator is often required, in which case we assume that the robot can explicitly request human intervention through its text-to-speech module.

IV. Proposed Solution

To achieve the task described in Section III-B, we propose a shared control strategy that leverages a finite state machine algorithm to change the level of autonomy of the robot depending on the performances of the perception system. In detail, three possible control modes are envisioned: *autonomous* mode, *teleoperation* mode, and *hand-guiding* mode.

In *autonomous* mode, the robot autonomously executes the necessary end-effector motions to perform the harvesting procedure, taking the reference end-effector trajectories from the Trajectory Generation module. In this operational mode, the employed control law is an admittance controller that exploits the readings of a wrench sensor mounted on the wrist of the manipulator to make it compliant with respect to external forces, aiming at avoiding damage to the robot and/or the environment. The *teleoperation* mode allows a human operator to teleoperate the end-effector by moving his/her arm. To this aim, the position of his/her hand is estimated from two IMU sensors attached along his/her arm by a dedicated module. Finally, in *hand-guiding* mode, the human operator guides the end-effector towards the point to cut by physically moving it. This operational mode is obtained by employing the same admittance controller used in *autonomous* mode but with different gains.

The Finite State Machine algorithm that switches among these operational modes is shown in Figure 2. The system starts in *autonomous* mode, first acquiring a rough estimate of the peduncle position from the head camera, then moving the end-effector closer to the bunch and acquiring a new image from the wrist-mounted camera. If the perception system manages to provide an accurate peduncle position, the rest of the harvesting operation is executed without changing the operational mode. In case of occlusions, instead, it is changed to *teleoperation*, and the robot makes use of the text-to-speech module to notify the change to human operator. At this point, the operator starts teleoperating the end-effector aiming at acquiring new images from different points of view. If the perception system still does not succeed in providing an accurate estimation after a predefined amount of attempts, the operational mode is changed to *hand-guiding*, and the operator physically drives the end-effector toward the point to cut. After a predefined amount of time in which the robot does not sense any external forces, the operational mode is switched back to *autonomous*, and the rest of the harvesting operation is carried out autonomously.

In the following, the main blocks needed to build such a strategy are detailed.

A. Admittance control

When in *autonomous* mode, the control objective included in the HQP formulation allows moving its end-effector while exhibiting a compliant behavior with respect to external forces that might occur for accidental contact between the robot and the environment.

Let us assume that the desired trajectory for the end-effector can be generated by a Trajectory Generation module as:

$$\boldsymbol{a}^d = \begin{bmatrix} \ddot{\boldsymbol{p}}^d \\ \boldsymbol{\alpha}^d \end{bmatrix} \quad \boldsymbol{v}^d = \begin{bmatrix} \dot{\boldsymbol{p}}^d \\ \boldsymbol{\omega}^d \end{bmatrix} \quad \boldsymbol{x}^d = \begin{bmatrix} \boldsymbol{p}^d \\ \boldsymbol{o}^d \end{bmatrix} , \tag{4}$$

where $\ddot{\boldsymbol{p}}^d, \dot{\boldsymbol{p}}^d$ and $\boldsymbol{p}^d$ express the desired linear acceleration, linear velocity and position and $\boldsymbol{\alpha}^d, \boldsymbol{\omega}^d$ and $\boldsymbol{o}^d$ express

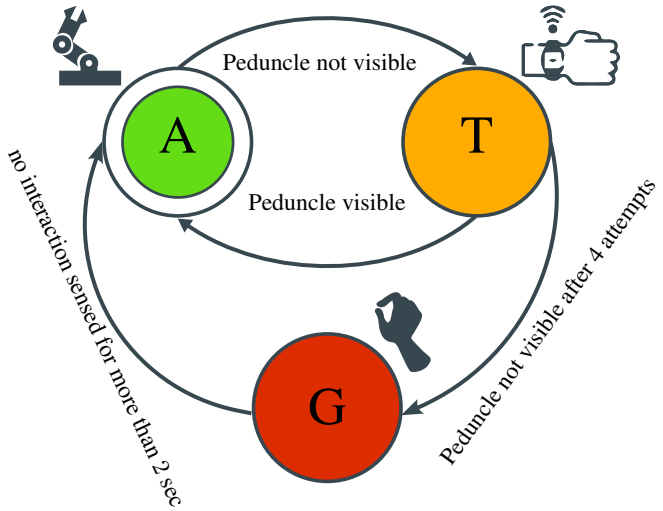

Fig. 2: Finite State Machine describing the shared control approach. The letters A, T, and G refer to the *autonomous*, *teleoperation*, and *hand-guiding* modes, respectively, i.e., to the three operating modes used to handle the robot autonomy.

the desired angular acceleration, angular velocity and quaternion, respectively. Assuming the presence of a wrench sensor mounted on the wrist of the manipulator, it is possible to define the vector:

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\mu} \end{bmatrix} \in \mathbb{R}^6 \ , \tag{5}$$

as the external measured wrench vector stacking the forces $\boldsymbol{f} \in \mathbb{R}^3$ and moments $\boldsymbol{\mu} \in \mathbb{R}^3$ for the end-effector. We want the system to exhibit the following virtual dynamics:

$$\boldsymbol{K}_M \, \tilde{\boldsymbol{a}} + \boldsymbol{K}_D \, \tilde{\boldsymbol{v}} + \boldsymbol{K}_P \, \tilde{\boldsymbol{x}} = \boldsymbol{h}, \tag{6}$$

where:

$$\tilde{\boldsymbol{a}} = \boldsymbol{a}^d - \boldsymbol{a} \quad \tilde{\boldsymbol{v}} = \boldsymbol{v}^d - \boldsymbol{v} \quad \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{p}^d - \boldsymbol{p} \\ \boldsymbol{o}^d * \boldsymbol{o}^{-1} \end{bmatrix} \tag{7}$$

are the acceleration, velocity and configuration errors, respectively, and $\boldsymbol{K}_M \in \mathbb{R}^{6\times6}$, $\boldsymbol{K}_D \in \mathbb{R}^{6\times6}$ and $\boldsymbol{K}_P \in \mathbb{R}^{6\times6}$ are the desired virtual mass, damping and stiffness, respectively. The acceleration of the end-effector can be approximated by numerically deriving its velocity assuming a sampling time T_s, as:

$$\boldsymbol{a}(t) = \frac{\boldsymbol{v}(t) - \boldsymbol{v}(t - T_s)}{T_s} \ , \tag{8}$$

where $\boldsymbol{v}(t)$ is the current velocity and $\boldsymbol{v}(t - T_s)$ is the velocity at the previous time step. Substituting this expression in Eq. (6):

$$\boldsymbol{K}_M \boldsymbol{a}^d - \boldsymbol{K}_M \frac{\boldsymbol{v}}{T_s} + \boldsymbol{K}_M \frac{\boldsymbol{v}(t - T_s)}{T_s} + \boldsymbol{K}_D \boldsymbol{v} + \boldsymbol{K}_P \tilde{\boldsymbol{x}} = \boldsymbol{h} \ , \tag{9}$$

which can be rewritten as:

$$\left(-\frac{\boldsymbol{K}_M}{T_s} - \boldsymbol{K}_D \right) \boldsymbol{v} = \boldsymbol{h} - \boldsymbol{\gamma} \ , \tag{10}$$

where :

$$\boldsymbol{\gamma} = \boldsymbol{K}_M \boldsymbol{a}^d - \frac{\boldsymbol{K}_M}{T_s} \boldsymbol{v}(t - T_s) - \boldsymbol{K}_D \boldsymbol{v}^d - \boldsymbol{K}_P \tilde{\boldsymbol{x}} \ . \tag{11}$$

By folding Eq. (2) in Eq. (10), one finally obtains:

$$\underbrace{\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D \right) \boldsymbol{J}}_{\boldsymbol{J}_{\text{adm}}} \dot{\boldsymbol{q}} = \underbrace{-\boldsymbol{\gamma} - \boldsymbol{h}}_{\boldsymbol{b}_{\text{adm}}} \ , \tag{12}$$

which is the expression of the equality constraint to use in the HQP formulation in Eq. (3).

B. Human-arm IMU-based Trajectory Estimation

When in *teleoperation* mode, the human operator drives the end-effector by moving his/her arm. In this section, we describe how the human arm position is estimated by using off-the-shelf IMU devices and by leveraging a similar approach as in [18]. In detail, let us assume that two 9-axes IMUs are mounted, for example, on the right arm of the robot as in Fig 3; that is, IMU_1 is mounted on the arm, IMU_2 is mounted on the forearm, while Σ_3 is a frame on the wrist.

A Kalman filter, to estimate the orientation of the IMUs sensor reference frames with respect to the a global reference frame from gyroscope, accelerometer, and magnetometer signals is assumed to be available. This is not a strong assumption since many off-the-shelf platforms already come with a built-in filter. However, existing approaches, like the one in [19], can be utilized for this purpose. At this point, let Σ_s be the shoulder reference frame and let be Σ_i $(i = 1, 2, 3)$ the IMUs reference frame as depicted in Fig. 3. The aforementioned filter allows to estimate matrix $\boldsymbol{R}_i(t)$ $(i = 1, 2, 3)$ expressing the orientation of frame Σ_i over time. Furthermore, let l_1, l_2 the lengths of the arm, and forearm, respectively. The objective is to continuously estimate the position of frame Σ_3 with respect to Σ_0. Since it is not foreseen to use an additional sensor positioned on the shoulder and acting as base reference frame, we assume that the arm estimation procedure begins at time t_0 and, at this moment, the arm configuration is as depicted in Figure 3 (left), i.e., with the arm fully extended and pointing downward. The following rotation matrices are defined: $\bar{\boldsymbol{R}}_i = \boldsymbol{R}_i^{\mathrm{T}}(t_0)\boldsymbol{R}_i(t)$ $(i = 1, 2)$, that are such as $\bar{\boldsymbol{R}}_i(t_0) = \boldsymbol{I}$. Finally, we can derive the position of frame three (that is of the wrist) as:

$$\boldsymbol{p}_w = \bar{\boldsymbol{R}}_1 \boldsymbol{r}_1 + \bar{\boldsymbol{R}}_2 \boldsymbol{r}_2 \tag{13}$$

where $r_i = l_i \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$ $(i = 1, 2)$.

C. Hand-Guiding control

When the system is in *hand-guiding* mode, the operator can freely move the end-effector by physically interacting with it. This behavior can be easily obtained from the admittance controller used in *autonomous* mode by setting $\boldsymbol{K}_p = \boldsymbol{O}_{6\times6}$ and $\boldsymbol{a}^d = \boldsymbol{v}^d = \boldsymbol{0}$ in Eq. (12), thus leading to the following form of the constraint in the HQP formulation:

$$\underbrace{\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D \right) \boldsymbol{J}}_{\boldsymbol{J}_{\text{hg}}} \dot{\boldsymbol{q}} = \underbrace{-\boldsymbol{\gamma} - \boldsymbol{h}}_{\boldsymbol{b}_{\text{hg}}} \ , \tag{14}$$

where:

$$\boldsymbol{\gamma} = -\frac{\boldsymbol{K}_M}{T_s} \boldsymbol{v}(t - T_s) \ . \tag{15}$$

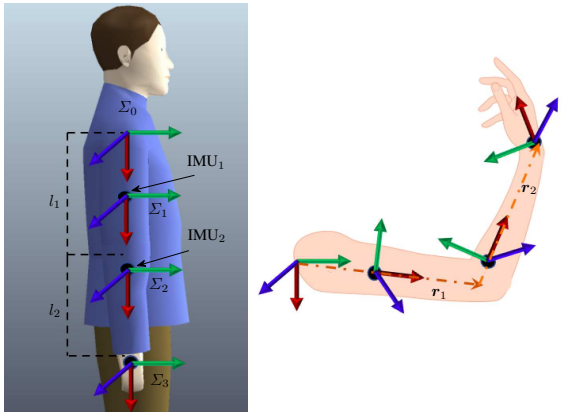

Fig. 3: IMU sensors mounted on the human arm. Σ_0 is the shoulder reference frame, Σ_1 is the upper arm reference frame relative to IMU$_1$, Σ_2 is the forearm reference frame relative to IMU$_2$, Σ_3 is the wrist reference frame. l_1 and l_2 are the lengths of upper-arm and forearm respectively.

V. Experimental Validation

In this section, we go through some experiments performed to validate the proposed architecture. The experiments are conducted in an indoor and controlled environment using fake bunches and fake leaves to simulate the occlusions. In particular, in this section, we examine two harvesting operations, each one with a diverse evolution of the FSM shown in Figure 2. In the first one, the operator teleoperates the end-effector of the robot to change the field of view of the wrist-mounted camera until the perception system succeeds in accurately estimating the peduncle to cut. In this regard, the operator, through the approach described in Section IV-B, directly changes the robot end-effector position; the orientation, instead, is controlled by the robot, which modifies it in such a way as to keep the grape at the center of the image view. In the second experiment, instead, the system switches to *hand-guiding* mode, and the operator physically moves the end-effector by leveraging the approach in Section IV-A. The two experiments are both shown in the video at the following link[2].

With regards to parameter values, l_1 and l_2 in Section IV-B are set to 0.28m and 0.23m, respectively. While, it is in relation to admittance parameters in Section IV-A, $T_s = 0.01$s,

$$\boldsymbol{K}_M = \text{diag}\{\begin{bmatrix}20 & 20 & 20 & 3 & 3 & 3\end{bmatrix}\}$$
$$\boldsymbol{K}_D = \text{diag}\{\begin{bmatrix}253 & 253 & 253 & 27 & 27 & 27\end{bmatrix}\}$$

and $\boldsymbol{K}_P = \boldsymbol{O}$. Finally, $\boldsymbol{Q}_u$ and $\boldsymbol{Q}_{w,k}$ are set to the identity in Section II-B

In the first experiment, the robot is in *autonomous* mode until the perception software does not provide it with the cut position estimation of one of the bunches in its surroundings (Figure 4. Top Left). To validate the proposed strategy, we considered a scenario with a single grape bunch whose peduncle is occluded. In this kind of situation, the perception system can detect and localize the bunch, but not the exact position of the peduncle to cut. For this reason, the robot brings its harvesting tool close to the estimated bunch position and switches

[2]https://youtu.be/iKkRoL3vqKA

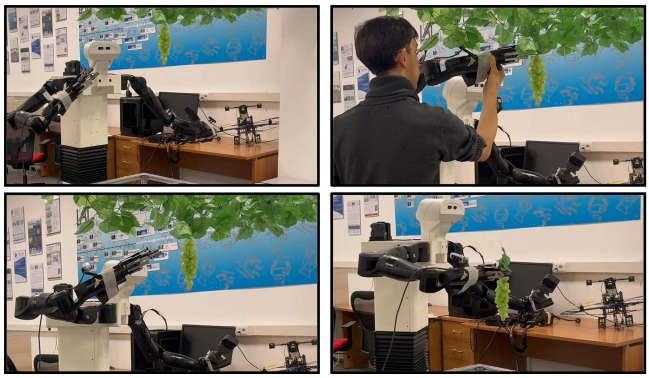

Fig. 4: Top left: agricultural robot in a generic configuration. Top right: harvesting tool in the pre-grasping position with grape partially occluded by leaves. Bottom left: human assisting the robot in the harvesting procedure. Bottom right: harvesting tool in the pre-release position with the harvested grape.

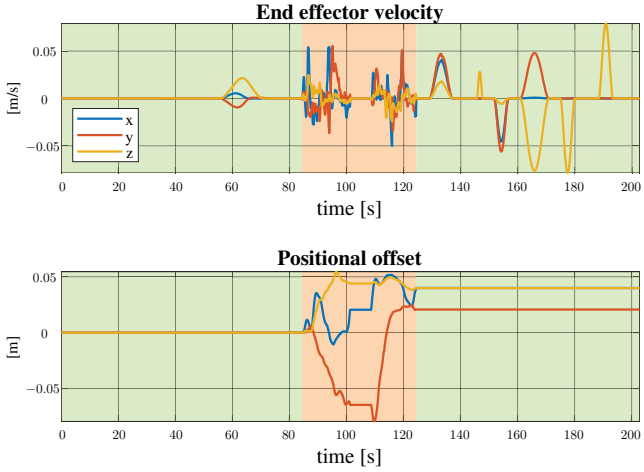

Fig. 5: First experiment. (Top) Linear velocity of the end effector. (Bottom) Positional offset generated by the human operator through the IMU sensors attached along his/her arm. The background color of the plots encodes the active operational mode. In particular, the green regions are related to *autonomous* mode, and the orange ones to *teleoperation* mode.

its control mode from *autonomous* to *teleoperation*. As stated in Section IV, the *teleoperation* mode allows any human operator equipped with suitable wearable sensors, such as the IMU sensors described in Section IV-B, to adjust the end-effector position. However, in our case study, it also allows the human operator to prompt new estimations. In detail, by pressing a button placed on one of the IMUs, he/she can prompt the perception software to perform a new attempt using the data collected by the wrist RGB-D sensor. In this experiment, the operator moves the end-effector and prompts a new estimation attempt twice times. At this point, the perception system succeeds in getting an accurate peduncle position estimation. Hence, the robot switches back to the *autonomous* mode and performs the remaining steps of the harvesting procedure on its own.

As specified in Figure 2, the number of queries that the human can send to the perception software is limited. In the second experiment, the number of queries sent to the software reaches the maximum threshold, which we have set equal to four, and the robot switches its operational mode to *hand-*

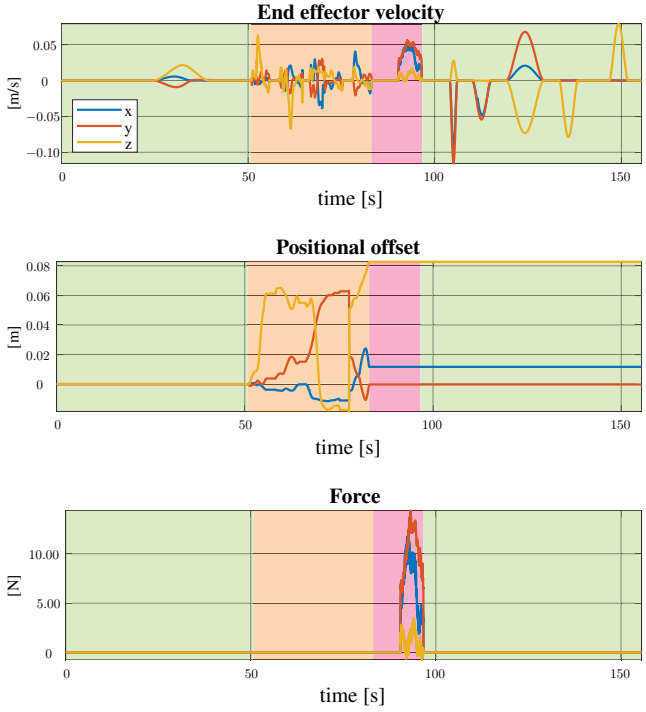

Fig. 6: Second experiment. (Top) Linear velocity of the end effector. (Middle) Positional offset generated by the human operator through the IMU sensors attached along his/her arm. (Bottom) Wrench detected by the wrist F/T sensor. The background color of the plots encodes the active operational mode. In particular, the green regions are related to *autonomous* mode, the orange ones to *teleoperation* mode, and the red ones to the *hand-guiding* mode.

guiding. As stated in Section IV, the *hand-guiding* mode allows the human operator to manually move the end-effector and take it to a desired pose, such as a feasible cutting point. According to Figure 2, the robot remains in the *hand-guiding* mode as long as it senses external forces and/or torques with its wrench sensor. When the robot does not sense any forces and/or torques attributable to interactions for more than 2 seconds, it switches back to the *autonomous* mode and resumes the harvesting procedure starting from the operation of closure of the gripper.

Figures 5) - 6) show the results obtained from the two experiments. In particular, proceeding from the top to the bottom, we have the time evolutions of the linear velocity of the end effector, the positional offset generated by the human operator through the IMU sensors attached along his/her arm, and the force detected by the wrist F/T sensor throughout the experiments. The background color of the plots encodes the operational mode. In particular, we highlighted in green the regions related to the *autonomous* mode, in orange the regions associated with the *teleoperation* mode, and in red the regions related to the *hand-guiding* mode.

VI. Conclusions

In this paper, we proposed a human-robot shared control strategy aimed at allowing a human operator to assist a robotic system in accomplishing the targets envisaged by the given task while preserving the autonomy of the robot as much as possible. We based the control strategy on three operating modes, each one capable of ensuring a different degree of autonomy to the robot, and implemented an FSM for managing the switch from one operating mode to another (see Figure 2). Finally, we validated the effectiveness of the proposed strategy in an indoor environment through a dual-arm robot. As future development, we aim both to validate the effectiveness of the strategy in outdoor environments and to make the human operator capable of handling both the orientation of the end effector, in addition to its position, and the swivel angle of the arm.

References

[1] P. Di Lillo, D. Di Vito, and G. Antonelli, "Merging global and local planners: real-time replanning algorithm of redundant robots within a task-priority framework," *IEEE Transactions on Automation Science and Engineering*, 2022.

[2] B. Lacevic, P. Rocco, and A. M. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1257–1270, 2013.

[3] M. Lippi and A. Marino, "Human multi-robot safe interaction: A trajectory scaling approach based on safety assessment," *IEEE Trans. Control Syst. Technol.*, pp. 1–16, 2020.

[4] B. Xi, S. Wang, X. Ye, Y. Cai, T. Lu, and R. Wang, "A robotic shared control teleoperation method based on learning from demonstrations," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, 2019.

[5] D. Losey, C. McDonald, E. Battaglia, and M. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, 01 2018.

[6] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Continuous role adaptation for human–robot shared control," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 672–681, June 2015.

[7] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *Int. J. Robot. Res.*, vol. 31, no. 13, pp. 1656–1674, 2012.

[8] J. R. Medina, T. Lorenz, and S. Hirche, "Synthesizing anticipatory haptic assistance considering human behavior uncertainty," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 180–190, Feb 2015.

[9] D. P. Losey and M. K. O'Malley, "Trajectory deformations from physical human–robot interaction," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 126–138, Feb 2018.

[10] V. Izadi, A. Bhardwaj, and A. H. Ghasemi, "Impedance modulation for negotiating control authority in a haptic shared control paradigm," in *2020 American Control Conference (ACC)*, 2020, pp. 2478–2483.

[11] R. Balachandran, H. Mishra, M. Cappelli, B. Weber, C. Secchi, C. Ott, and A. Albu-Schaeffer, "Adaptive authority allocation in shared control of robots using bayesian filters," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11 298–11 304.

[12] M. Hagenow, E. Senft, R. Radwin, M. Gleicher, B. Mutlu, and M. Zinn, "Corrective shared autonomy for addressing task variability," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3720–3727, 2021.

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

[14] S. Kim, K. Jang, S. Park, Y. Lee, S. Y. Lee, and J. Park, "Continuous task transition approach for robot controller based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1603–1610, 2019.

[15] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Confer. Decis. Control*, 2014, pp. 6271–6278.

[16] J. Palmieri, P. Di Lillo, M. Lippi, S. Chiaverini, and A. Marino, "A control architecture for safe trajectory generation in human–robot collaborative settings," *IEEE Transactions on Automation Science and Engineering*, pp. 1–0, 2024.

[17] G. Coll-Ribes, I. J. Torres-Rodríguez, A. Grau, E. Guerra, and A. Sanfeliu, "Accurate detection and depth estimation of table grapes and peduncles for robot harvesting, combining monocular depth estimation and cnn methods," *Computers and Electronics in Agriculture*, vol. 215, December 2023, 108362.

[18] I. Prayudi and D. Kim, "Design and implementation of imu-based human arm motion capture system," in *2012 IEEE International Conference on Mechatronics and Automation*, 2012, pp. 670–675.

[19] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.