2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Perception-Driven Shared Control Architecture for Agricultural Robots Performing Harvesting Tasks

Jozsef Palmieri[1], Paolo Di Lillo[1], Alberto Sanfeliu[2], Alessandro Marino[1]

***Abstract*— This paper introduces a shared control framework designed specifically for agricultural mobile manipulators engaged in harvesting operations. The shared control strategy allows for achieving such operations by dynamically exchanging the control between the robotic system and a human operator depending on the uncertainty in the environment perception. For this purpose, the robot's behavior is dynamically adapted to switch between two control modes with a different level of autonomy of the robot. The level of autonomy is encoded in two different admittance behaviors which are included in a first-order Hierarchical Quadratic Programming (HQP) control framework, that allows the robot to simultaneously address other control objectives at the same time. Experimental results with a dual-arm mobile robot, developed as part of the EU-funded CANOPIES project, demonstrate the effectiveness of the proposed method in real conditions.**

I. Introduction

Modern agriculture faces the challenge of meeting increasing global food demands while minimizing resource consumption and environmental impact. In response, precision agriculture has emerged as a transformative approach, integrating robotics and automation into farming practices. Nonetheless, for such kind of systems achieving complete autonomy is often infeasible or economically inefficient for the majority of tasks in real-world scenarios, not confined solely to agricultural contexts. Therefore, in recent years there has been a growing trend of integrating human-robot collaboration in various industries and more domestic environments. This is primarily driven by the recognition that humans and robots possess unique and complementary capabilities that can be combined to improve performance in tasks execution and maximize productivity. The collaboration can take different forms, such as *i*) pure sharing a workspace where humans and robots work together on different tasks or *ii*) engaging in collaborative tasks where humans provide their cognitive skills also intentionally exchanging 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].

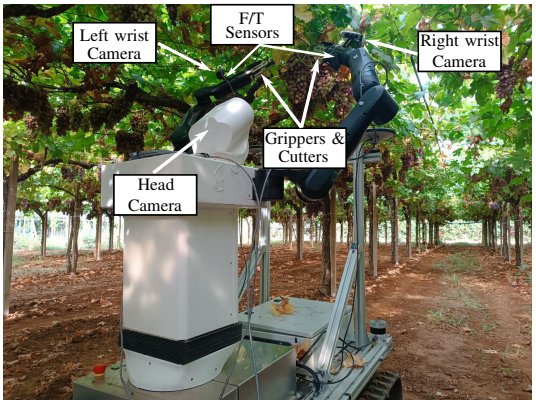

Fig. 1. Farming robot designed for the EU-funded project CANOPIES. In particular, it is highlighted: the head and the wrist cameras, the end-effectors equipped with gripper and cutter, and Force/Torque (F/T) sensors.

More interesting is the second case, where the synergy between humans and robots really comes into play. 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.

In this paper, we focus on shared control in the context of physical human-robot interaction. Several approaches have been proposed in the last decades [5] in this context; 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. 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. Furthermore, in [8], a solution based on data-driven

[1]J. Palmieri, P. Di Lillo and A. Marino are with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Via G. Di Biasio 43, Cassino, Italy, {jozsef.palmieri, pa.dilillo, al.marino}@unicas.it.

[2]Alberto Sanfeliu is with the Institut de Robòtica i Informatica Industrial (CSIC-UPC), Universitat Politècnica de Catalunya, Llores Artigas 4-6, Barcelona, Spain, {sanfeliu}@iri.upc.edu.

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, partially, by 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.

stochastic modeling is devised. It addresses uncertainties in the human behavior model through the formulation of a risk-sensitive optimization problem. More recently, in [9] proposes achieving a similar behavior by adjusting the robot's trajectory in response to the exerted human forces. 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. A different approach to a similar problem is presented in [11]. In detail, the allocation of control is determined using a metric derived from a Bayesian filter, which continually adapts based on on-line sensor measurements. 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 starting from an autonomous task model. In the framework of shared control and physical interaction, the work in [13] addresses variable admittance considering the adaptation to human intentions and the maintenance of system passivity, to ensure a safe and intuitive human-robot interaction. Passivity in mixed-initiative shared control is tackled also in [14], in which the concept of adaptive mixed-initiative shared control is redefined as an adaptive stiffness control strategy that preserves passivity when scaling forces.

In this paper, in the framework of the European project CANOPIES[1] 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 bi-manual mobile robot equipped with advanced perception capabilities and F/T sensors (Figure 1) whose level of autonomy is adjusted according to the surrounding scenario, the output of the perception software used to detect the bunches and estimate their cut points, and the progress status of the given task. In detail, the robot might ask for human intervention in case of perception or goal-reaching failures. Such perception failures can happen when the vineyard grapes (bunches) overlap with other grapes or the grape peduncles are rolled up a vineyard cane, which makes it very difficult to find the cutting point.

In this framework, the level of autonomy is embedded in a variable admittance controller, whose gains are adapted to let the robot or human lead depending on the performances of the perception system. Noticeably, the admittance behavior is seamlessly integrated within the Hierarchical Quadratic Programming control framework [15], which allows the robot to fulfill multiple tasks simultaneously. More in detail, joint constraints such as their position and velocity limits are handled by resorting to the Control Barrier Functions (CBFs) theory [16], while the secondary task in the HQP formulation is switched between two possible admittance behaviors to encode different levels of robot autonomy.

[1]www.canopies-project.eu

II. Implemented control architecture

Figure 2 shows the developed control architecture where the main blocks of the devised approach are depicted. In detail, *i*) the *Perception algorithm* block is responsible for the environment perception and grape position estimation; *ii*) the *Wrench Sensors* one for estimating the interaction wrench between the end-effector and the environment; *iii*) the *Trajectory generation* module allows to plan desired end-effectors' trajectories to both acquire information about the environment and perform harvesting tasks; while *iv*) the *HQP controller* block implements the control algorithm. In the

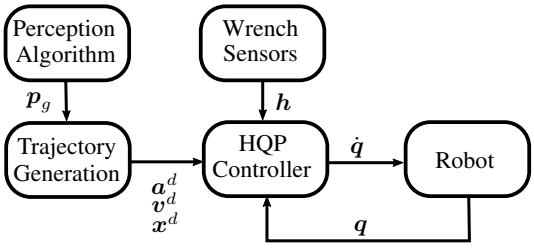

Fig. 2. Overall control architecture showing the different components and flows of information.

following, we give details about: i) the kinematic modeling of the considered robotic system; ii) the HQP controller that is used as a low-level motion controller, detailing all the considered tasks for implementing the shared control strategy; iii) the perception algorithm which is responsible for the recognition of the grapes and the estimation of the 3D position of the peduncle to cut.

A. System modeling

Let us consider a mobile robot equipped with a movable torso and a dual arm system, in which each end-effector configuration is denoted by $\boldsymbol{x}_y = [\boldsymbol{p}_y^{\mathrm{T}}\ \boldsymbol{o}_y^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^7$, ($y = L, R$, Left and Right, respectively) where $\boldsymbol{p}_y \in \mathbb{R}^3$ is the position part, and $\boldsymbol{o}_y \in \mathbb{R}^4$ represents the unit quaternion expressing the orientation. We denote by $\boldsymbol{x} = [\boldsymbol{x}_L^{\mathrm{T}}\ \boldsymbol{x}_R^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{14}$ the vector collecting all end-effector configurations. Moreover, let the joint vector $\boldsymbol{q} \in \mathbb{R}^n$ defined as:

$$\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_B^{\mathrm{T}} & \boldsymbol{q}_T^{\mathrm{T}} & \boldsymbol{q}_L^{\mathrm{T}} & \boldsymbol{q}_R^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},$$

where $\boldsymbol{q}_B = \begin{bmatrix}\boldsymbol{p}_B^{\mathrm{T}},\ \boldsymbol{o}_B^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^b$ is the vector describing the mobile base configuration in terms of position and orientation expressed as quaternion, $\boldsymbol{q}_T \in \mathbb{R}^t$ gathers the joint variables of the torso, while $\boldsymbol{q}_L, \boldsymbol{q}_R \in \mathbb{R}^p$ are the vectors of joint variables relative to arms, and $n = b + t + 2p$ is the total number of degrees of freedom (DoFs). The collective vector of end-effector linear/angular velocities is described by $\boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_L^{\mathrm{T}} & \boldsymbol{v}_R^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{12}$ and its relationship with the system velocity vector can be expressed as:

$$\boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_L \\ \boldsymbol{v}_R\end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{u}\ , \tag{1}$$

where we assumed the following first order model $\dot{\boldsymbol{q}} = \boldsymbol{u}$ for the robot, being $\boldsymbol{u} \in \mathbb{R}^{12}$ the control input, and with the

Jacobian $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{12\times n}$ partitioned as follows:

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{B,L}(\boldsymbol{q}_B) & \boldsymbol{J}_{T,L}(\boldsymbol{q}_T) & \boldsymbol{J}_L(\boldsymbol{q}_L) & \boldsymbol{O}_{6\times a} \\ \boldsymbol{J}_{B,R}(\boldsymbol{q}_B) & \boldsymbol{J}_{T,R}(\boldsymbol{q}_T) & \boldsymbol{O}_{6\times a} & \boldsymbol{J}_R(\boldsymbol{q}_R) \end{bmatrix}, \tag{2}$$

where $\boldsymbol{O}_{x\times y}$ represents the $x\times y$ null matrix. The structure of Eq. (2) highlights that the joint velocities of the common base and torso contribute to the velocities of both end-effectors, while $\dot{\boldsymbol{q}}_L$ ($\dot{\boldsymbol{q}}_R$) only contributes to the velocity of the left (right) end-effector.

Finally, we assume the robot is subject to the following joint-space kinematic constraints:

$$\begin{cases} \underline{\boldsymbol{q}} \le \boldsymbol{q} \le \bar{\boldsymbol{q}} \\ \underline{\dot{\boldsymbol{q}}} \le \dot{\boldsymbol{q}} \le \bar{\dot{\boldsymbol{q}}}, \end{cases} \tag{3}$$

where $\underline{\boldsymbol{q}}$ ($\underline{\dot{\boldsymbol{q}}}$) and $\bar{\boldsymbol{q}}$ ($\bar{\dot{\boldsymbol{q}}}$) are the minimum and maximum joint configuration (velocity) values, respectively.

B. HQP controller and considered tasks

In order to safely operate in a dynamic and unstructured environment, a robot has to be capable of respecting several constraints, e.g., joint position/velocity limits in Eq. (3) , and of performing several tasks simultaneously.

A popular control framework that allows assigning a strict priority order to multiple tasks and computing a solution that fulfills the hierarchy is the Hierarchical Quadratic Programming (HQP) [15], [17]. In this formulation, the task hierarchy is implemented 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. In this way, the solution of the lower-priority tasks does not affect the execution of the higher-priority ones.

More in detail, let us consider a hierarchy composed of h arbitrary tasks $\{\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_h\}$. The solution that fulfills the task hierarchy is computed by solving the cascade of h QP problems, leading to a sequence of optimal slack variables $\{\boldsymbol{w}_1^\star, \boldsymbol{w}_2^\star, \cdots, \boldsymbol{w}_h^\star\}$ that is minimal with respect to a lexicographic order. In detail, the i-th QP problem has the following structure:

$$\begin{aligned} \min_{\boldsymbol{w}_i, \dot{\boldsymbol{q}}} \quad & \frac{1}{2}\boldsymbol{w}_i^T \boldsymbol{Q}_{w,i} \boldsymbol{w}_i \\ \text{s.t.} \quad & \underline{\boldsymbol{b}}_k \le \boldsymbol{J}_k(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{w}_k^\star \le \bar{\boldsymbol{b}}_k, \qquad \forall k \in 1, \ldots, i-1 \\ & \underline{\boldsymbol{b}}_i \le \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{w}_i \le \bar{\boldsymbol{b}}_i, \end{aligned} \tag{4}$$

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 ith task, respectively, while $\underline{\boldsymbol{b}}_i \in \mathbb{R}^{m_i}$ and $\bar{\boldsymbol{b}}_i \in \mathbb{R}^{m_i}$ are the minimum and maximum desired task velocities. The slack variables are needed to relax the constraints in case they are conflicting with those included by the higher-priority tasks, or in case of unfeasibility of the primary task, and their minimization is aimed at minimizing the error in the execution of the task.

1) Primary task. Joint limits tasks: The primary task that we consider in our hierarchy allows the robot to respect the joint constraints in Eq. (3). They can be properly included in the HQP formulation by exploiting the Control Barrier Functions (CBFs) framework, which will be briefly recalled here. Let us consider a system with dynamics:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(t, \boldsymbol{\xi}) + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u}, \tag{5}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are Lipschitz-continuous vector fields, $\boldsymbol{\xi} \in \mathcal{D} \subset \mathbb{R}^l$ and $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^q$ are state and input of the system, respectively. Let the kth generic constraint be expressed in the general form $h_k(\boldsymbol{\xi}) \ge 0$, where $h_k(\cdot)$ is a continuous differentiable function in the domain $\mathcal{D}$. According to the CBF framework, let $\mathcal{C}_k \subset \mathcal{D}$ be defined as:

$$\begin{aligned} \mathcal{C}_k &= \{\boldsymbol{\xi} \in \mathbb{R}^l : h_k(\boldsymbol{\xi}) \ge 0\}, \\ \partial\mathcal{C}_k &= \{\boldsymbol{\xi} \in \mathbb{R}^l : h_k(\boldsymbol{\xi}) = 0\}, \\ \text{Int}(\mathcal{C}_k) &= \{\boldsymbol{\xi} \in \mathbb{R}^l : h_k(\boldsymbol{\xi}) > 0\}, \end{aligned} \tag{6}$$

implying that the state $\boldsymbol{\xi}$ is required to belong to the set $\mathcal{C}_k$ in order to satisfy constraint k. Function h_k is a CBF if an extended class $\mathcal{K}_\infty$ function α_k exists such that, for a dynamic system represented as in Eq. (5), it holds

$$\sup_{\boldsymbol{u}\in\mathcal{U}} [L_f h_k(\boldsymbol{\xi}) + L_g h_k(\boldsymbol{\xi})\boldsymbol{u}] \ge -\phi_k \alpha_k(h_k(\boldsymbol{\xi})), \tag{7}$$

where $\phi_k > 0$, and $L_f h_k$ and $L_g h_k$ are the Lie derivatives of function h_k with respect to f and g, respectively. Then, the following theorem holds [18]:

Theorem 1: Let function $h_k : \mathcal{D} \subset \mathbb{R}^l \to \mathbb{R}$ be a continuously differentiable function and the corresponding set $\mathcal{C}_k$ defined as in Eq. (6). If h_k is a CBF on $\mathcal{D}$ and $\frac{\partial h_k}{\partial \xi}(\xi) \ne 0 \ \forall \xi \in \partial\mathcal{C}_k$, then any Lipschitz continuous controller $u(\xi)$ for system in Eq. (5) satisfying Eq. (7) renders the set $\mathcal{C}_k$ asymptotically stable.

Since Eq. (7) is affine in the control input $\boldsymbol{u}$, the latter can be computed as the result of a convex optimization problem subject to constraint:

$$\begin{aligned} \boldsymbol{u}^\star = \arg\min_{\boldsymbol{u}} \ & \frac{1}{2}\left(\boldsymbol{u} - \boldsymbol{u}_{(\cdot)}\right)^{\mathrm{T}} \boldsymbol{Q}\left(\boldsymbol{u} - \boldsymbol{u}_{(\cdot)}\right) \\ \text{s.t.} \quad & \sup_{\boldsymbol{u}\in\mathcal{U}} [L_f h_k(\boldsymbol{\xi}) + L_g h_k(\boldsymbol{\xi})\boldsymbol{u}] \ge -\phi_k h_k(\boldsymbol{\xi}), \quad \forall\, k \end{aligned} \tag{8}$$

where $\boldsymbol{u}_{(\cdot)}$ is any nominal input for the system and $\boldsymbol{Q} \in \mathbb{R}^{q\times q}$ is a positive definite weight matrix. Now, in order to exploit this framework for handling the joint constraints, let us introduce the following functions:

$$\begin{cases} \underline{h}_i(q_i) = q_i - \underline{q}_i \ge \ 0, & i = 1, \cdots, n \\ \bar{h}_i(q_i) = \bar{q}_i - q_i \ge \ 0, & i = 1, \cdots, n \end{cases} \tag{9}$$

that lead to

$$\begin{cases} \dot{q}_i \ge -\underline{\phi}_i \underline{h}_i(q_i) & i = 1, \cdots, n \\ -\dot{q}_i \ge -\bar{\phi}_i \bar{h}_i(q_i) & i = 1, \cdots, n. \end{cases} \tag{10}$$

with $\bar{\phi}_i, \underline{\phi}_i > 0$. Let us define vector functions $\underline{\boldsymbol{h}} = [\underline{h}_1, \underline{h}_2 \cdots, \underline{h}_n]$ and $\bar{\boldsymbol{h}} = [\bar{h}_1, \bar{h}_2 \cdots, \bar{h}_n]$, and matrices $\underline{\boldsymbol{\Phi}} = \text{diag}\{\underline{\phi}_1, \underline{\phi}_2, \cdots, \underline{\phi}_n\}$ and $\bar{\boldsymbol{\Phi}} = \text{diag}\{\bar{\phi}_1, \bar{\phi}_2, \cdots, \bar{\phi}_n\}$; then, it is straightforward to verify that Eq. (10) can be more conveniently expressed as

$$\underline{\boldsymbol{\Phi}}\underline{\boldsymbol{h}}(\boldsymbol{q}) \le \dot{\boldsymbol{q}} \le \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{h}}(\boldsymbol{q}). \tag{11}$$

By combining Eq. (11) with the second of constraints in Eq. (3), it holds

$$\underline{\boldsymbol{b}}_c = \begin{bmatrix} -\underline{\boldsymbol{\Phi}}\underline{\boldsymbol{h}}(\boldsymbol{q}) \\ \dot{\underline{\boldsymbol{q}}} \end{bmatrix} \leq \boldsymbol{J}_{\text{jl}}\dot{\boldsymbol{q}} \leq \begin{bmatrix} \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{h}}(\boldsymbol{q}) \\ \dot{\bar{\boldsymbol{q}}} \end{bmatrix} = \bar{\boldsymbol{b}}_c, \tag{12}$$

where $\boldsymbol{J}_{\text{jl}} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{I}_n \end{bmatrix}^{\text{T}}$ is the constraints Jacobian.

2) Secondary Task. Admittance Task: A secondary task that is included in the task hierarchy allows the robot to follow a desired trajectory with its end-effectors 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-effectors can be generated by a Trajectory Generation module as:

$$\boldsymbol{a}^d = \begin{bmatrix} \ddot{\boldsymbol{p}}_L^d \\ \boldsymbol{\alpha}_L^d \\ \ddot{\boldsymbol{p}}_R^d \\ \boldsymbol{\alpha}_R^d \end{bmatrix} \quad \boldsymbol{v}^d = \begin{bmatrix} \dot{\boldsymbol{p}}_L^d \\ \boldsymbol{\omega}_L^d \\ \dot{\boldsymbol{p}}_R^d \\ \boldsymbol{\omega}_R^d \end{bmatrix} \quad \boldsymbol{x}^d = \begin{bmatrix} \boldsymbol{p}_L^d \\ \boldsymbol{o}_L^d \\ \boldsymbol{p}_L^d \\ \boldsymbol{o}_L^d \end{bmatrix} , \tag{13}$$

where $\ddot{\boldsymbol{p}}_{(\cdot)}^d, \dot{\boldsymbol{p}}_{(\cdot)}^d$ and $\boldsymbol{p}_{(\cdot)}^d$ express the desired linear acceleration, linear velocity and position and $\boldsymbol{\alpha}_{(\cdot)}^d, \boldsymbol{\omega}_{(\cdot)}^d$ and $\boldsymbol{o}_{(\cdot)}^d$ express the desired angular acceleration, angular velocity and quaternion, respectively. Assuming the presence of a wrench sensor mounted on the wrist of the manipulators, it is possible to define the vector:

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f}_L \\ \boldsymbol{\mu}_L \\ \boldsymbol{f}_R \\ \boldsymbol{\mu}_R \end{bmatrix} \tag{14}$$

as the external measured wrench vector stacking the forces $\boldsymbol{f}_{(\cdot)}$ and moments $\boldsymbol{\mu}_{(\cdot)}$ for both the end-effectors. We want the system to exhibit the following dynamics:

$$\boldsymbol{K}_M\, \tilde{\boldsymbol{a}} + \boldsymbol{K}_D\, \tilde{\boldsymbol{v}} + \boldsymbol{K}_P\, \tilde{\boldsymbol{x}} = \boldsymbol{h}, \tag{15}$$

where:

$$\tilde{\boldsymbol{a}} = \boldsymbol{a}^d - \boldsymbol{a} \quad \tilde{\boldsymbol{v}} = \boldsymbol{v}^d - \boldsymbol{v} \quad \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{p}_L^d - \boldsymbol{p}_L \\ \boldsymbol{o}_L^d * \boldsymbol{o}_L^{-1} \\ \boldsymbol{p}_R^d - \boldsymbol{p}_R \\ \boldsymbol{o}_R^d * \boldsymbol{o}_R^{-1} \end{bmatrix} \tag{16}$$

are the acceleration, velocity and configuration errors, respectively, and:

$$\boldsymbol{K}_M = \begin{bmatrix} \boldsymbol{K}_{M,L} & \boldsymbol{O}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{K}_{M,R} \end{bmatrix} \in \mathbb{R}^{12\times12} \tag{17}$$

$$\boldsymbol{K}_D = \begin{bmatrix} \boldsymbol{K}_{D,L} & \boldsymbol{O}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{K}_{D,R} \end{bmatrix} \in \mathbb{R}^{12\times12} \tag{18}$$

$$\boldsymbol{K}_P = \begin{bmatrix} \boldsymbol{K}_{P,L} & \boldsymbol{O}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{K}_{P,R} \end{bmatrix} \in \mathbb{R}^{12\times12} \tag{19}$$

are the desired virtual mass, damping and stiffness, respectively.

A constraint involving the end-effectors acceleration can not be directly included in Eq. (4), thus it has to be first manipulated to be expressed as a constraint on the end-effector velocity. In the following, it is shown how second-order constraints can be included in a first order-kinematic control law by leveraging the HQP framework. The acceleration of the end-effectors can be approximated by numerically derivating their velocity assuming a sampling time T_s, as:

$$\boldsymbol{a}(t) = \frac{\boldsymbol{v}(t) - \boldsymbol{v}(t - T_s)}{T_s} , \tag{20}$$

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. (15):

$$\boldsymbol{K}_M\boldsymbol{a}^d - \boldsymbol{K}_M\frac{\boldsymbol{v}}{T_s} + \boldsymbol{K}_M\frac{\boldsymbol{v}(t-T_s)}{T_s} + \boldsymbol{K}_D\boldsymbol{v} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{h} , \tag{21}$$

which can be rewritten as:

$$\left(-\frac{\boldsymbol{K}_M}{T_s} - \boldsymbol{K}_D\right)\boldsymbol{v} = \boldsymbol{h} - \boldsymbol{\gamma} , \tag{22}$$

where :

$$\boldsymbol{\gamma} = \boldsymbol{K}_M\boldsymbol{a}^d - \frac{\boldsymbol{K}_M}{T_s}\boldsymbol{v}(t-T_s) - \boldsymbol{K}_D\boldsymbol{v}^d - \boldsymbol{K}_P\tilde{\boldsymbol{x}} . \tag{23}$$

By folding Eq. (1) in Eq. (22), one finally obtains:

$$\underbrace{\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D\right)\boldsymbol{J}}_{\boldsymbol{J}_{\text{adm}}}\ \dot{\boldsymbol{q}} = \underbrace{-\boldsymbol{\gamma} - \boldsymbol{h}}_{\boldsymbol{b}_{\text{adm}}} \tag{24}$$

which is the expression of the constraint to use in the HQP formulation.

3) Secondary task. Hand-Guiding Task: An alternative secondary task is the *hand-guiding* task, whose aim is to allow the human operator to fully adjust the end-effector configuration in case of complete loss of autonomy due to environmental conditions. It can be easily obtained from the admittance task by setting $\boldsymbol{K}_p = \boldsymbol{O}_{12\times12}$ and $\boldsymbol{a}^d = \boldsymbol{v}^d = \boldsymbol{0}$ in Eq. (24), thus leading to the following form of the constraint in the HQP formulation

$$\underbrace{\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D\right)\boldsymbol{J}}_{\boldsymbol{J}_{\text{hg}}}\ \dot{\boldsymbol{q}} = \underbrace{-\boldsymbol{\gamma} - \boldsymbol{h}}_{\boldsymbol{b}_{\text{hg}}} \tag{25}$$

where:

$$\boldsymbol{\gamma} = -\frac{\boldsymbol{K}_M}{T_s}\boldsymbol{v}(t-T_s) . \tag{26}$$

C. Perception algorithm for the detection of grapes and peduncles

The farming robot can use up to three different Realsense D435i RGB-D cameras located in the head and the wrists of the robot, to detect the grapes, compute the quality of the grapes and localize the grapes and their peduncles. The method that is explained in detail in the article [19], is divided in two parts: (1) detection of the grapes and peduncles; (2) computation of the grape and peduncle localization. The detection of the grapes is done by a method that combines monocular depth [20] and Mask Region-based CNN [21] methods. The computation of localization of the peduncles is done by using two methods: (1) depth estimation of peduncle using the depth map; (2) direct measurement method. Both methods are fused to obtain better results. The resulting depth

estimation achieves a mean squared error of 2.66 cm within a distance of 1 m in the CANOPIES dataset. In Figure 3, the detection of grapes and peduncles with a score above 0.9 is shown. Since the grape detection method computes relative distances using monocular depth, the grapes that are further a specific distance of the others are not considered for detection, that means only grapes closer to the robot arm are taking into account. The detection method can detect several grapes although they are partially overlapped, and in this case, only the area of the seen grape can be computed, and the border between one grape and the other is separated by a straight vertical line. In some of the grapes, several peduncles can be detected in the bunch, and in this case, the point to cut it is the closer to the vineyard cane from where the peduncle is connected. If the peduncle is not detected, the method can estimate where the peduncle is, which can be used to approach the gripper to the peduncle, and then take another image to detect the peduncle and compute the cutting point. However, if finally the peduncle is not seen, as it is the case of Figure 6 bottom Left, the peduncle detection fails.

Fig. 3. Detection of grapes and peduncles. It is only shown the detection outcome considering a confidence score above 0.9.

III. Proposed Shared Control Strategy

The proposed shared control strategy is devised for an agricultural robot engaged in harvesting operations and assumes that the robot is equipped with a specific sensor suite. In the following, we describe: i) the robotic platform taken into consideration; ii) the details of a typical harvesting operation; iii) the devised shared control strategy to change the level of autonomy of the robot.

A. Robot description

We consider a typical use case scenario addressed within the EU-funded project CANOPIES, which foresees the development of a multi-robot system for precision agriculture applications. More in detail, the envisaged robotic platform (named Farming Robot and shown in Figure 1) has a kinematic structure that reflects the one described in Section II-B and has several components, including a tracked mobile base, a torso with 2 degrees of freedom (DOFs), two manipulators with 7 DOFs each, and a head with 2 DOFs. As for the upper body, the torso features a rotational joint and a prismatic joint, enabling adjustments in its height. The two manipulators are equipped with wrench sensors, and each one of them has an Intel RealSense D435 RGB-D sensor mounted on their wrists. Additionally, the head's two DOFs facilitate pan and tilt movements for another Intel RealSense D435 RGB-D sensor situated inside the robot's head.

B. Harvesting operation description

To effectively harvest a grape bunch, the robot must first identify the bunches in its surroundings and accurately estimate the 3D position of the peduncle to cut. The bunches are detected by the perception algorithm explained before and is performed in real-time. Once an estimate of the peduncle's position is available, the harvesting procedure is started and the Trajectory Generation module generates the desired end-effector position and orientation trajectories that connect a sequence of appropriate waypoints using trapezoidal velocity profiles, allowing 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.

In this procedure, the accuracy of peduncle position estimation is particularly critical given the varying distances between the cameras and the grape bunches, as well as potential occlusions caused by leaves or other grape bunches. For this reason, it is often mandatory to perform the grapes and peduncle position estimation procedure from different points of view, and the proposed strategy is to exploit the RGB-D sensors mounted on the wrists for this purpose. Initially, the perception software utilizes RGB-D data from the head-mounted sensor to provide an initial estimate of the bunches' position. Then, if the peduncles are occluded by leaves or other grape bunches (or if the grapes are positioned too far away for a precise estimation by the head camera), we move one of the end-effectors close to the detected bunch to reattempt and improve the estimation of the peduncle position using the wrist-mounted cameras. After this second run of the perception software, the harvesting procedure mentioned above is started.

C. Shared control strategy

Even employing the strategy described in the previous subsection, it is not uncommon for the robot to still fail to get a reliable peduncle position estimate. In order to increase the effectiveness of the harvesting operation, we propose to change the level of autonomy depending on the performances of the perception system. We have envisioned two possible control modes: *autonomous* mode and *hand-guiding* mode. In *autonomous* mode, the robot autonomously performs the needed end-effector motions to perform the entire harvesting procedure, in case a good estimate has been acquired; in *hand-guiding* mode, part of the control of the operation is left to a human operator in case of failures of the perception system, thus resorting to its superior perception and cognitive capabilities. More in detail, in this case, the human operator guides the end-effector towards the point to cut by physically moving the end-effector. Then, the robot

control mode switches back to *autonomous*, and it continues the remaining steps of the harvesting procedure on its own.

The robot behaviors related to the two control modes are implemented by switching the secondary task in the HQP framework. When in *autonomous* mode, the secondary task is the Admittance in Section II-B.2, and the desired mass $\boldsymbol{K}_M$, damping $\boldsymbol{K}_D$ and stiffness $\boldsymbol{K}_P$ are set to:

$$\begin{aligned} \boldsymbol{K}_M &= \text{diag}\{\begin{bmatrix}20 & 20 & 20 & 3 & 3 & 3\end{bmatrix}\} \\ \boldsymbol{K}_D &= \text{diag}\{\begin{bmatrix}253 & 253 & 253 & 27 & 27 & 27\end{bmatrix}\} \\ \boldsymbol{K}_P &= \text{diag}\{\begin{bmatrix}800 & 800 & 800 & 60 & 60 & 60\end{bmatrix}\}. \end{aligned} \tag{27}$$

Moreover, the desired velocity $\boldsymbol{v}^d$ and acceleration $\boldsymbol{a}^d$ are set as the ones computed by the Trajectory Generation module. When in *hand-guiding* mode, the parameters in Eq. (25) are as before, except $\boldsymbol{K}_P$ which is set to $\boldsymbol{K}_P = \boldsymbol{O}_{12\times12}$, and the desired acceleration, velocity and position are all set to zero, allowing the human operator to freely move the end-effector to the actual location of the peduncle to cut Figure 4 shows a graphical representation of the Finite State Machine diagram that implements our shared control strategy.

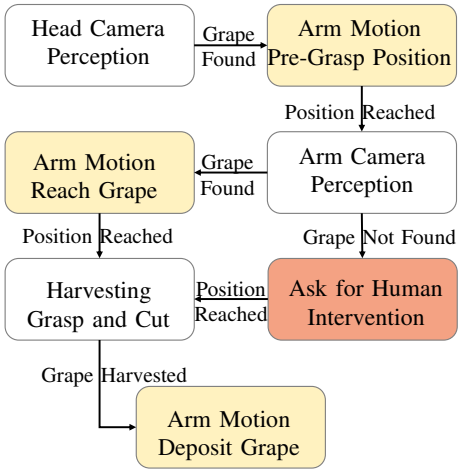

Fig. 4. State Machine describing the sequence of operations for human-robot collaboration in grape harvesting. The colors of the blocks encode the robot control mode. Yellow: *autonomous*, orange: *hand-guiding*.

IV. Experiments

In the following, we report the results obtained during one of the harvesting experiments conducted during the experimental campaigns of the CANOPIES project, held in Aprilia (Italy) in September 2023. We selected a scenario in which the robot required operation is to harvest a bunch of grapes occluded by leaves, to validate and test the proposed shared control strategy. The described experiment is shown in the video accompanying the present paper.

Figure 5 (top) shows the interaction wrench of the right end-effector during the harvesting experiment, Figure 5 (middle) reports the normalized joint positions and velocities and the corresponding thresholds, and Figure 5 (bottom) depicts the x, y and z coordinates of the actual position of the end-effector and the corresponding desired values. The cyan

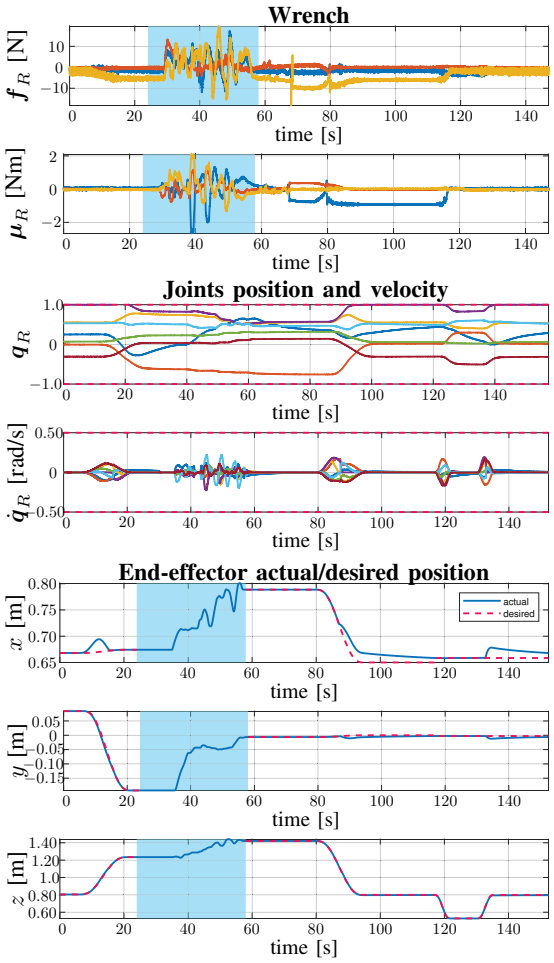

Fig. 5. Top. Interaction wrench of the right end-effector over time during the harvesting experiment. Middle. Normalized joint positions and joint velocities and normalized lower and upper bounds for the right arm. Bottom. End-effector x, y and z actual (blue-solid line) and desired (red-dashed line) coordinates over time. The cyan background highlights the phase in which the system is in *Hand-Guiding Mode*.

background of the top and bottom plots highlights the phase in which the robot is in *Hand-Guiding Mode*, while the white background indicates that it is in *Autonomous Mode*. According to Figure 4, at the beginning the robot is in *autonomous* control mode and runs the perception software making use of the RGB-D sensor of the head (Figure 6. Top Left). Some of the present bunches are detected and localized by the perception system (Figure 6. Top Right); however, peduncles are not clearly visible. In this case, the perception system can estimate the position of the peduncle, and give this information to the robot to move the arm closer and take a new image, but if finally the peduncle is not seen, then the perception system is not able to compute the cutting point.

Then, from $t = 0$s to $t \approx 20$s the robot moves the right end-effector in a position at a predefined distance from one of the bunch, and makes a new estimation attempt (Figure 6. Bottom Left). Even in this case, the peduncle of the target bunch was not recognized and, at this point, the

Fig. 6. Top Left: image of the grapes from the head camera of the robot. Top Right: grapes recognition and peduncle localization. Bottom left: images of grapes taken from the right wrist camera. Bottom Right: human assisting the robot for grape harvesting.

robot asks for the human operator's assistance (by exploiting its text-to-speech module), and its control mode switches to *Hand-Guiding* and remains in this state until $t \approx 58$s. The admittance parameters are changed according to this mode, allowing the human operator to grab the end-effector and manually place it on the point to cut. This is clear from Fig. 5 (top), which shows large interaction forces and moments during this phase, and from Fig. 5 (bottom), which does not show any desired trajectory given that the operator freely drives the end-effector by hand. When he/she releases the end-effector, the robot detects this condition by comparing the norm of the interaction forces and torques with given thresholds $f_{\min}$ and $\tau_{\min}$, which have been set to 3N and 1 Nm, respectively; in detail, when the $||\boldsymbol{f}_R|| \leq f_{\min}$ and $||\boldsymbol{\tau}_R|| \leq \tau_{\min}$ for more than 3s the robot switches back to *autonomous* mode. This parameters have been tuned in function of the noise and the polarization in the wrench readings of the employed sensor. From that point on, the robot autonomously continues the harvesting procedures with the admittance parameters as is Eq. (27). It is worth noticing that, during this phase, the measured interaction wrench is mainly represented by the gravity due to the grape weight (yellow line in the top plot in Fig. 5 (top). Finally, it is worth noticing that Fig. 5 (middle) shows that the (normalized) joint position and velocity limits are always respected during the entire experiment.

V. Conclusions

In this paper, we have envisioned a shared control strategy that dynamically switches the robot mode control depending on task execution conditions as estimated by a perception system. The proposed shared control strategy is based on a proper dynamic modification of the admittance parameters that allows a human operator to physically help the robot when the grape peduncle to cut is partially or totally occluded. Experimental results in an actual vineyard have been shown and described, proving the effectiveness of the proposed strategy. Possible directions for future could be: *i*) adjusting the admittance parameters on the base of EMG sensors mounted on the human arm; *ii*) estimating the human intention to make the robot capable of exhibiting a proactive behavior in the interaction; and *iii*) exploiting the dual-arm nature of the system achieving *active-perception* approaches.

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, p. 1729881419857428, 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 Conf. 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 Rob. and Autom. Letters*, vol. 6, no. 2, pp. 3720–3727, 2021.

[13] J. Chen and P. I. Ro, "Human intention-oriented variable admittance control with power envelope regulation in physical human-robot interaction," *Mechatronics*, vol. 84, p. 102802, 2022.

[14] R. Balachandran, M. De Stefano, H. Mishra, C. Ott, and A. Albu-Schaeffer, "Passive arbitration in adaptive shared control of robots with variable force and stiffness scaling," *Mechatronics*, vol. 90, p. 102930, 2023.

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

[16] 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.

[17] 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.

[18] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[19] 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, 2023.

[20] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.