Ref. Ares(2023)8905618 - 30/12/2023

This project is funded by the European Union
G.A No 101016906

Project Number: 101016906
Start Date of Project: 2021/01/01
Duration: 48 months

Type of document: D5.3 – V1

Mobile arms physical interaction

Dissemination Level	PUBLIC
Submission Date	2023-12-30
Work Package	WP5
Task	T5.3
Type	Report
Version	1
Author	UNICLAM
Approved by	UNIROMA3, KTH

DISCLAIMER:
The sole responsibility for the content of this deliverable rests with the authors. It does not necessarily reflect the opinion of the European Union. Neither the Directorate-General for Communications Networks, Content and Technology, Artificial Intelligence and Digital Industry nor the European Commission are responsible for any use that may be made of the information contained therein.

This project is funded by the European Union
G.A No 101016906

Executive Summary
This document describes the design and testing of the physical interaction functionalities developed for the CANOPIES Farming Robot. We first identify several basic interaction strategies that allow the robot to react in different manners to external forces intentionally exerted by the human operator. Then, a method for classifying the type of contact between *intentional* and *accidental* is presented, with the latter requiring to account for the human safety. In detail, the basic interaction strategies are included in a Finite State Machine algorithm that switches among them depending on the classified type of contact, to increase the flexibility of the robot in physical human-robot collaboration scenarios. The envisioned basic physical interaction strategies and the high-level policies in which they are employed are validated via dedicated simulations and indoor experiments of the CANOPIES farming robot.

Table of Contents

1 Introduction . . . 6
2 System modeling . . . 7
2.1 Robot kinematic model . . . 7
2.2 Robot dynamic model . . . 7
2.3 Human model . . . 10
2.4 Including acceleration constraints in the first order HQP framework . . . 10
3 Designed basic physical human-robot-interaction strategies . . . 13
3.1 Admittance control strategy . . . 13
3.2 Hand-guiding control strategy . . . 14
3.3 Shared control strategy . . . 14
3.3.1 Designed strategy . . . 15
3.3.2 Human parameters estimation . . . 16
3.3.3 Solution of the optimal LQT problem in Section 3.3.2 . . . 17
4 Extension to distributed cooperative multi-robot systems . . . 19
4.1 Human-multi robot system modeling . . . 20
4.2 Communication modeling . . . 21
4.3 Distributed human and internal wrench estimation . . . 21
4.4 Generation and estimation of the object trajectory $\boldsymbol{x}_v$ . . . 23
4.4.1 Leader robot . . . 23
4.4.2 Follower robot . . . 23
4.5 Distributed low-level control . . . 24
5 Neural-network based contact detection and classification . . . 26
5.1 Introduction and motivation . . . 26
5.2 Problem formulation and possible solutions overview . . . 27
5.2.1 Contact detection based on joint torque sensors . . . 27
5.2.2 Contact detection by wrist wrench sensors . . . 30
6 Simulation and experimental validation . . . 32
6.1 Basic human-robot interaction strategies . . . 32
6.1.1 Admittance control . . . 32
6.1.2 Hand-guiding control . . . 35
6.2 Human-robot cooperative harvesting . . . 37
6.2.1 Harvesting operation description . . . 37
6.2.2 Human-robot cooperation algorithm . . . 37
6.2.3 Experimental results . . . 38
6.2.4 Shared control . . . 40
6.3 Neural-network contact detection and classification . . . 49
6.3.1 Experiments on a Kinova Jaco^2 robot . . . 49
6.3.2 Experiment on the Canopies farming robot . . . 53
7 Conclusions . . . 56
8 Appendix . . . 57
8.1 Proof of Theorem 1 . . . 57

This project is funded by the European Union G.A No 101016906

9 Reference to web content 59
10 References 59

This project is funded by the European Union
G.A No 101016906

Abbreviations and Acronyms

CBF(s)	Control Barrier Function(s)
DoF(s)	Degree(s) of Freedom
(H)QP	(Hierarchical) Quadratic Programming
RLS	Recursive Least Square

This project is funded by the European Union G.A No 101016906

1 Introduction

One of the objectives of the CANOPIES project is to embed the Farming Robot with the capability to physically interact with human operators and the environment, always ensuring the integrity of the robot itself and the safety of the farmers during the interaction. In order to provide such capability, the design of proper interaction and force control algorithms and strategies is needed. This Deliverable gives details about the implemented algorithms and the high-level architectures in which they are employed, in order to enhance the execution of the typical tasks that the Farming Robot has to perform (such as the harvesting procedure) including the possibility of physically cooperating with a human operator.

In Section 2, we first describe the kinematic and dynamic model of the robot, as well as the considered human model. These preliminaries are needed for the following derivation of the designed basic human-robot interaction strategies, which are detailed in Section 3. Such strategies have to be considered as basic blocks that can be properly adopted following high-level policies to build different functional behaviors that the robot has to follow depending on the context. For example, in a harvesting task, the behavior of the robot can be changed to give the possibility to ask for help in case of particularly challenging scenarios. In detail, when there are occlusions that prevent the perception system from correctly detecting the grapes, the human may grasp the end-effector and bring it, according to a hand-guiding behavior, to the correct location, enabling a cooperative harvesting operation between the human and the robot. Furthermore, in Section 4 it is shown how the proposed strategies might be extended to the case of multiple mobile manipulators grasping a rigid object which the human operator might interact with and in absence of a central control unit, thus requiring the design of distributed control approaches.

Since in complex scenarios, accidental contacts might happen between the human and robot, in Section 5, we describe a methodology for allowing the robot to classify the kind of contact between accidental and intentional, and this information is fed into a Finite State Machine algorithm that changes the basic human-robot interaction strategy, allowing the creation of different behaviors that can be adapted to the specific scenario at hand. Finally, in Section 6, we describe experimental and simulation results of the implemented strategies.

This project is funded by the European Union G.A No 101016906

2 System modeling

2.1 Robot kinematic model

Let us consider a mobile robot equipped with a movable torso and a dual arm system like the CANOPIES Farming Robot in Figure 1, in which each end-effector configuration is denoted by $\boldsymbol{\rho}_y = [\boldsymbol{p}_y^{\mathrm{T}} \ \boldsymbol{o}_y^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^7$, ($y = L, R$, 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 $\boldsymbol{\rho} = [\boldsymbol{\rho}_L^{\mathrm{T}} \ \boldsymbol{\rho}_R^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{14}$ 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:

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

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

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

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.

2.2 Robot dynamic model

With reference to the Figure 1, the following dynamical model is assumed for the robot

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\mathcal{T}} + \boldsymbol{\tau}_h \tag{3}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint position vector, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the joint torque vector, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix, $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the centrifugal and Coriolis terms vector, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity terms vector, $\boldsymbol{\tau}_{\mathcal{T}} = \boldsymbol{J}_{(\cdot)}(\boldsymbol{q})^{\mathrm{T}}\boldsymbol{h}_{\mathcal{T}}$ (where $(\cdot)$ is either L or R) is the torque vector induced by the interaction $\boldsymbol{h}_{\mathcal{T}} \in \mathbb{R}^6$ of the left or right end-effector with the environment to carry out a given task $\mathcal{T}$, $\boldsymbol{\tau}_h = \boldsymbol{J}_{(\cdot),P}(\boldsymbol{q})^{\mathrm{T}}\boldsymbol{h}_h$ is the torque vector induced by the human wrench $\boldsymbol{h}_h \in \mathbb{R}^6$ exerted at a generic point P along either the left or right arm, with $\boldsymbol{J}_{(\cdot),P}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ the robot Jacobian matrix at this point. The interaction wrench $\boldsymbol{h}_{(\cdot)}$ is such that $\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{T}} & \boldsymbol{\mu}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6$ that is composed of force, $\boldsymbol{f} \in \mathbb{R}^3$, and torque, $\boldsymbol{\mu} \in \mathbb{R}^3$, components. It is worth noticing that in (3) it was assumed, for simplicity, that only one arm at the

This project is funded by the European Union G.A No 101016906



Figure 1: The human hand interacting with the right arm of the CANOPIES farming robot (simulation environment on the left and real scenario on the right).

time is interacting with the environment or with the human.

Since in many applications the human operator is expected to interact with the end-effector of the robot (it is assumed, as before, that the interaction happens with one end-effector at a time), it is worth considering the corresponding operational space dynamics of the model in (3) away from kinematic singularities and that, for each arm, can be expressed as [22]

$$\boldsymbol{M}_{(\cdot)}(\boldsymbol{x}_{(\cdot)})\ddot{\boldsymbol{x}}_{(\cdot)}+\boldsymbol{C}_{(\cdot)}(\boldsymbol{x}_{(\cdot)},\dot{\boldsymbol{x}}_{(\cdot)})\dot{\boldsymbol{x}}_{(\cdot)}+\boldsymbol{\eta}_{(\cdot)}(\boldsymbol{x}_{(\cdot)},\dot{\boldsymbol{x}}_{(\cdot)})=\boldsymbol{u}_{(\cdot),x}-\boldsymbol{h}_{(\cdot)}=\boldsymbol{Y}_{(\cdot)}(\boldsymbol{x}_{(\cdot)},\dot{\boldsymbol{x}}_{(\cdot)},\ddot{\boldsymbol{x}}_{(\cdot)})\boldsymbol{\pi}_{(\cdot)} \tag{4}$$

where $(\cdot)$ is either L or R, $\boldsymbol{u}_x \in \mathbb{R}^p$ is the control input, $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{p}^{\mathrm{T}} & \boldsymbol{\phi}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^p$ is the end-effector configuration in terms of position $\boldsymbol{p}$ and orientation $\boldsymbol{\phi}$ of frame Σ_r with respect to Σ_w (by using Euler angles), $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ is the symmetric positive definite inertia matrix, $\boldsymbol{C} \in \mathbb{R}^{p\times p}$ is the centrifugal and Coriolis terms matrix, $\boldsymbol{\eta} \in \mathbb{R}^p$ is the viscous friction and gravity terms vector, and $\boldsymbol{h} \in \mathbb{R}^6$, similarly to how defined before, is the interaction wrench (either with the human or the environment). In addition, $\boldsymbol{Y} \in \mathbb{R}^{p\times n_\pi}$ is the regressor matrix and $\boldsymbol{\pi} \in \mathbb{R}^{n_\pi}$ is the constant vector of the dynamic parameters of the robot. By adopting Christoffel symbols of the first kind associated with the matrix $\boldsymbol{M}$ for $\boldsymbol{C}$, the following property holds true [10]. In the model above, we assumed that the model might differ between arms.

Property 1. *Matrix $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew-symmetric such that $\boldsymbol{v}^{\mathrm{T}}(\dot{\boldsymbol{M}} - 2\boldsymbol{C})\boldsymbol{v} = 0, \forall\, \boldsymbol{v} \in \mathbb{R}^p$.*

Considering that, in real settings, only an estimate of model (4) is available, this model can be rewritten as:

$$\hat{\boldsymbol{M}}_{(\cdot),i}\ddot{\boldsymbol{x}}_{(\cdot)} + \hat{\boldsymbol{C}}_{(\cdot),i}\dot{\boldsymbol{x}}_{(\cdot)} + \hat{\boldsymbol{\eta}}_{(\cdot),i} = \boldsymbol{Y}(\boldsymbol{x}_{(\cdot)},\dot{\boldsymbol{x}}_{(\cdot)},\ddot{\boldsymbol{x}}_{(\cdot)})\hat{\boldsymbol{\pi}} = \boldsymbol{u}_{(\cdot),x} - \boldsymbol{h}_{(\cdot)} - \boldsymbol{Y}(\boldsymbol{x}_{(\cdot)},\dot{\boldsymbol{x}}_{(\cdot)},\ddot{\boldsymbol{x}}_{(\cdot)})\tilde{\boldsymbol{\pi}}_{(\cdot)} \tag{5}$$

where $\tilde{\boldsymbol{\pi}}_{(\cdot)} = \boldsymbol{\pi}_{(\cdot)} - \hat{\boldsymbol{\pi}}_{(\cdot)}$ and the symbol $\hat{\cdot}$ denotes the estimate of the corresponding quantity (their dependency on the state variables were omitted for the sake of brevity).

Coherently with Deliverables 5.1 and 5.2 and with the specifications of the CANOPIES Farming Robot, also for the case of human-robot physical interaction, we assume that the robot can track a joint space reference trajectory $\boldsymbol{q}_r(t) \in \mathbb{R}^n$, i.e. $\boldsymbol{q}_r \approx \boldsymbol{q}$. In detail, we assume the following second-order model

$$\ddot{\boldsymbol{q}} = \boldsymbol{u} \tag{6}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is a virtual control input to be designed.
Additionally, by virtue of eq. (6), the well-known second-order kinematic relationship at a generic point P on the robot structure gives

$$\ddot{\boldsymbol{x}}_P = \boldsymbol{J}_P(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_P(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{J}_P(\boldsymbol{q})\boldsymbol{u} + \dot{\boldsymbol{J}}_P(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \tag{7}$$

where $\boldsymbol{x}_P = \begin{bmatrix}\boldsymbol{p}_P^{\mathrm{T}} & \boldsymbol{\phi}_P^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6$ is the configuration of a frame centered in P with position $\boldsymbol{p}_P$ and orientation $\boldsymbol{\phi}_P$. In the following, we omit subscript P when the kinematics is referred to one of the robot end-effector.

Finally, concerning the adopted notation, in the case of scenarios involving multi-robot systems tackled in Section 4, we use the subscript $(\cdot)_i$ to denote that the corresponding variable is related to robot i. For instance, the dynamic model in (5) becomes

$$\hat{\boldsymbol{M}}_{(\cdot),i}\ddot{\boldsymbol{x}}_{(\cdot),i} + \hat{\boldsymbol{C}}_{(\cdot),i}\dot{\boldsymbol{x}}_{(\cdot),i} + \hat{\boldsymbol{\eta}}_{(\cdot),i} = \boldsymbol{Y}_{(\cdot),i}(\boldsymbol{x}_{(\cdot),i}, \dot{\boldsymbol{x}}_{(\cdot),i}, \ddot{\boldsymbol{x}}_{(\cdot),i})\hat{\boldsymbol{\pi}}_{(\cdot),i} = \boldsymbol{u}_{(\cdot),x,i} - \boldsymbol{h}_{(\cdot),i} - \boldsymbol{Y}_{(\cdot),i}(\boldsymbol{x}_{(\cdot),i}, \dot{\boldsymbol{x}}_{(\cdot),i}, \ddot{\boldsymbol{x}}_{(\cdot),i})\tilde{\boldsymbol{\pi}}_{(\cdot),i} \tag{8}$$

Concerning the notation used in the following, let us define the following quantities for a dual arm system:

$$\ddot{\bar{\boldsymbol{x}}} = \begin{bmatrix}\ddot{\boldsymbol{x}}_L \\ \ddot{\boldsymbol{x}}_R\end{bmatrix} = \begin{bmatrix}\ddot{\boldsymbol{p}}_L \\ \ddot{\boldsymbol{\phi}}_L \\ \ddot{\boldsymbol{p}}_R \\ \ddot{\boldsymbol{\phi}}_R\end{bmatrix} \quad \dot{\bar{\boldsymbol{x}}} = \begin{bmatrix}\dot{\boldsymbol{x}}_L \\ \dot{\boldsymbol{x}}_R\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{p}}_L \\ \dot{\boldsymbol{\phi}}_L \\ \dot{\boldsymbol{p}}_R \\ \dot{\boldsymbol{\phi}}_R\end{bmatrix} \quad \bar{\boldsymbol{x}} = \begin{bmatrix}\boldsymbol{x}_L \\ \boldsymbol{x}_R\end{bmatrix} = \begin{bmatrix}\boldsymbol{p}_L \\ \boldsymbol{\phi}_L \\ \boldsymbol{p}_R \\ \boldsymbol{\phi}_R\end{bmatrix}, \tag{9}$$

that make use, as done in (4), of Euler angles for the orientation.
Similarly, let us introduce the following vectors:

$$\bar{\boldsymbol{a}} = \begin{bmatrix}\boldsymbol{a}_L \\ \boldsymbol{a}_R\end{bmatrix} = \begin{bmatrix}\ddot{\boldsymbol{p}}_L \\ \dot{\boldsymbol{\omega}}_L \\ \ddot{\boldsymbol{p}}_R \\ \dot{\boldsymbol{\omega}}_R\end{bmatrix}, \quad \bar{\boldsymbol{v}} = \begin{bmatrix}\boldsymbol{v}_L \\ \boldsymbol{v}_R\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{p}}_L \\ \boldsymbol{\omega}_L \\ \dot{\boldsymbol{p}}_R \\ \boldsymbol{\omega}_R\end{bmatrix}, \quad \bar{\boldsymbol{\rho}} = \begin{bmatrix}\boldsymbol{\rho}_L \\ \boldsymbol{\rho}_R\end{bmatrix} = \begin{bmatrix}\boldsymbol{p}_L \\ \boldsymbol{o}_L \\ \boldsymbol{p}_R \\ \boldsymbol{o}_R\end{bmatrix}, \tag{10}$$

where $\dot{\boldsymbol{\omega}}_{(\cdot)}, \boldsymbol{\omega}_{(\cdot)}$ and $\boldsymbol{o}_{(\cdot)}$ express the angular acceleration, angular velocity and quaternion of the $\{L\text{eft}, R\text{ight}\}$ end-effector, respectively. Therefore, differently than (9), quaternions are used in this case for representing the orientation of the end-effectors.
Finally, the following remark holds for the remaining of this deliverable.

Remark 1. *For the sake of notation compactness, we remove in the following, when no ambiguity is generated, the explicit reference to the left or right arm unless otherwise specified, in the sense that the relationships and strategies designed are valid for both arms.*

2.3 Human model

The model assumed for the human arm endpoint, which will be referred to as the human model in the following sections, is

$$-\boldsymbol{D}_h\dot{\boldsymbol{p}}_h + \boldsymbol{K}_h(\boldsymbol{p}_h^d - \boldsymbol{p}_h) = \boldsymbol{f}_h \tag{11}$$

where $\boldsymbol{f}_h \in \mathbb{R}^m$ is the vector of the forces exerted by the human, $\boldsymbol{p}_h \in \mathbb{R}^m$ is the human arm end point position, which will be referred to as human position, with respect to Σ_w (see Figure 1), $\boldsymbol{p}_h^d \in \mathbb{R}^m$ is the human desired position and $\boldsymbol{D}_h \in \mathbb{R}^{m\times m}$ and $\boldsymbol{K}_h \in \mathbb{R}^{m\times m}$ are the matrices that regulate the human damping and stiffness actions, respectively. As stated, for example, in [6] and [38], these matrices are generally time-varying and their variation depends on the activation of the human arm muscles. The model in (11) is commonly assumed as representative for the human arm end point, e.g. in [14, 28, 44], and its validation can be found, for instance, in [2, 38] with the inertial term shown to be negligible compared to damping and stiffness. Moreover, by following the assumptions in [14, 37, 44], the decoupling of the impedance parameters in the different directions is assumed, i.e., $\boldsymbol{D}_h = \mathrm{diag}(\boldsymbol{d}_h)$ and $\boldsymbol{K}_h = \mathrm{diag}(\boldsymbol{k}_h)$ with $\boldsymbol{d}_h \in \mathbb{R}^m$ and $\boldsymbol{k}_h \in \mathbb{R}^m$. Then, the model in (11) can be rewritten as

$$\boldsymbol{Y}_h(\boldsymbol{p}_h, \dot{\boldsymbol{p}}_h)\boldsymbol{\pi}_h = \boldsymbol{f}_h \tag{12}$$

with

$$\begin{cases} \boldsymbol{Y}_h &= [-\mathrm{diag}(\dot{\boldsymbol{p}}_h) \quad -\mathrm{diag}(\boldsymbol{p}_h) \quad \boldsymbol{I}_m] \in \mathbb{R}^{m\times 3m} \\ \boldsymbol{\pi}_h &= [\boldsymbol{d}_h^{\mathrm{T}} \quad \boldsymbol{k}_h^{\mathrm{T}} \quad (\boldsymbol{K}_h\boldsymbol{p}_h^d)^{\mathrm{T}})]^{\mathrm{T}} \in \mathbb{R}^{3m}. \end{cases} \tag{13}$$

It is worth remarking that the human parameters in $\boldsymbol{\pi}_h$ are *unknown* and might be time-varying.
Finally, the model in (11), as stressed in [44], is motivated by the fact that the stiffness range is typically greater than those associated with inertia and damping terms.

2.4 Including acceleration constraints in the first order HQP framework

The basic human-robot interaction strategies devised in Section 3 are relative to manipulators which allow to input joint torques for control purposes (see input $\boldsymbol{\tau}$ in model (3)). However, since many off-the-shelf manipulators do not offer this possibility and neither does the first prototype of the CANOPIES robot, a strategy to equip these robots with interaction capabilities is described in this section. In detail, it is shown how second-order constraints (like those generated by the human-robot interaction strategies) are included in a first order-kinematic control law by leveraging the Hierarchical Quadratic Programming (HQP) framework described in Deliverable 5.1 and briefly recalled here for self-consistency.
The HQP control framework [15, 23] allows assigning a strict priority order to multiple tasks and computing a solution that fulfills the hierarchy. 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:

$$\begin{aligned} \min_{\boldsymbol{w}_i, \dot{\boldsymbol{q}}} \quad & \frac{1}{2} \boldsymbol{w}_i^T \boldsymbol{Q}_{w,i} \boldsymbol{w}_i \\ \text{s.t.} \quad & \underline{\boldsymbol{b}}_k \leq \boldsymbol{J}_k(\boldsymbol{q}) \dot{\boldsymbol{q}} + \boldsymbol{w}_k^\star \leq \bar{\boldsymbol{b}}_k, \qquad \forall k \in 1, \ldots, i-1 \\ & \underline{\boldsymbol{b}}_i \leq \boldsymbol{J}_i(\boldsymbol{q}) \dot{\boldsymbol{q}} + \boldsymbol{w}_i \leq \bar{\boldsymbol{b}}_i, \end{aligned} \tag{14}$$

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, 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 in conflict with those included in the higher-priority tasks, and their minimization is aimed at minimizing the error in the execution of the task.
As it will be detailed in Section 3, the physical interaction approaches are aimed at assigning a specific dynamics to the system, which can be expressed in the general forms:

$$\boldsymbol{K}_M \, \bar{\boldsymbol{a}}(t) + \boldsymbol{K}_D \, \bar{\boldsymbol{v}}(t) + \boldsymbol{K}_P \, \bar{\boldsymbol{\rho}}(t) = \boldsymbol{u}_{\mathsf{v}}(t) \, , \tag{15}$$

or:

$$\boldsymbol{K}_M \, \ddot{\bar{\boldsymbol{x}}}(t) + \boldsymbol{K}_D \, \dot{\bar{\boldsymbol{x}}}(t) + \boldsymbol{K}_P \, \bar{\boldsymbol{x}}(t) = \boldsymbol{u}_{\mathsf{v,x}}(t) \, , \tag{16}$$

depending on the considered orientation representation, where:

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

are the desired virtual mass, damping and stiffness, respectively, and $\boldsymbol{u}_{\mathsf{v}}$ is a generic virtual control input that will be defined in the following Section depending on the adopted interaction strategy.
A constraint involving the end-effectors acceleration can not be directly included in Eq. (14), thus it has to be first manipulated to be expressed as a constraint on the end-effector velocity.
The acceleration of the end-effectors can be approximated by numerically derivating their velocity assuming a sampling time T_s (given the digital implementation of the control law), as:

$$\bar{\boldsymbol{a}}(t) = \frac{\bar{\boldsymbol{v}}(t) - \bar{\boldsymbol{v}}(t - T_s)}{T_s} \, , \tag{18}$$

where $\bar{\boldsymbol{v}}(t)$ is the current velocity and $\bar{\boldsymbol{v}}(t - T_s)$ is the velocity at the previous time step. Substituting this expression in Eq. (15), it is possible to rearrange the terms to obtain the following expression:

$$\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D \right) \bar{\boldsymbol{v}}(t) = \boldsymbol{u}_{\mathsf{v}}(t) - \frac{\boldsymbol{K}_M}{T_s} \, \bar{\boldsymbol{v}}(t - T_s) - \boldsymbol{K}_P \, \bar{\boldsymbol{\rho}}(t) \, , \tag{19}$$

which, by folding Eq. (1), can be rewritten as:

$$\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D \right) \boldsymbol{J} \dot{\boldsymbol{q}}(t) = \boldsymbol{u}_{\mathsf{v}}(t) - \frac{\boldsymbol{K}_M}{T_s} \bar{\boldsymbol{v}}(t - T_s) - \boldsymbol{K}_P \, \bar{\boldsymbol{\rho}}(t) \, . \tag{20}$$

This kind of expression can be included in the HQP control framework in Eq. (14) at the desired priority level, as an equality constraint on the joint velocities vector in the form:

This project is funded by the European Union G.A No 101016906

$$\boldsymbol{J}_{\mathsf{x}}\,\dot{\boldsymbol{q}} + \boldsymbol{w}_{\mathsf{x}} = \boldsymbol{b}_{\mathsf{x}}\ , \tag{21}$$

where $\boldsymbol{J}_{\mathsf{x}} = \left[\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D\right)\boldsymbol{J}\right], \boldsymbol{b}_{\mathsf{x}} = \boldsymbol{u}_{\mathsf{v}}(t) - \frac{\boldsymbol{K}_M}{T_s}\,\bar{\boldsymbol{v}}(t-T_s) - \boldsymbol{K}_P\,\boldsymbol{\rho}(t)$ and $\boldsymbol{w}_{\mathsf{X}}$ is a slack variable needed if higher-priority constraints are included in the hierarchy.
It is worth noticing that the same considerations can be done by replacing $\bar{\boldsymbol{a}}$, $\bar{\boldsymbol{v}}$ and $\bar{\boldsymbol{\rho}}$ with $\ddot{\bar{\boldsymbol{x}}}$, $\dot{\bar{\boldsymbol{x}}}$ and $\bar{\boldsymbol{x}}$, respectively, and by using the analytic jcobian instead of the geometric one in the above equations.

3 Designed basic physical human-robot-interaction strategies

The following designed basic strategies for physical human-robot interaction are described in detail:

- Admittance control strategy (Section 3.1): an admittance behavior is imposed to the robot's end-effector where admittance parameters can be changed according to the desired interaction dynamics;
- Hand-guiding control strategy (Section 3.2): it allows the user to program the robot by physically dragging it between the desired locations in the workspace;
- Shared control strategy (Section 3.3): both the robot and the human operator contribute to the control of the robot, and the level of autonomy of the latter depends on the human control input.

3.1 Admittance control strategy

This human-robot interaction strategy allows the robot to follow a desired trajectory with its end-effectors while exhibiting a compliant behavior with respect to external forces (in order to limit them) that might occur for contact between the robot and a human operator or the environment.
Let us assume that the desired trajectory for the end-effectors can be generated by a dedicated module as:

$$\bar{\boldsymbol{a}}^d = \begin{bmatrix} \boldsymbol{a}_L^d \\ \boldsymbol{a}_R^d \end{bmatrix} = \begin{bmatrix} \ddot{\boldsymbol{p}}_L^d \\ \boldsymbol{\alpha}_L^d \\ \ddot{\boldsymbol{p}}_R^d \\ \boldsymbol{\alpha}_R^d \end{bmatrix} \quad \bar{\boldsymbol{v}}^d = \begin{bmatrix} \boldsymbol{v}_L^d \\ \boldsymbol{v}_R^d \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_L^d \\ \boldsymbol{\omega}_L^d \\ \dot{\boldsymbol{p}}_R^d \\ \boldsymbol{\omega}_R^d \end{bmatrix} \quad \bar{\boldsymbol{\rho}}^d = \begin{bmatrix} \boldsymbol{\rho}_L^d \\ \boldsymbol{\rho}_R^d \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_L^d \\ \boldsymbol{o}_L^d \\ \boldsymbol{p}_R^d \\ \boldsymbol{o}_R^d \end{bmatrix}, \tag{22}$$

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, for the Right and Left end-effectors.
Assuming the presence of a wrench sensor mounted on the wrist of the manipulators, it is possible to define the vector:

$$\bar{\boldsymbol{h}} = \begin{bmatrix} \boldsymbol{h}_L \\ \boldsymbol{h}_R \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_L \\ \boldsymbol{\mu}_L \\ \boldsymbol{f}_R \\ \boldsymbol{\mu}_R \end{bmatrix} \tag{23}$$

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:

$$\boldsymbol{K}_M \, \tilde{\bar{\boldsymbol{a}}} + \boldsymbol{K}_D \, \tilde{\bar{\boldsymbol{v}}} + \boldsymbol{K}_P \, \tilde{\bar{\boldsymbol{\rho}}} = \bar{\boldsymbol{h}}, \tag{24}$$

where:

$$\tilde{\bar{\boldsymbol{a}}} = \bar{\boldsymbol{a}}^d - \bar{\boldsymbol{a}}, \quad \tilde{\bar{\boldsymbol{v}}} = \bar{\boldsymbol{v}}^d - \bar{\boldsymbol{v}}, \quad \tilde{\bar{\boldsymbol{\rho}}} = \begin{bmatrix} \boldsymbol{p}_L^d - \boldsymbol{p}_L \\ \boldsymbol{o}_L^d * \boldsymbol{o}_L^{-1} \\ \boldsymbol{p}_R^d - \boldsymbol{p}_R \\ \boldsymbol{o}_R^d * \boldsymbol{o}_R^{-1} \end{bmatrix} \tag{25}$$

are the acceleration, velocity and configuration errors, and $\boldsymbol{K}_M, \boldsymbol{K}_D, \boldsymbol{K}_P \in \mathbb{R}^6$ are the desired virtual mass, damping and stiffness, respectively, that can be either constant or time-varying.

This project is funded by the European Union G.A No 101016906

Approximating the acceleration of the end-effectors by numerically deriving their velocity as in Eq. (18), and substituting its expression in Eq. (24):

$$\boldsymbol{K}_M \bar{\boldsymbol{a}}^d - \boldsymbol{K}_M \frac{\bar{\boldsymbol{v}}}{T_s} + \boldsymbol{K}_M \frac{\bar{\boldsymbol{v}}(t-T_s)}{T_s} + \boldsymbol{K}_D\, \bar{\boldsymbol{v}} + \boldsymbol{K}_P\, \tilde{\bar{\boldsymbol{\rho}}} = \bar{\boldsymbol{h}}\ , \tag{26}$$

which can be rewritten as:

$$\left(-\frac{\boldsymbol{K}_M}{T_s} - \boldsymbol{K}_D\right) \bar{\boldsymbol{v}} = \bar{\boldsymbol{h}} - \boldsymbol{\gamma}\ , \tag{27}$$

where :

$$\boldsymbol{\gamma} = \boldsymbol{K}_M\, \bar{\boldsymbol{a}}^d - \frac{\boldsymbol{K}_M}{T_s}\, \bar{\boldsymbol{v}}(t-T_s) - \boldsymbol{K}_D\, \bar{\boldsymbol{v}}^d - \boldsymbol{K}_P\, \tilde{\bar{\boldsymbol{\rho}}}\ . \tag{28}$$

By folding Eq. (1) in Eq. (27), one finally obtains:

$$\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D\right) \boldsymbol{J}\, \dot{\boldsymbol{q}} = -\boldsymbol{\gamma} - \bar{\boldsymbol{h}}. \tag{29}$$

This expression can be used as a constraint in the Hierarchical Quadratic Programming control framework, as stated in Section 2.4, as:

$$\boldsymbol{J}_{\text{adm}}\, \dot{\boldsymbol{q}} = \boldsymbol{b}_{\text{adm}}\ , \tag{30}$$

where $\boldsymbol{J}_{\text{adm}} = \left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D\right) \boldsymbol{J}$ and $\boldsymbol{b}_{\text{adm}} = -\boldsymbol{\gamma} - \bar{\boldsymbol{h}}$.

3.2 Hand-guiding control strategy

An alternative human-robot interaction strategy is the hand-guiding control, whose aim is to allow the human operator to fully adjust the end-effector configuration by manually carrying it to the desired location. This feature can be useful, for example, to carry the cutting tool to the grape peduncle location in those cases where it is obstructed by leaves (see Section 6.2). It can be easily obtained from the admittance task by setting $\boldsymbol{K}_p = \boldsymbol{O}_{12\times 12}$ and $\bar{\boldsymbol{a}}^d = \bar{\boldsymbol{v}}^d = \boldsymbol{0}$ in Eq. (24), thus leading to the following form of the constraint in the HQP formulation:

$$\underbrace{\left(\frac{\boldsymbol{K}_M}{T_s} + \boldsymbol{K}_D\right) \boldsymbol{J}}_{\boldsymbol{J}_{\text{hg}}}\ \dot{\boldsymbol{q}} = \underbrace{-\boldsymbol{\gamma} - \bar{\boldsymbol{h}}}_{\boldsymbol{b}_{\text{hg}}} \tag{31}$$

where:

$$\boldsymbol{\gamma} = -\frac{\boldsymbol{K}_M}{T_s}\, \bar{\boldsymbol{v}}(t-T_s)\ . \tag{32}$$

3.3 Shared control strategy

With regards to the shared control strategy described in this section, we envisage the case in which the robot can dynamically balance its action based on human and robot tasks and human arm parameters. As per Remark 1, for the sake of simplicity the devised strategy is presented here in the present deliverable for the single arm case. In detail, let us assume that the robot's desired trajectory $\boldsymbol{x}_r^d \in \mathbb{R}^p$ (with derivative $\dot{\boldsymbol{x}}_r^d \in \mathbb{R}^p$) is assigned.

The objective is to design the robot control input $\boldsymbol{u}_x$ in (4) such that the robot end-effectors behave according to the following dynamics

$$\boldsymbol{K}_M \ddot{\boldsymbol{x}}_v = \boldsymbol{u}_v \tag{33}$$

where $\boldsymbol{K}_M \in \mathbb{R}^{p\times p}$ is the virtual inertia matrix and $\boldsymbol{u}_v \in \mathbb{R}^p$ represents the virtual input to be defined to optimize the following cost function

$$J = \frac{1}{2}\int_{t_0}^{+\infty} \Big(\underbrace{(\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_r^d)^{\mathrm{T}} \boldsymbol{Q}_{r,d} (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_r^d)}_{i} + \underbrace{\boldsymbol{u}_v^{\mathrm{T}} \boldsymbol{R}_v \boldsymbol{u}_v}_{ii} + \underbrace{(\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{p}}_h^d)^{\mathrm{T}} \boldsymbol{Q}_{h,d} (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{p}}_h^d)}_{iii} + \underbrace{\boldsymbol{f}_h^{\mathrm{T}} \boldsymbol{R}_h \boldsymbol{f}_h}_{iv} \Big) dt \tag{34}$$

where $\bar{\boldsymbol{x}}_v = \begin{bmatrix} \boldsymbol{x}_v^{\mathrm{T}} & \dot{\boldsymbol{x}}_v^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2p}$ is the state aggregate vector, $\bar{\boldsymbol{x}}_r^d = \begin{bmatrix} (\boldsymbol{x}_r^d)^{\mathrm{T}} & (\dot{\boldsymbol{x}}_r^d)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2p}$ and $\bar{\boldsymbol{p}}_h^d = \begin{bmatrix} (\boldsymbol{p}_h^d)^{\mathrm{T}} & \boldsymbol{0}_{2p-m}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2p}$ are the aggregate vectors of the robots and human desired trajectory, respectively, $\boldsymbol{R}_h \in \mathbb{R}^{m\times m}$ and $\boldsymbol{R}_v \in \mathbb{R}^{p\times p}$ are symmetric positive definite matrices, $\boldsymbol{Q}_{r,d} \in \mathbb{R}^{2p\times 2p}$ is a symmetric positive semi-definite matrix, $\boldsymbol{Q}_{h,d} \in \mathbb{R}^{2p\times 2p}$ is defined as

$$\boldsymbol{Q}_{h,d} = \boldsymbol{S}^{\mathrm{T}} \boldsymbol{W}_{h,d} \boldsymbol{S}, \quad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O}_{m\times 2p-m} \\ \boldsymbol{O}_{m\times 2p-m} & \boldsymbol{I}_m \end{bmatrix}, \tag{35}$$

being $\boldsymbol{W}_{h,d} \in \mathbb{R}^{2m\times 2m}$ a symmetric positive semi-definite weighting matrix. The rationale behind (34) is to continuously combine the robot's desired trajectory (i) with the human's desired one (iii) and, similarly, the virtual effort of the robot (ii) and of the human (iv) based on the respective weighting matrices. This means that by increasing $\boldsymbol{Q}_{h,d}$ and $\boldsymbol{R}_h$ with respect to $\boldsymbol{Q}_{r,d}$ the human intention overtakes the robot one (and vice-versa) according to an optimal formulation. In addition, the proposed approach is also suitable for an assistance task, whose aim is to minimize the human effort, by setting $\boldsymbol{Q}_{r,d} = \boldsymbol{O}_{2p}$.
In the above formulation, the human intention in terms of object orientation is not taken into account which means that robots have full control of it. Like in [40], this is motivated by the fact that the human is in charge of modifying the object's motion while robots autonomously and precisely control the orientation.

3.3.1 Designed strategy

The two-layer architecture depicted in Figure 2 is devised. The high-level is in charge of defining the virtual control input in (33) and, then, of the object reference trajectory $\boldsymbol{x}_v$ as the solution of the optimal problem in (34) based on human intentions, while the low-level is in charge of defining the local control inputs $\boldsymbol{u}_x$ in (4) to actually track the trajectory while achieving wrench regulation.

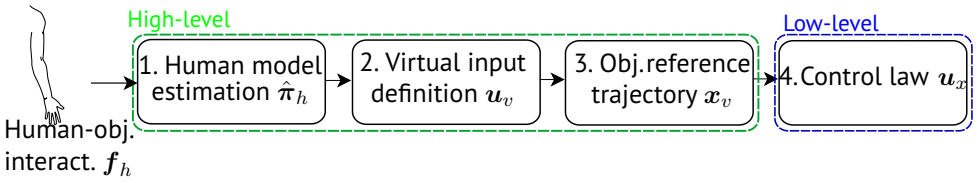

Figure 2: Two-layer architecture of the proposed solution.

Concerning the high-level, the formulation in (33) requires the virtual input $\boldsymbol{u}_v$ to be designed (block 2) so as

to optimize (34). By replacing the human model (11) in the cost function (34), the latter can be rewritten as

$$J = \frac{1}{2}\int_{t_0}^{+\infty} \Big((\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_r^d)^{\mathrm{T}} \boldsymbol{Q}_{r,d} (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{x}}_r^d) + \boldsymbol{u}_v^{\mathrm{T}} \boldsymbol{R}_v \boldsymbol{u}_v + (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{p}}_h^d)^{\mathrm{T}} \bar{\boldsymbol{Q}}_h (\bar{\boldsymbol{x}}_v - \bar{\boldsymbol{p}}_h^d) \Big) dt \tag{36}$$

with $\bar{\boldsymbol{Q}}_h = \boldsymbol{S}^{\mathrm{T}} \bar{\boldsymbol{W}}_{h,d} \boldsymbol{S}$ and

$$\bar{\boldsymbol{W}}_{h,d} = \boldsymbol{W}_{h,d} + \begin{bmatrix} \boldsymbol{K}_h^{\mathrm{T}} \boldsymbol{R}_h \boldsymbol{K}_h & \boldsymbol{K}_h^{\mathrm{T}} \boldsymbol{R}_h \boldsymbol{D}_h \\ \boldsymbol{D}_h^{\mathrm{T}} \boldsymbol{R}_h \boldsymbol{K}_h & \boldsymbol{D}_h^{\mathrm{T}} \boldsymbol{R}_h \boldsymbol{D}_h \end{bmatrix}. \tag{37}$$

It is worth noticing that, according to (37), the weight associated with the human desired motion dynamically adapts based on the human impedance parameters $\boldsymbol{K}_h$ and $\boldsymbol{D}_h$, e.g., when the human stiffens the arm (i.e., $\boldsymbol{K}_h$ increases) the corresponding quantities in $\bar{\boldsymbol{Q}}_h$ *automatically* increase as well.
The formulation in (36) and (37) shows the dependence of the objective function on the *unknown* human parameters $\boldsymbol{K}_h, \boldsymbol{D}_h, \boldsymbol{p}_h^d$, that is on $\boldsymbol{\pi}_h$ in (13). Therefore, an online estimation of these parameters is needed by exploiting the knowledge of human force and motion, i.e. $\boldsymbol{f}_h, \boldsymbol{p}_h, \dot{\boldsymbol{p}}_h$ (block 1 in Figure 2), and is then used to find $\boldsymbol{u}_v$ which minimizes (36). Afterward, the virtual end-effector trajectory $\boldsymbol{x}_v$ is computed according to the dynamics in (33) (block 3) and is adopted by the robot as the actual reference trajectory for the end-effector. Details are here omitted concerning the low-level control since a general distributed solution to this problem involving multiple cooperative robots is presented in Section 4.

3.3.2 Human parameters estimation

A Recursive Least Square (RLS) method with a forgetting factor is here proposed to estimate the human arm parameter $\boldsymbol{\pi}_h$ defined in (12) needed in (36). Let us consider that the measurements are acquired at each kT, with $k \in \mathbb{N}$ the discrete time index and $T \in \mathbb{R}$ the sampling time, and let us denote with $(\cdot)_k$ the corresponding quantity at time kT. The estimation error $\boldsymbol{e}_{h,k} \in \mathbb{R}^m$ is defined as

$$\boldsymbol{e}_{h,k} = \boldsymbol{f}_{h,k} - \boldsymbol{Y}_{h,k} \hat{\boldsymbol{\pi}}_{h,k-1} \tag{38}$$

where $\boldsymbol{Y}_{h,k}$ is the human regressor defined in (12) and $\hat{\boldsymbol{\pi}}_{h,k} \in \mathbb{R}^{3m}$ is the estimate of the unknown human parameters. The computation of (38) requires the knowledge of $\boldsymbol{f}_h$ which can be retrieved by using a wrench sensor. By leveraging [20], the parameters estimate is updated as

$$\hat{\boldsymbol{\pi}}_{h,k} = \hat{\boldsymbol{\pi}}_{h,k-1} + \boldsymbol{L}_k \boldsymbol{e}_{h,k} \tag{39}$$

where the matrices $\boldsymbol{L}_k \in \mathbb{R}^{3m \times m}$ and $\boldsymbol{P}_k \in \mathbb{R}^{3m \times 3m}$ are

$$\begin{aligned} \boldsymbol{L}_k &= \boldsymbol{P}_{k-1} \boldsymbol{Y}_{h,k}^{\mathrm{T}} (\lambda \boldsymbol{I}_m + \boldsymbol{Y}_{h,k} \boldsymbol{P}_{k-1} \boldsymbol{Y}_{h,k}^{\mathrm{T}})^{-1} \\ \boldsymbol{P}_k &= \frac{1}{\lambda} (\boldsymbol{I}_{3m} - \boldsymbol{L}_k \boldsymbol{Y}_{h,k}) \boldsymbol{P}_{k-1}, \end{aligned} \tag{40}$$

being $\lambda \in (0,\, 1]$ the forgetting factor which regulates the amount of data to be forgotten at each estimate update: the lower the forgetting factor, the lower the weight associated with past inputs. This implies that the lower the forgetting factor, the more the estimation can track changes in the parameters but the more the misadjustment and the possible instability [20]. Implementation details of (39) and (40) are not addressed

This project is funded by the European Union G.A No 101016906

here but a detailed discussion can be found in [33].
However, for the estimate $\hat{\boldsymbol{\pi}}_h$ in (39) to converge to the real values $\boldsymbol{\pi}_h$, the estimator input variables, i.e. $\boldsymbol{p}_h$ and $\dot{\boldsymbol{p}}_h$, have to satisfy the persistence of excitation condition for which the approximation proposed in [43] is here considered, that is the following condition should be verified

$$\|\boldsymbol{P}_{k-1}\boldsymbol{Y}_{h,k}\| > \alpha, \quad \forall k \text{ with } \alpha \in \mathbb{R}^+ \tag{41}$$

Moreover, by leveraging the results in [14], a finite-time interval with predetermined duration Δ, during which the input signals are excited, allows the parameters to be estimated with a certain tolerance. For further details, the reader is referred to [14] and [43].
Based on the above considerations, we devise a confidence index $\mathcal{I}_c$ to establish the reliability of the human parameters estimate. The basic idea is that the latter is assumed reliable if there exists a time interval for which the system is "sufficiently" excited and as long as the estimation error is below a certain threshold, which leads to the following index

$$\mathcal{I}_c(\hat{\boldsymbol{\pi}}_{h,k}) = \begin{cases} 1, \text{ if } \|\boldsymbol{e}_{h,k}\| < \bar{e} \wedge \exists\, k_1, k_2 \in \mathbb{N} : T|k_2 - k_1| \geq \Delta \wedge \forall\, k \in [k_1,\, k_2] \text{ eq. (41) holds} \\ 0, \text{ otherwise} \end{cases} \tag{42}$$

with $\bar{e} \in \mathbb{R}$ a positive constant representing the maximum allowed estimation error.
Finally, it is worth remarking that any other technique than RLS, as the approach in [4], might be used in block 1 of Figure 2.

3.3.3 Solution of the optimal LQT problem in Section 3.3.2

The confidence index $\mathcal{I}_c$ in (42) specifies whether the human parameter estimation is reliable or not (as in the initial transient phase). Therefore, the latter case needs to be explicitly handled in order to avoid undesired interaction behavior. In detail, when $\mathcal{I}_c(\hat{\boldsymbol{\pi}}_{h,k}) = 0$, the estimate $\hat{\boldsymbol{\pi}}_{h,k}$ is discarded, and full control of the system is given to the human operator (instead of minimizing (36)) by setting $\boldsymbol{u}_v$ in (33) as

$$\boldsymbol{u}_v = -\boldsymbol{D}_v\dot{\boldsymbol{x}}_v + \boldsymbol{\gamma}(\boldsymbol{h}_h)$$

which leads to the following virtual model

$$\boldsymbol{M}_v\ddot{\boldsymbol{x}}_v + \boldsymbol{D}_v\dot{\boldsymbol{x}}_v = \boldsymbol{\gamma}(\boldsymbol{h}_h) \tag{43}$$

where $\boldsymbol{D}_v \in \mathbb{R}^{p\times p}$ represents a damping matrix and $\boldsymbol{\gamma}(\cdot) : \mathbb{R}^p \to \mathbb{R}^p$ is a function of the human wrench possibly chosen as a deadzone function with threshold $t_h \in \mathbb{R}^+$, that is

$$\boldsymbol{\gamma}(\boldsymbol{h}_h) = \begin{cases} \boldsymbol{K}_\gamma\big(\boldsymbol{h}_h - t_h\dfrac{\boldsymbol{h}_h}{\|\boldsymbol{h}_h\|}\big) & \text{if } \|\boldsymbol{h}_h\| > t_h \\ \boldsymbol{0}_p & \text{if } \|\boldsymbol{h}_h\| \leq t_h \end{cases} \tag{44}$$

with $\boldsymbol{K}_\gamma \in \mathbb{R}^{p\times p}$ a positive definite matrix. Dynamics in (43) allows the human to move the object with the robots executing no autonomous tasks. On the contrary, when $\mathcal{I}_c(\hat{\boldsymbol{\pi}}_{h,k}) = 1$, the human estimate is exploited to minimize the cost in (36). To this aim, the following lemma holds true.

Lemma 1. *Let us consider a time-invariant linear system*

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{45}$$

with $\boldsymbol{x} \in \mathbb{R}^n$ *and* $\boldsymbol{u} \in \mathbb{R}^m$ *the state and the input of the system, and* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ *the dynamic and input matrices, respectively. Let us consider the following infinite-horizon cost function*

$$J = \frac{1}{2}\int_{t_0}^{\infty} \Big((\boldsymbol{x}-\boldsymbol{r}_1)^{\mathrm{T}} \boldsymbol{Q}_1 (\boldsymbol{x}-\boldsymbol{r}_1) + (\boldsymbol{x}-\boldsymbol{r}_2)^{\mathrm{T}} \boldsymbol{Q}_2 (\boldsymbol{x}-\boldsymbol{r}_2) + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u} \Big) dt \tag{46}$$

with $\boldsymbol{r}_1 \in \mathbb{R}^n$ *and* $\boldsymbol{r}_2 \in \mathbb{R}^n$ *reference signals,* $\boldsymbol{Q}_1 \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{Q}_2 \in \mathbb{R}^{n\times n}$ *symmetric semi-definite positive matrices and* $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ *a symmetric definite positive matrix. If the pair* $(\boldsymbol{A},\ \boldsymbol{B})$ *is reachable and the pair* $(\boldsymbol{A},\ [\boldsymbol{Q}_1\ \boldsymbol{Q}_2]^{\mathrm{T}})$ *is observable, then the optimal control input minimizing* (46) *is*

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} + \boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{v} \tag{47}$$

where $\boldsymbol{K} \in \mathbb{R}^{m\times n}$ *is a gain matrix defined as*

$$\begin{cases} \boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{T} \\ \boldsymbol{T}\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{T} - \boldsymbol{T}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{T} + \boldsymbol{Q}_1 + \boldsymbol{Q}_2 = \boldsymbol{O}_n \end{cases} \tag{48}$$

and $\boldsymbol{v} \in \mathbb{R}^n$ *is an auxiliary signal that evolves according to*

$$-\dot{\boldsymbol{v}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{Q}_1\boldsymbol{r}_1 + \boldsymbol{Q}_2\boldsymbol{r}_2 \tag{49}$$

Proof. The result follows by considering an aggregate weighting matrix $\bar{\boldsymbol{Q}} = [\boldsymbol{Q}_1\ \boldsymbol{Q}_2]^{\mathrm{T}} \in \mathbb{R}^{2n\times n}$ with the aggregate reference signal $\bar{\boldsymbol{r}} = [\boldsymbol{r}_1^{\mathrm{T}}\ \boldsymbol{r}_2^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2n}$ and by extending the reasoning presented in [27] for the LQT problem. □

By recalling (33), the state vector $\bar{\boldsymbol{x}}_v$ evolves according to

$$\dot{\bar{\boldsymbol{x}}}_v = \begin{bmatrix} \boldsymbol{O}_p & \boldsymbol{I}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p \end{bmatrix} \bar{\boldsymbol{x}}_v + \begin{bmatrix} \boldsymbol{O}_p \\ \boldsymbol{K}_M^{-1} \end{bmatrix} \boldsymbol{u}_v \tag{50}$$

By noticing that (36) and (50) are in the form of (46) and (45), respectively, and that reachability and observability conditions are fulfilled, Lemma 1 applies. The optimal virtual input $\boldsymbol{u}_v$ for (36) can be thus defined according to (47), by considering the estimated human parameters $\hat{\boldsymbol{\pi}}_h$ in place of the respective values in the cost function (36).

This project is funded by the European Union G.A No 101016906

4 Extension to distributed cooperative multi-robot systems

As already mentioned, the architecture presented in Figure 2 requires a low-level layer able to compute the control input $\boldsymbol{u}_x$ in Eq. (5) in order to track the generated reference. In this section, instead of presenting the control law for the case of a single-robot scenario, we present the solution for the multi-robot case, notably, in the absence of a central control unit. In detail, being multi-robot systems an important component of the CANOPIES project, the interaction strategies proposed in Section 3 have been extended to the multi-robot scenario where two or more mobile manipulators handle a common rigid object and the human operator interacts with the object (see Figure 4). The overall idea is to devise a two-layer architecture where, at the top layer, the human operator interacts with the common manipulated object (and not directly with the end-effector) according to any of the devised interaction strategies, while, the bottom layer is in charge of handling the distributed and multi-robot nature of the system by providing proper robots command inputs. Furthermore, since the manipulated object might be rigid, internal wrenches (wrenches that do not contribute to the object motion but only to its deformation) caused by uncertainties and disturbances need to be taken into account by the bottom layer.
However, since in a distributed framework each robot has only access to the local wrench $\boldsymbol{h}_i$ and the human operator interacts with the object co-manipulated by *all* the robots, there are global quantities to be estimated that are: (i) the measure of the force $\boldsymbol{f}_h$ exerted by the human operator and (ii) the contribution of each robot to the internal wrenches $\boldsymbol{h}_{int}$ in Eq. (52) which are required to compute the local control input $\boldsymbol{u}_x$ in Eq. (5). Therefore, the basic idea is to provide a decentralized estimation of the needed *global* information; moreover, concerning the human-robot (or human-object) interaction strategy, we propose a leader-follower approach in which the leader, based on the estimation of $\boldsymbol{f}_h$, is in charge of setting the high-level strategy according to any of the solutions presented in Section 3 and computes the object reference trajectory $\boldsymbol{x}_v, \dot{\boldsymbol{x}}_v$ accordingly, as in the centralized counterpart, while the followers estimate this trajectory via decentralized observers. These quantities feed a low-level control law which allows each robot to track the reference trajectory while regulating the internal wrenches. The resulting architecture is shown in Figure 3 and details are in the following.

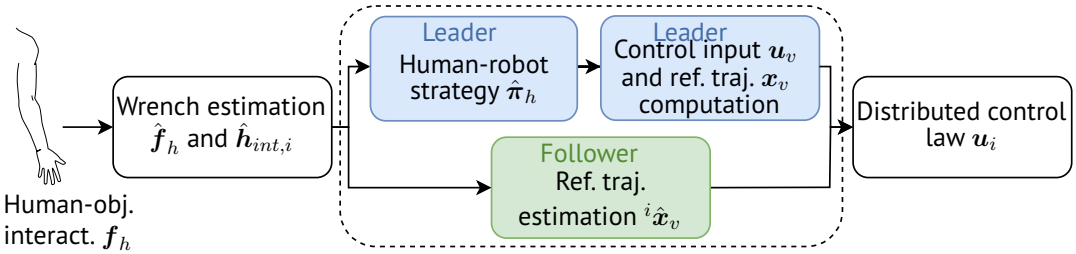

Figure 3: Distributed architecture implemented by robot i. The dashed box denotes that, depending on the role of the robot, different blocks are executed: blue blocks on the top in the case of leader, green blocks on the bottom in the case of follower.

Before presenting the devised architecture, the models of the object subject to robots and human wrenches and of the communication network are presented.

This project is funded by the European Union G.A No 101016906

4.1 Human-multi robot system modeling

Concerning this scenario, it is assumed that N mobile manipulators with model in (5) tightly grasp a rigid object and a human operator co-manipulates the same object as in Figure 4. Furthermore, robots' kinematics and dynamics in the operational space are referred to the human arm end point frame Σ_h (i.e., $\Sigma_{r,i} \equiv \Sigma_h$, $\forall\, i$). At the same way, the object dynamics is referred to the same frame Σ_h (i.e., $\Sigma_o \equiv \Sigma_h$). The above

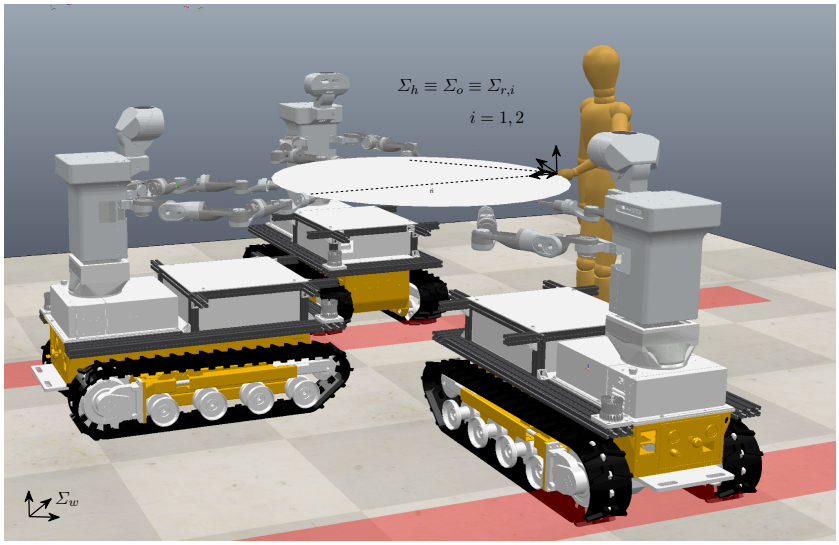

Figure 4: Representation of the considered system composed of multiple robots physically interacting with a human operator; the reference frames Σ_w, Σ_h, Σ_o and $\Sigma_{r,i}$ are reported. The cooperative human-multi-robot setup is made of three farming robots holding a common object together with a human operator.

assumption is not unrealistic in many practical scenarios since the geometry of the object is known or can be estimated beforehand.
Furthermore, it is necessary to model the overall system dynamics made of the robots and the object. To the scope, the object dynamics is

$$\boldsymbol{M}_o\ddot{\boldsymbol{x}}_o + \boldsymbol{C}_o(\boldsymbol{x}_o, \dot{\boldsymbol{x}}_o)\dot{\boldsymbol{x}}_o + \boldsymbol{g}_o = \sum_{i=1}^{N} \boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{h}_h \tag{51}$$

where $\boldsymbol{x}_o = \begin{bmatrix}\boldsymbol{p}_o^{\mathrm{T}} & \boldsymbol{\phi}_o^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^p$ represents the configuration in terms of position $\boldsymbol{p}_o$ and orientation $\boldsymbol{\phi}_o$ of the object reference frame Σ_o with respect to Σ_w, $\boldsymbol{M}_o \in \mathbb{R}^{p\times p}$ is the inertia matrix, $\boldsymbol{C} \in \mathbb{R}^{p\times p}$ is the matrix of centrifugal and Coriolis terms, $\boldsymbol{g}_o \in \mathbb{R}^p$ is the vector of the gravity terms, $\boldsymbol{G}_i \in \mathbb{R}^{p\times p}$ is the grasping matrix associated with the i th robot for which, since $\Sigma_h \equiv \Sigma_o \equiv \Sigma_{r,i}$, it holds $\boldsymbol{G}_i = \boldsymbol{I}_p\, \forall i$ and $\boldsymbol{h}_h = \boldsymbol{G}_h\boldsymbol{f}_h \in \mathbb{R}^p$ is the human wrench with $\boldsymbol{G}_h = [\boldsymbol{I}_m \quad \boldsymbol{O}_m]^{\mathrm{T}} \in \mathbb{R}^{p\times m}$.
By denoting with $\boldsymbol{h}$ the stacked vector of the interaction wrenches and with $\boldsymbol{G}$ the collective grasping matrix, i.e.

$$\boldsymbol{h} = [\boldsymbol{h}_1^{\mathrm{T}} \ \ldots \ \boldsymbol{h}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Np},\ \boldsymbol{G} = [\boldsymbol{G}_1 \ \ldots \ \boldsymbol{G}_N] \in \mathbb{R}^{p\times Np}$$

the interaction wrenches can be divided into external $\boldsymbol{h}_e \in \mathbb{R}^{Np}$ and internal $\boldsymbol{h}_{int} \in \mathbb{R}^{Np}$ components (see [34]), i.e., those that contribute and do not contribute, respectively, to the object motion for which it holds

$$\boldsymbol{h} = \boldsymbol{h}_e + \boldsymbol{h}_{int} = \boldsymbol{G}^{\dagger}\boldsymbol{G}\boldsymbol{h} + (\boldsymbol{I}_{Np} - \boldsymbol{G}^{\dagger}\boldsymbol{G})\boldsymbol{h}. \tag{52}$$

We assume that the kinematic and dynamic parameters of the manipulated object model are known. This hypothesis can be easily overcome by resorting to *ad-hoc* techniques that allow to distributively estimate these parameters by properly interacting with the object like in [35].

4.2 Communication modeling

Since we aim at devising a distributed framework, a brief review of graph theory is provided. The information exchange between the robots is modeled by a connectivity graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ where $\mathcal{V}$ is the set of vertices representing the robots, and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is the set of edges representing the communication links, that is the ith robot can send information to the jth robot if the condition $(i,\ j) \in \mathcal{E}$ holds. The set of neighbors of the ith robot from which it receives information is defined as $\mathcal{N}_i = \{j \in \mathcal{V} :\ (j, i) \in \mathcal{E}\}$. The topology of the graph is described in terms of the $(N \times N)$ Laplacian matrix $\boldsymbol{L} = \{l_{ij}\}$ such that $l_{ii} = \sum_{j=1, j\neq i}^{N} a_{ij}$ and $l_{ij} = -a_{ij}$ when $i \neq j$, being $a_{ij} = 1$ if the jth robot can send information to ith robot (i.e. $(j, i) \in \mathcal{E}$), and 0 otherwise. Moreover, by construction, zero is always a right eigenvalue of the Laplacian matrix with the corresponding right eigenvector $(N \times 1)$ vector of all ones [36]. Therefore, $\mathrm{rank}(\boldsymbol{L}) \leq N - 1$ where the equality holds when the graph is connected. It is assumed that the communication graph is undirected and connected, i.e. all the edges are bi-directional and any couple of vertices of the graph can always be connected by an undirected path. Moreover, the approach requires information to be explicitly exchanged between robots; however, this is not an issue in practical scenarios since all mobile manipulators have such communication capabilities (this applies to the CANOPIES Farming Robot as well).

4.3 Distributed human and internal wrench estimation

Let us denote with $\boldsymbol{h}_{int,i}$ the ith sub-vector of $\boldsymbol{h}_{int}$ (i.e., the contribution of the ith robot to the internal wrenches $\boldsymbol{h}_{int}$). The objective is to define a local observer in order for the ith robot to estimate $\boldsymbol{h}_{int,i}$ as well as the human force $\boldsymbol{f}_h$ (or equivalently $\boldsymbol{h}_h$). In view of (52) and by introducing the following selection matrix $\boldsymbol{\Gamma}_i$

$$\boldsymbol{\Gamma}_i = \big[\, \boldsymbol{O}_p \ \dots \ \underbrace{\boldsymbol{I}_p}_{robot\ i} \ \dots \ \boldsymbol{O}_p \,\big] \in \mathbb{R}^{p \times Np}$$

the vector $\boldsymbol{h}_{int,i}$ can be computed as

$$\begin{aligned} \boldsymbol{h}_{int,i} &= \boldsymbol{\Gamma}_i \boldsymbol{h}_{int} = \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}\boldsymbol{h} = \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{G}_i \boldsymbol{h}_i + \sum_{j \neq i} \boldsymbol{G}_j \boldsymbol{h}_j) \\ &= \underbrace{(\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i)\boldsymbol{h}_i}_{known} \ \underbrace{- \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \sum_{j=1, j\neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j}_{unknown} \end{aligned} \tag{53}$$

which is composed of a first local known term and a second unknown term depending on the wrenches exerted by the other robots. Thus, by considering the ith robot, the object model in (51) can be reformulated as

$$\boldsymbol{M}_o \ddot{\boldsymbol{x}}_o = \boldsymbol{G}_i \boldsymbol{h}_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j + \boldsymbol{h}_h - \boldsymbol{C}_o \dot{\boldsymbol{x}}_o - \boldsymbol{g}_o \tag{54}$$

Based on the approach in [32], the following residual vector $\boldsymbol{\theta}_i(t) \in \mathbb{R}^p$ is introduced

$$\boldsymbol{\theta}_i(t) = \boldsymbol{K}_\theta \int_{t_0}^{t} (\boldsymbol{\alpha} - \boldsymbol{G}_i \boldsymbol{h}_i - \boldsymbol{\theta}_i) d\tau + \boldsymbol{K}_\theta \boldsymbol{m}(t) \tag{55}$$

with $\boldsymbol{K}_\theta \in \mathbb{R}^{p\times p}$ a constant diagonal positive definite matrix, $\boldsymbol{m}(t) = \boldsymbol{M}_o \dot{\boldsymbol{x}}_o$ the generalized momentum of the object and $\boldsymbol{\alpha} = \boldsymbol{g}_o - \frac{1}{2}\dot{\boldsymbol{x}}_o^{\mathrm{T}} \frac{\partial \boldsymbol{M}_o}{\partial \boldsymbol{x}_o} \dot{\boldsymbol{x}}_o$. From [32], it follows

$$\dot{\boldsymbol{\theta}}_i(t) = -\boldsymbol{K}_\theta \boldsymbol{\theta}_i(t) + \boldsymbol{K}_\theta \Big(\sum\nolimits_{j=1, j\neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j + \boldsymbol{h}_h \Big) \tag{56}$$

which represents a low-pass filter whose bandwidth depends on the gain matrix $\boldsymbol{K}_\theta \in \mathbb{R}^{p\times p}$ and which leads to

$$\boldsymbol{\theta}_i(t) \approx \sum_{j=1, j\neq i}^{N} \boldsymbol{G}_j \boldsymbol{h}_j(t) + \boldsymbol{h}_h(t), \quad \forall t \tag{57}$$

Thus, by replacing (57) in (53), it holds

$$\boldsymbol{h}_{int,i} = \boldsymbol{\Gamma}_i (\boldsymbol{I}_{Np} - \boldsymbol{G}^\dagger \boldsymbol{G}) \boldsymbol{h} \approx (\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i) \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{\theta}_i - \boldsymbol{h}_h) \tag{58}$$

By virtue of (57), the approximation error made in (58) about the computation of $\boldsymbol{h}_{int,i}$ is

$$\boldsymbol{h}_{int,i} - \Big((\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i) \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{\theta}_i - \boldsymbol{h}_h) \Big) = \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger (\boldsymbol{\theta}_i - \sum_{j\neq i} \boldsymbol{G}_j \boldsymbol{h}_j - \boldsymbol{h}_h) \tag{59}$$

that is negligible only when the filter input (namely, $\sum_{j\neq i} \boldsymbol{G}_j \boldsymbol{h}_j + \boldsymbol{h}_h$) has a bandwidth much smaller than the cut-off frequency of the filter. Therefore, as made in [32] and citing works, a practical choice is to set $\boldsymbol{K}_\theta$ as high as possible subject to the potential digital implementation of the filter itself. Equation (58) makes evident that $\boldsymbol{h}_{int,i}$ can not be computed without the knowledge of $\boldsymbol{h}_h$ which is unknown as well. In this regard, let us define the following auxiliary signal $\boldsymbol{\xi}_i \in \mathbb{R}^p$

$$\boldsymbol{\xi}_i = (\boldsymbol{I}_p - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{G}_i) \boldsymbol{h}_i - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{\theta}_i \tag{60}$$

which, in light of (58) and by computing the term $\boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger$, can be rewritten as

$$\boldsymbol{\xi}_i \approx \boldsymbol{h}_{int,i} - \boldsymbol{\Gamma}_i \boldsymbol{G}^\dagger \boldsymbol{h}_h = \boldsymbol{h}_{int,i} - \frac{1}{N} \boldsymbol{h}_h \tag{61}$$

Note that neither $\boldsymbol{h}_{int,i}$ nor $\boldsymbol{h}_h$ are known in (61), but an approximation of $\boldsymbol{h}_{int,i} - \frac{1}{N}\boldsymbol{h}_h$ is provided by the right-hand side of (60), i.e. by the auxiliary variable $\boldsymbol{\xi}_i$. By resorting to the following lemma, the ith robot can compute the estimate ${}^i\hat{\boldsymbol{h}}_h \in \mathbb{R}^p$ of the human wrench $\boldsymbol{h}_h$ as well as the estimate $\hat{\boldsymbol{h}}_{int,i}$ of its contribution to the internal wrench.

Lemma 2. *Let each robot run the following observer*

$$\begin{cases} \dot{\boldsymbol{z}}_i = \gamma \sum\limits_{j\in\mathcal{N}_i} \operatorname{sign}({}^j\hat{\boldsymbol{h}}_h - {}^i\hat{\boldsymbol{h}}_h) \\ {}^i\hat{\boldsymbol{h}}_h = \boldsymbol{z}_i - N\boldsymbol{\xi}_i \\ \hat{\boldsymbol{h}}_{int,i} = \boldsymbol{\xi}_i + \dfrac{1}{N} {}^i\hat{\boldsymbol{h}}_h \end{cases} \tag{62}$$

where $\gamma \in \mathbb{R}$ is a positive constant, $\boldsymbol{z}_i \in \mathbb{R}^p$ is an auxiliary state, and $\mathrm{sign}(\cdot)$ is the component-wise signum function. Then,

- *${}^i\hat{\boldsymbol{h}}_h$ approaches $\boldsymbol{h}_h$ in finite time T_h, $\forall i = 1, ..., N$. Equivalently, by defining the estimation error ${}^i\tilde{\boldsymbol{h}}_h = \boldsymbol{h}_h - {}^i\hat{\boldsymbol{h}}_h$, it approaches the origin in finite time T_h.*
- *$\hat{\boldsymbol{h}}_{int,i}$ approaches $\boldsymbol{h}_{int,i}$ in finite time T_h, $\forall i = 1, ..., N$. Equivalently, by defining the estimation error $\tilde{\boldsymbol{h}}_{int,i} = \boldsymbol{h}_{int,i} - \hat{\boldsymbol{h}}_{int,i}$, it approaches the origin in finite time.*

Proof. By leveraging the same reasoning as in [8], it can be proved that the estimate ${}^i\hat{\boldsymbol{h}}_h$ converges to the average of $-N\boldsymbol{\xi}_i$ in finite-time T_h. Thus, by virtue of (61) and by recalling that, by construction, it holds $\sum_{i=1}^N \boldsymbol{h}_{int,i} = \boldsymbol{0}_p$, it follows ${}^i\hat{\boldsymbol{h}}_h \to -\frac{1}{N}\sum_{i=1}^N \boldsymbol{\xi}_i \approx \boldsymbol{h}_h$ or, equivalently, ${}^i\tilde{\boldsymbol{h}}_h \to \boldsymbol{0}_p$. Therefore, based on (61) and (62), the ith robot can also estimate its contribution to the internal wrenches $\boldsymbol{h}_{int,i}$ as follows

$$\hat{\boldsymbol{h}}_{int,i} = \boldsymbol{\xi}_i + \frac{1}{N}{}^i\hat{\boldsymbol{h}}_h \tag{63}$$

Since ${}^i\tilde{\boldsymbol{h}}_h \to 0$, from (61) it also holds

$$\hat{\boldsymbol{h}}_{int,i} \to \boldsymbol{\xi}_i + \frac{1}{N}\boldsymbol{h}_h = \boldsymbol{h}_{int,i} \tag{64}$$

This completes the proof. □

Finally, as consequence of the lemma above, the wrench contribution of robot i to the object motion can be estimated as

$$\boldsymbol{h}_i - \hat{\boldsymbol{h}}_{int,i} = \boldsymbol{h}_i - \boldsymbol{\xi}_i - \frac{1}{N}{}^i\hat{\boldsymbol{h}}_h = \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{\theta}_i - \frac{1}{N}{}^i\hat{\boldsymbol{h}}_h \tag{65}$$

which converges to the actual value as well.

4.4 Generation and estimation of the object trajectory $\boldsymbol{x}_v$

As per the architecture in Figure 3, depending on the role of the ith robot, different control inputs are designed.

4.4.1 Leader robot

The leader robot is in charge of estimating the human model (Section 3.3.2) based on the estimate made as in (62) and defining the virtual input $\boldsymbol{u}_v$ in (33) (Section 3.3.3), from which the object reference trajectory is then derived as in (50).

4.4.2 Follower robot

In order for the followers to estimate the reference trajectory, the solution in [16] is exploited. Let us denote with $\boldsymbol{\chi}_v$ the stacked vector of the desired trajectory, that is $\boldsymbol{\chi}_v = \left[\boldsymbol{x}_v{}^{\mathrm{T}}\ \dot{\boldsymbol{x}}_v{}^{\mathrm{T}}\ \ddot{\boldsymbol{x}}_v{}^{\mathrm{T}}\right]^T \in \mathbb{R}^{3p}$ and with ${}^i\hat{\boldsymbol{\chi}}_v$ the respective estimate made by the follower robot i. The following distributed observer is adopted by the followers

$$\begin{cases} {}^i\dot{\hat{\boldsymbol{\chi}}}_v = \boldsymbol{A}_\chi\,{}^i\hat{\boldsymbol{\chi}}_v - \mu_{v,1}\boldsymbol{B}_\chi\boldsymbol{B}_\chi^{\mathrm{T}}\boldsymbol{P}_\chi^{-1}\,{}^i\hat{\boldsymbol{\nu}}_v - \mu_{v,2}\mathrm{sign}\left(\boldsymbol{P}_\chi^{-1}\,{}^i\hat{\boldsymbol{\nu}}_v\right) \\ {}^i\hat{\boldsymbol{\nu}}_v = \sum_{j\in\mathcal{N}_i,\, j\neq l}\left({}^i\hat{\boldsymbol{\chi}}_v - {}^j\hat{\boldsymbol{\chi}}_v\right) + b_i\left({}^i\hat{\boldsymbol{\chi}}_v - \boldsymbol{\chi}_v\right) \end{cases} \tag{66}$$

This project is funded by the European Union G.A No 101016906

where l is the index associated with the leader robot, i belongs to the set of the follower robots, b_i is 1 if the leader belongs to $\mathcal{N}_i$ and 0 otherwise, $\mu_{v,1}, \mu_{v,2} \in \mathbb{R}$ are positive gains, $\boldsymbol{A}_\chi \in \mathbb{R}^{3p\times 3p}$ and $\boldsymbol{B}_\chi \in \mathbb{R}^{3p}$ are matrices selected as

$$\boldsymbol{A}_\chi = \begin{bmatrix} \boldsymbol{O}_p & \boldsymbol{I}_p & \boldsymbol{O}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p & \boldsymbol{I}_p \\ \boldsymbol{O}_p & \boldsymbol{O}_p & \boldsymbol{O}_p \end{bmatrix}, \quad \boldsymbol{B}_\chi = \begin{bmatrix} \boldsymbol{O}_p & \boldsymbol{O}_p & \boldsymbol{I}_p \end{bmatrix}^T$$

and $\boldsymbol{P}_\chi \in \mathbb{R}^{3p\times 3p}$ is a positive definite matrix. By following the same reasoning as in [16], it can be proved that, under a proper selection of gains, the observer in (66) guarantees the finite-time leader tracking, i.e. it holds in finite-time T_χ

$$\|{}^i\hat{\boldsymbol{\chi}}_v(t) - \boldsymbol{\chi}_v(t)\| = 0 \,\forall i \neq l,\ t \geq T_\chi \tag{67}$$

4.5 Distributed low-level control

By always considering the scenario in Figure 4 and the architecture in Figure 3, the control input $\boldsymbol{u}_{x,i}$ in Eq. (8) to cooperatively track the object reference trajectory and control the internal stresses, in the hypothesis of uncertain dynamics (5), is here devised. Let us denote with $\boldsymbol{e}_{int,i}$ the internal wrench error vector of robot i, i.e. $\boldsymbol{e}_{int,i} = \boldsymbol{h}^d_{int,i} - \boldsymbol{h}_{int,i} \in \mathbb{R}^p$, being $\boldsymbol{h}^d_{int,i} = \boldsymbol{\Gamma}_i \boldsymbol{h}^d_{int}$, and let us introduce the following integral error $\Delta\boldsymbol{u}_{f,i} \in \mathbb{R}^p$

$$\Delta\boldsymbol{u}_{f,i}(t) = k_f \int_{t_0}^{t} \boldsymbol{e}_{int,i}\, d\tau \tag{68}$$

with $k_f \in \mathbb{R}$ a positive scalar regulating the internal wrench control, and the error vector $\boldsymbol{e}_{x,i} \in \mathbb{R}^p$

$$\boldsymbol{e}_{x,i}(t) = ({}^i\hat{\boldsymbol{x}}_v - \boldsymbol{x}_i) + k_c \int_{t_0}^{t} \sum\nolimits_{j\in\mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_i) d\tau \tag{69}$$

with $k_c \in \mathbb{R}$ a positive constant, which is composed of a first term for the tracking of the object trajectory and a second synchronization term aimed at limiting the internal forces during the transient phases or due to unmodeled dynamics. Finally, let us introduce the following auxiliary variables

$$\begin{cases} \boldsymbol{\zeta}_i = {}^i\dot{\hat{\boldsymbol{x}}}_v + k_c \sum\nolimits_{j\in\mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_i) + k_p \boldsymbol{e}_{x,i} \quad \in \mathbb{R}^p \\ \tilde{\boldsymbol{\zeta}}_i = \boldsymbol{\zeta}_i - \dot{\boldsymbol{x}}_i = \dot{\boldsymbol{e}}_{x,i} + k_p \boldsymbol{e}_{x,i} \quad \in \mathbb{R}^p \\ \boldsymbol{\rho}_i = \boldsymbol{\zeta}_i + \Delta\boldsymbol{u}_{f,i} \quad \in \mathbb{R}^p \\ \boldsymbol{s}_i = \tilde{\boldsymbol{\zeta}}_i + \Delta\boldsymbol{u}_{f,i} \quad \in \mathbb{R}^p \end{cases} \tag{70}$$

with $k_p \in \mathbb{R}$ a positive constant and $\hat{\Delta}\boldsymbol{u}_{f,i}$ the estimate of $\Delta\boldsymbol{u}_{f,i}(t)$ made by robot i as

$$\hat{\Delta}\boldsymbol{u}_{f,i}(t) = k_f \int_{t_0}^{t} \hat{\boldsymbol{e}}_{int,i}\, d\tau \tag{71}$$

where $\hat{\boldsymbol{e}}_{int,i} = \boldsymbol{h}^d_{int,i} - \hat{\boldsymbol{h}}_{int,i}$ which takes into account that the internal wrench $\boldsymbol{h}_{int,i}$ is not known but only locally estimated as in (64). By extending the approach in [34], the following control law is proposed

$$\boldsymbol{u}_i = \hat{\boldsymbol{M}}_i \dot{\boldsymbol{\rho}}_i + \hat{\boldsymbol{C}}_i \boldsymbol{\rho}_i + \hat{\boldsymbol{\eta}}_i + \boldsymbol{K}_s \boldsymbol{s}_i + \Delta\boldsymbol{u}_i = \boldsymbol{Y}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\hat{\boldsymbol{\pi}}_i + \boldsymbol{K}_s \boldsymbol{s}_i + \Delta\boldsymbol{u}_i \tag{72}$$

being $\boldsymbol{K}_s \in \mathbb{R}^{p\times p}$ a positive definite matrix and $\Delta\boldsymbol{u}_{x,i} \in \mathbb{R}^p$ selected as

$$\begin{aligned}\Delta\boldsymbol{u}_{x,i} =& \kappa_i(t)\boldsymbol{s}_i + \boldsymbol{h}^d_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i} + \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{\theta}_i - \frac{1}{N}{}^i\hat{\boldsymbol{h}}_h \\ =& \underbrace{\kappa_i(t)\boldsymbol{s}_i}_{a} + \underbrace{\boldsymbol{h}^d_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i}}_{b} + \underbrace{\boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{G}_i\boldsymbol{h}_i + \boldsymbol{\Gamma}_i\boldsymbol{G}^\dagger\boldsymbol{\theta}_i - \frac{1}{N}\boldsymbol{h}_h + \frac{1}{N}{}^i\tilde{\boldsymbol{h}}_h}_{c}\end{aligned} \tag{73}$$

with $\kappa_i(t) \in \mathbb{R}$ a positive time-varying gain, and where

- a) is a robust term to guarantee convergence of internal forces in the transient phase and despite model uncertainties;
- b) is a force feed-forward and integral error contribution;
- c) represents, based on (65), the compensation of the contribution to the external generalized forces made by robot i on the object.

Moreover, the dynamic parameters of robot i are updated as

$$\dot{\hat{\boldsymbol{\pi}}}_i = \boldsymbol{K}_\pi^{-1}\boldsymbol{Y}_i^{\mathrm{T}}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)^{\mathrm{T}}\boldsymbol{s}_i \tag{74}$$

with $\boldsymbol{K}_\pi \in \mathbb{R}^{n_{\pi_i}\times n_{\pi_i}}$ a positive definite matrix.
The analysis of control law $\boldsymbol{u}_{x,i}$ is provided in the theorem.

Theorem 1. *Let us consider N robots with dynamics in* (4) *for which an estimate is known as in* (8). *Let us consider the observer in* (66)*, the control law in* (72) *and the parameters update law in* (74). *Then, it asymptotically holds $\boldsymbol{x}_i \to \boldsymbol{x}_v$ and $\boldsymbol{e}_{int,i} \to \boldsymbol{0}_p\ \forall i = 1, ..., N$.*

Proof. The proof is provided in the Appendix 8.1. □

This project is funded by the European Union G.A No 101016906

5 Neural-network based contact detection and classification

5.1 Introduction and motivation

Collaboration between humans and robots can envisage the case of pure workspace sharing, in which humans and robots work on autonomous or coordinated tasks without the need for physical contact or voluntary exchange of forces to achieve a common goal. In both cases, human safety is the primary issue. However, while in the workspace sharing case, human safety can be achieved by delivering proper collision avoidance strategies that prevent unsafe contacts [26, 31], in the other case, safety requirements are more challenging to meet being the exchange of forces envisaged by the task itself (like in the case of kinesthetic teaching [24]). Indeed, in the latter case not only *intentional* contacts but also *accidental* collisions should be considered. The two cases require the robot to adopt different reaction strategies. In particular, in the case of intentional contact, admittance or impedance control are traditionally employed, which confer a compliant behavior to the robot structure through a mass-spring-damper model [13, 44]. On the contrary, accidental collisions are representative of unexpected and dangerous situations, which need to be promptly detected and handled by avoidance strategies like in [26]. However, to undertake a suitable reaction, it is first required to detect the occurrence of a contact and, then, to recognize whether it is intentional or accidental [19]. Contact detection solutions often exploit the dynamical model of the robot and joint torque sensors, as in [32], or motor currents as in [17]. Other approaches, like the one in [25], leverage the different frequency characteristics of accidental and intentional contacts to achieve classification in the frequency domain. Additional solutions might also leverage proper artificial skins for robots as in [9, 1].

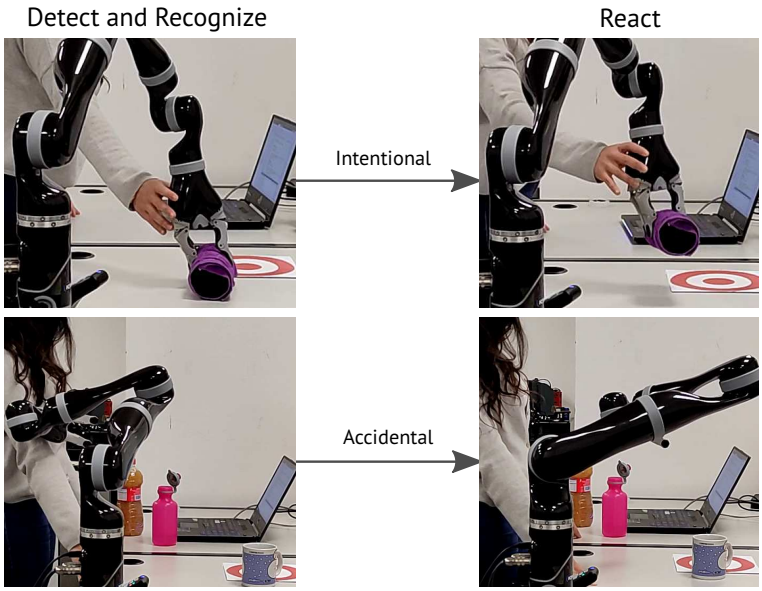

Figure 5: Framework example: the human contact is detected and classified (on the left) and the system reacts according to its nature (on the right), i.e. intentional (top row) or accidental (bottom row).

Classical approaches, like the ones cited above, require thresholds to be manually tuned to take into account sensor noise and different contact types, which results in poor flexibility and robustness of the overall sys-

This project is funded by the European Union G.A No 101016906

tem. For this reason, data-driven approaches have been devised in recent years for contact detection and classification, due to their flexibility and capability of handling the non-linearity and variety of human-robot contact. In [41, 42], Neural Networks (NNs) are used to detect sole accidental collisions on the basis of data coming from joint torque sensors. The same objective is achieved in [21] by using a deep learning approach which requires the tuning of a moving time window. Finally, the study in [5] uses NNs to detect intentional contacts as well, but limited to the upper and lower parts of the robot.
In this part of the Deliverable, we aim to devise an architecture where the behavior of the robot is adjusted according to the type of interaction between human and robot and the robot task. The robot is endowed with the ability to perform tasks autonomously, while the human operator is allowed to work side-by-side with the robot to perform additional tasks or to change the robot configuration by physically interacting with it. Therefore, similarly to the papers cited above and as depicted in Figure 5, both accidental and intentional contacts might arise and need to be detected and classified; this is achieved by following two different approaches, depending on the available sensors. Based on the output of these classifiers, the human-robot interaction strategies described in Section 3 are employed in the case of intentional contacts.

5.2 Problem formulation and possible solutions overview

Formally, the following problem is addressed in this section.

Problem 1. *Consider the robot dynamics in* (3) *and a task represented by a desired end effector trajectory* $\boldsymbol{x}_d(t)$ *(*$\dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t)$*) which can possibly involve interaction* $\boldsymbol{h}_\mathcal{T}$ *with the environment. Assume that a human operator can intentionally or accidentally physically interact with any point of the robot structure, i.e.* $\|\boldsymbol{\tau}_h\| > 0$*. The aim is to design a strategy enabling the robot to detect and classify possible contacts with the human and to undertake proper reaction behaviors, while complying with possible robot constraints.*

In the following, we propose two different approaches to solve Problem 1, depending on the sensor suite that the specific robot is equipped with. In case torque sensors for the manipulator's joints are available, we propose the architecture described in Section 5.2.1 that leverages the knowledge of the dynamic model of the manipulator and the torque sensors measurements to estimate the wrench exerted by the human as well as the contact point localization along the robot structure. This information is then used for the type of contact classification. In case only a wrench sensor on the end-effector is available, without any torque sensor on the joints, we propose the architecture described in Section 5.2.2, which classifies the type of contact that occurs only on the end-effector. In both the proposed solutions, the output of the classification procedure is used to change the specific behavior of the robot among the basic human-robot interaction strategies described in Section 3, following a Finite-State-Machine algorithm.
It is worth noticing that the design of the solution in Section 5.2.2 is needed to be used on the CANOPIES robot, which, at the time of the submission of the present Deliverable, is not equipped with joint torque sensors.

5.2.1 Contact detection based on joint torque sensors

The modular architecture shown in Figure 6 is proposed as the first solution to the problem above. In detail, a *Contact Detection module* is in charge of assessing whether or not a contact with a human operator is taking place by considering that the robot might be interacting with the environment. Next, a *Contact Type Recognition module* establishes the nature of the human interaction when contact is detected. To this aim,

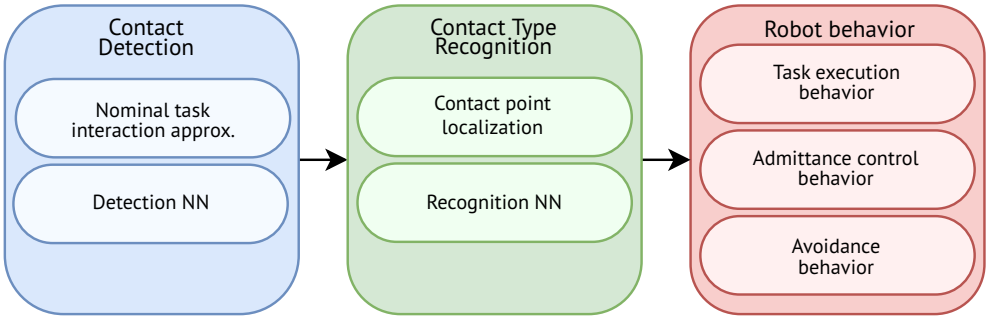

Figure 6: Overview of the proposed approach to detect human contact (in blue), recognize its type (in green) and react accordingly (in red).

the contact point along the robot structure is localized, and the estimated wrench exerted by the human is evaluated according to Eq. (78) to evaluate whether or not the contact is intentional. Finally, a *Robot Behavior module* determines the robot virtual input by applying one of the basic human-robot interaction strategies described in Section 3, considering the type of contact (if any) and the current task.

5.2.1.1 Human force and torque estimation. In order to detect and recognize the type of contact, an estimation of the human force and torque is needed. To this aim, ad-hoc hardware devices can be integrated, as for example in [1], or momentum-based observers that only rely on on-board current/torque sensors can be leveraged [32]. For the sake of generality, the latter approach is pursued in the following.
Let us introduce the residual vector $\boldsymbol{r}(t) \in \mathbb{R}^6$ defined as

$$\boldsymbol{r}(t) = \boldsymbol{K}\left(\int_0^t (\boldsymbol{\alpha} - \boldsymbol{\tau} - \boldsymbol{r})d\tau + \boldsymbol{m}(t)\right) \tag{75}$$

with $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ a constant diagonal positive definite matrix, $\boldsymbol{m}(t) = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$ the generalized momentum of the manipulator and $\boldsymbol{\alpha} = \boldsymbol{g} - 1/2\,\dot{\boldsymbol{q}}^{\mathrm{T}}(\partial \boldsymbol{M}/\partial \boldsymbol{q})\dot{\boldsymbol{q}}$. By following the steps in [32] and for high values of $\boldsymbol{K}$, the following holds

$$\boldsymbol{r}(t) \approx (\boldsymbol{\tau}_{\mathcal{T}} + \boldsymbol{\tau}_h). \tag{76}$$

Let us assume that an estimate $\hat{\boldsymbol{\tau}}_{\mathcal{T}}$ of $\boldsymbol{\tau}_{\mathcal{T}}$ in (3) is available and that, in the case of human interaction, the contact point P along the robot structure is estimated. From (76), an estimate $\hat{\boldsymbol{\tau}}_h$ of $\boldsymbol{\tau}_h$ is obtained as

$$\hat{\boldsymbol{\tau}}_h(t) = \boldsymbol{r}(t) - \hat{\boldsymbol{\tau}}_{\mathcal{T}} \tag{77}$$

based on which the components of $\boldsymbol{h}_h(t)$ not belonging to the null space of $\boldsymbol{J}_P^{\mathrm{T}}$ can be retrieved by following, for example, the approach in [12]

$$\hat{\boldsymbol{h}}_h(t) = \left(\boldsymbol{J}_P\left(\boldsymbol{q}(t)\right)^{\mathrm{T}}\right)^{\dagger} \hat{\boldsymbol{\tau}}_h(t). \tag{78}$$

5.2.1.2 Nominal task interaction approximation. In order to assess whether or not contact with a human operator is occurring, we define an approximation $\hat{\boldsymbol{\tau}}_{\mathcal{T}}$ (blue block of Figure 6) of the expected external torque $\boldsymbol{\tau}_{\mathcal{T}}$, arising from the interaction with the environment for task $\mathcal{T}$. Indeed, based on this estimation

and on the estimated overall external torque in (76), the estimated torques induced by the *human* interaction $\hat{\boldsymbol{\tau}}_h$ can be derived.
We propose to use Gaussian Mixture Models to approximate the nominal wrench profile. In this way, a *rough* definition of the expected nominal profile can be derived on the basis of *few* demonstrations of the nominal interaction with the environment. Let us introduce the extended vector of the nominal wrench of task $\mathcal{T}$ at time t as $\boldsymbol{\zeta}_{\mathcal{T},t} = [t \;\; \boldsymbol{h}_{\mathcal{T},n}(t)^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^7$. This can be probabilistically modeled as a mixture of $K_{\mathcal{T}}$ Gaussian distributions as follows

$$p(\boldsymbol{\zeta}_{\mathcal{T},t}) = \sum_{k=1}^{K_{\mathcal{T}}} \pi_{\mathcal{T},k} \, \mathcal{N}(\boldsymbol{\zeta}_{\mathcal{T},t} | \boldsymbol{\mu}_{\mathcal{T},k}, \boldsymbol{\Sigma}_{\mathcal{T},k}) \tag{79}$$

where $\mathcal{N}(\boldsymbol{\zeta}_{\mathcal{T},t} | \boldsymbol{\mu}_{\mathcal{T},k}, \boldsymbol{\Sigma}_{\mathcal{T},k})$ is the Gaussian distribution with mean $\boldsymbol{\mu}_{\mathcal{T},k} \in \mathbb{R}^7$ and covariance $\boldsymbol{\Sigma}_{\mathcal{T},k} \in \mathbb{R}^{7\times 7}$ and $\pi_{\mathcal{T},k}$ is the k th mixture coefficient such that $\pi_{\mathcal{T},k} \in [0, 1]$ and $\sum_{k=1}^{K_{\mathcal{T}}} \pi_{\mathcal{T},k} = 1$.
The Expectation-Maximization (EM) algorithm [39] is then leveraged to estimate the GMM parameters $\boldsymbol{\mu}_{\mathcal{T},k}$, $\boldsymbol{\Sigma}_{\mathcal{T},k}$ and $\pi_{\mathcal{T},k}$ $\forall k \in \{1, \ldots, K_{\mathcal{T}}\}$ based on demonstrations of $\boldsymbol{\zeta}_{\mathcal{T},t}$, $\forall t$. To gather the latter, the task $\mathcal{T}$ is executed few times in the absence of human wrench, and the residual vector $\boldsymbol{r}(t)$ in (75) is recorded; the latter coincides with the estimate of $\boldsymbol{\tau}_{\mathcal{T}}(t)$ when $\boldsymbol{\tau}_h = \mathbf{0}_n$, being $\mathbf{0}_n$ the null vector of n elements. Each datapoint $\boldsymbol{h}_{\mathcal{T},n}(t)$ in the demonstrations of $\boldsymbol{\zeta}_{\mathcal{T},t}$ is then obtained as

$$\boldsymbol{h}_{\mathcal{T},n}(t) = \left(\boldsymbol{J}\left(\boldsymbol{q}(t)\right)^{\mathrm{T}}\right)^{\dagger} \boldsymbol{r}(t). \tag{80}$$

The choice of approximating the expected wrench $\boldsymbol{h}_{\mathcal{T},n}$ instead of the expected external torque $\boldsymbol{\tau}_{\mathcal{T}}$ enables the proposed methodology to be independent from the joint space configuration while executing the task.
Finally, Gaussian Mixture Regression, as done for instance in [7], is used to get the probabilistic model of $\boldsymbol{h}_{\mathcal{T},n}(t)$ at each time t, with mean $\hat{\boldsymbol{h}}_{\mathcal{T},n}(t) = \mathbb{E}(\boldsymbol{h}_{\mathcal{T},n}(t))$. The estimate $\hat{\boldsymbol{\tau}}_{\mathcal{T}}(t)$ in eq. (77) is then achieved, i.e. $\hat{\boldsymbol{\tau}}_{\mathcal{T}}(t) = \boldsymbol{J}(\boldsymbol{q}(t))^{\mathrm{T}} \hat{\boldsymbol{h}}_{\mathcal{T},n}(t)$.

5.2.1.3 Detection and recognition classifiers. As mentioned above, the estimated torques $\hat{\boldsymbol{\tau}}_{\mathcal{T}}$ are exploited to compute the human torques $\hat{\boldsymbol{\tau}}_h$. Based on the latter, two RNNs are trained to determine the presence of a human contact and its type. In detail, the first network, referred to as *detection NN*, outputs the information on whether or not human contact is occurring, while the second network, referred to as *recognition NN*, outputs the information on whether the contact is intentional or accidental. In light of the binary classification problems, the two networks are trained to minimize the log loss. The same architecture is devised for both the networks, and it comprises: *(i)* a recurrent Long Short Term Memory layer that enables to learn the long-term dependencies between data samples in the time series, *(ii)* a fully connected layer, *(iii)* a logistic activation layer that outputs the prediction.
The training dataset of the detection NN consists of labeled time series representing the norm of the estimated human torque, i.e. $\|\hat{\boldsymbol{\tau}}_h(t)\|$ in eq. (77), while estimated human wrenches are considered for the recognition NN, i.e. the dataset consists of the datapoints $\|\hat{\boldsymbol{h}}_h(t)\|$ in eq. (78). The use of norm values allows to take into account the intensity and variation of the human interaction, ignoring the directions in which the interaction occurs. The datasets were collected by voluntarily and accidentally interacting with different points along the robot structure during free space motion (i.e. with $\|\boldsymbol{\tau}_{\mathcal{T}}\| = 0$) and recording the labeled respective quantities. Note that the datasets only contain data acquired with no environment interaction; however, the proposed approach is able to tackle the case of $\|\boldsymbol{\tau}_{\mathcal{T}}\| > 0$ thanks to the GMM modeling and without wrench sensors mounted on the wrist. In this way, an arbitrary number of tasks can be performed

that require interaction with the environment without the need to re-train the networks and adapt their datasets. Finally, note that the generalization capabilities of the networks allow tackling without manually tuned thresholds, noisy measures, model uncertainties in (3), and task interaction estimation inaccuracies, which generally leads to not accurate human torque estimation, i.e. it generally holds $||\hat{\boldsymbol{\tau}}_h|| > 0$ even if no human-robot contact is happening.

5.2.1.4 Contact point localization. We foresee the presence of a procedure to localize the contact point P, needed to estimate the human wrench in Eq. (78) used by the recognition classifier and the behavior module. Not to limit the possible approaches and since we do not aim at improving localization techniques, we do not fix a specific procedure, but any appropriate solution can be applied to the purpose. As an example, an external vision algorithm can be integrated which tracks the human motion and determines the contact point. Alternatively, different ad-hoc sensors, such as IMU sensors [45], can be integrated for the purpose.

5.2.2 Contact detection by wrist wrench sensors

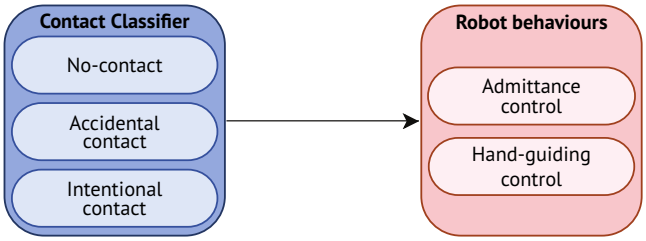

Figure 7: Overview of the approach proposed for robots provided only with a wrench sensor at the end-effector to detect human contacts, identify their type (in blue), and react appropriately to them (in red).

The modular architecture shown in Figure 7 is proposed as a second solution to Problem 1 for those cases where joint torque sensors are not available but, instead, wrist wrench sensor is. In detail, a *detection and recognition module* is in charge of both evaluating whether or not a contact with a human operator has occurred and establishing whether it has been intentional or accidental. Finally, as in the previous solution, a *behavior module* is responsible for generating the control input, taking into account both the type of contact and the given task. Details about the detection and recognition module are given in the following.

5.2.2.1 Detection and recognition classifier. In order to establish if a contact has occurred or not and to classify it as intentional or accidental, we trained a cubic KNN (K-Nearest-Neighbors), which is a KNN classifier having a cubic distance as a distance metric. The cubic distance is a Minkowski distance, defined as:

$$d = \left(\sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{y}_i|^{\beta} \right)^{\frac{1}{\beta}} \tag{81}$$

with $\beta = 3$, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the position vectors of two generic points of the $\mathbb{R}^n$ and β is a parameter whose variation allows one to get some of the most common distance metrics, such as the Manhattan

This project is funded by the European Union G.A No 101016906

distance ($\beta = 1$), the Euclidean distance ($\beta = 2$), the Cubic distance ($\beta = 3$) and the Chebyshev distance ($\beta = \infty$).
In order to prevent an overfitting of the model, its training has been performed via a validation scheme. In detail, it has been performed with the k-*fold cross-validation*, which involves the sub-division of the dataset in k-sets or folds and the iteration of the training and validation processes. The training and validation processes have to be iterated for k-time using one fold as validation set and the others as training set. The fold used as validation set has to be changed at each iteration. This implies that, at the i-th iteration, the model has to be trained using the i-th fold as validation set and the others as training set. At the $(i + 1)$-th iteration, instead, it has to be trained using the $(i + 1)$-th fold as validation set and the others as training set.
The dataset used for training and validating the classifier consists of multiple sets of features, which have been extracted from two sets of time series representing the time evolution of the norm of the linear forces detected by a wrench sensor mounted on the manipulator's wrist. In detail, the dataset generation has been conducted as follows: first, two sets of time series representing the time evolution of the norm of the forces measured by the wrench sensor have been collected. The first set has been acquired by voluntarily interacting with the robot, while the second one has been acquired by emulating some accidental interactions with the robot. It is worth underlining that, also in this case, the data have been acquired by avoiding any type of interaction between the robot and the environment. At the end of the acquisition process, each time series has been split into multiple time windows, which have been properly labelled.
Finally, a set of features has been extracted from each window, and each one has been labelled with the same label given to the window from which it has been extracted. The features extracted from the windowed signals are:

- average value;
- average and maximum value of the time derivative;
- squared sum of data below the 25 and 75 percentile;
- maximum signal frequency;
- the sum of the amplitude spectrum components below 0.9 Hz and above 1 Hz.

6 Simulation and experimental validation

In this section, we show experimental and simulative results obtained while testing the algorithms described in Section 3 and Section 5. All the experiments in this section have been performed in an indoor environment, more specifically in the LAI Robotics laboratory of the University of Cassino and Southern Lazio. Videos of the described experiments are included as multimedia material for the present Deliverable in Section 9.
More in detail, at first we show results aimed at validating the implemented basic human-robot interaction strategies described in Section 3.1 (admittance control), Section 3.2 (hand-guiding control), and Sections 3.3 and 4 (shared control in a multi-robot distributed setting); then, we show experimental results of a control architecture that makes use of the human-robot interaction strategies described before in a cooperative harvesting scenario; finally, we present experimental results obtained by applying the devised contact classification strategies on a Kinova Jaco^2 robot (for the case of joint torque sensors available as per Section 5.2.1) and on the CANOPIES farming robot (for the case of wrench wrist sensor as per Section 5.2.2).

6.1 Basic human-robot interaction strategies

6.1.1 Admittance control

In this section, we show the results of a simple experiment in an indoor environment conducted to test the basic admittance interaction strategy. In detail, the desired left end-effector trajectory is a constant pose, i.e, with reference to Eq. (10):

$$\boldsymbol{a}_L^d = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \boldsymbol{v}_L^d = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \boldsymbol{x}_L^d = \begin{bmatrix} \boldsymbol{p}_L^{\text{ini}} \\ \boldsymbol{o}_L^{\text{ini}} \end{bmatrix} , \tag{82}$$

where $\boldsymbol{p}_L^{\text{ini}}$ and $\boldsymbol{o}_L^{\text{ini}}$ are the initial left end-effector position and quaternion, respectively. The right end-effector is not considered in this experiment, and it is left still. Then, a human operator starts physically interacting with the end-effector, and the robot moves it, showing a compliant behavior with respect to the external force. Figures 8 and 9 show the obtained results.

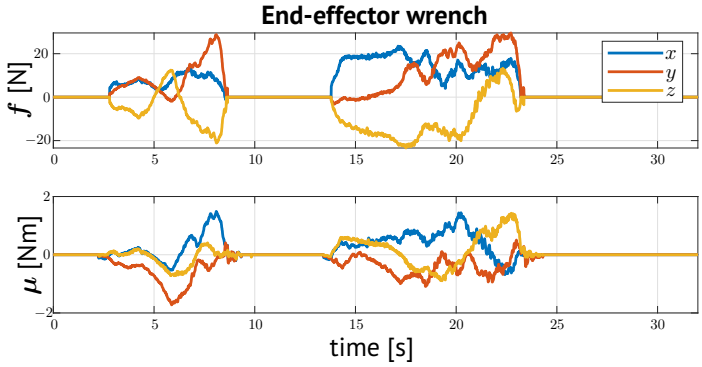

Figure 8: Interaction wrench of the end-effector over time during the experiment on admittance control.

Figure 8 shows the wrench measured by the wrench sensor mounted on the wrist of the left manipulator, while Figure 9 shows the joints position and velocity during the experiment. It is possible to notice that the human operator interacts twice with the end-effector, and after he releases it, the robot performs a movement to return to the desired position, always keeping the joints position and velocity within the

This project is funded by the European Union G.A No 101016906

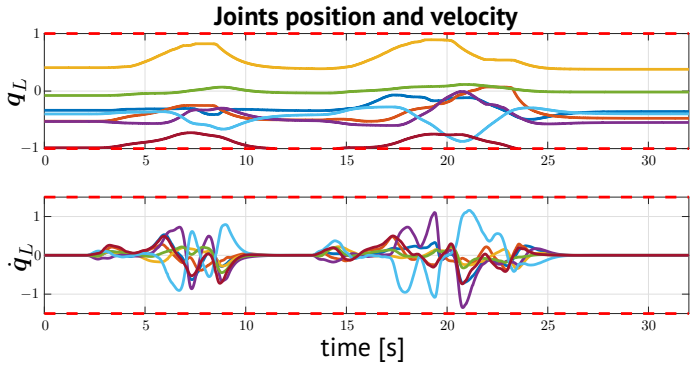

Figure 9: Top: normalized joints position over time during the experiment (solid lines) and their thresholds (red dashed-lines); Bottom: joints velocity over time during the experiment (solid lines) and their thresholds (red dashed-lines)

imposed limits. Figure 10 shows some screenshots of the video of the experiment, which can be found in Section 9 (Admittance.mp4).

This project is funded by the European Union G.A No 101016906



Figure 10: Screenshot of the video showing the admittance control experiment. Top-left: the human operator starts physically interacting with the end-effector. Top-right: the human operator pulls down the end-effector. Bottom-left: the human operator stops interacting with the end-effector. Bottom-right: the end-effector returns to the desired position.

This project is funded by the European Union G.A No 101016906

6.1.2 Hand-guiding control

In this section, we show the results of an experiment in an indoor environment conducted to test the basic hand-guiding interaction strategy. The robot starts in an initial joint configuration, and the hand-guiding strategy is enabled on the left end-effector while the right one is left still. Then, a human operator starts physically interacting with the end-effector and the robot moves it, allowing the human to freely change its configuration. Figures 11 and 12 show the obtained results.

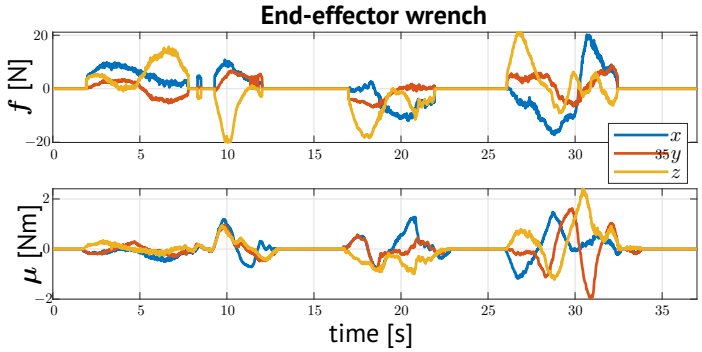

Figure 11: Interaction wrench of the end-effector over time during the experiment.

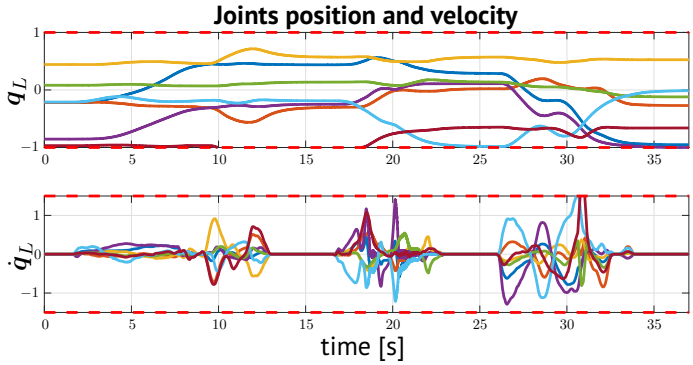

Figure 12: Top: normalized joints position over time during the experiment (solid lines) and their thresholds (red dashed-lines). Bottom: joints velocity over time during the experiment (solid lines) and their thresholds (red dashed-lines)

Figure 11 shows the wrench measured by the sensor mounted on the wrist of the left manipulator, while Figure 12 shows the joints position and velocity during the experiment. It is possible to notice that the human operator interacts in three different phases with the end-effector, and each time he releases it, differently from the previous experiment, the end-effector remains in the current configuration. It is worth noticing that, even in this case, the joints position and velocity remain within the imposed thresholds. Figure 13 shows some screenshots of the video of the experiment, which can be found in Section 9 (Hand_guiding.mp4).

This project is funded by the European Union G.A No 101016906



Figure 13: Screenshot of the video showing the hand-guiding control experiment. Top-left: the human operator gets close to the robot. Top-right: the human operator starts physically interacting with the end-effector. Bottom-left the human operator pulls up the end-effector. Bottom-right: the human operator stops interacting with the end-effector and it remains still.

This project is funded by the European Union G.A No 101016906

6.2 Human-robot cooperative harvesting

6.2.1 Harvesting operation description

Here we present how to harvest a grape bunch effectively. First, the robot has to identify the bunches in its surroundings and accurately estimate the 3D position of the cut-point of each peduncle. This task is performed in real-time by the perception software developed by UPC ([11]) and makes use of the data coming from RGB-D cameras. Once an estimation of the peduncle's cut-point position is available, the harvesting procedure is started, and a 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.

6.2.2 Human-robot cooperation algorithm

Even employing the strategy described in the previous section, 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, the robot changes 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, i.e. the robot employs the human-robot interaction strategy described in Section 3.2. 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. In *autonomous* mode, the secondary task is the Admittance in Section 3.1, and the desired mass $\boldsymbol{K}_M$, damping $\boldsymbol{K}_D$ and stiffness $\boldsymbol{K}_P$ are set to:

$$\begin{aligned} \boldsymbol{K}_M &= \mathrm{diag}\{\begin{bmatrix}20 & 20 & 20 & 3 & 3 & 3\end{bmatrix}\} \\ \boldsymbol{K}_D &= \mathrm{diag}\{\begin{bmatrix}253 & 253 & 253 & 27 & 27 & 27\end{bmatrix}\} \\ \boldsymbol{K}_P &= \mathrm{diag}\{\begin{bmatrix}800 & 800 & 800 & 60 & 60 & 60\end{bmatrix}\}\,. \end{aligned} \tag{83}$$

Moreover, the desired velocity $\boldsymbol{v}^d$ and acceleration $\boldsymbol{a}^d$ are set as the ones computed by a trajectory generation module. These parameters, which have been empirically tuned, ensure that the robot can follow the desired trajectory while exhibiting sufficiently compliant behavior with respect to unexpected external forces caused by accidental contact with the environment. In *hand-guiding* mode, the parameters in Eq. (24) 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. As described in Section 3.2, this set of parameters allows the human operator to freely move the end-effector to the actual location of the peduncle to cut.
Figure 14 shows a graphical representation of the Finite State Machine diagram that implements our shared control strategy.

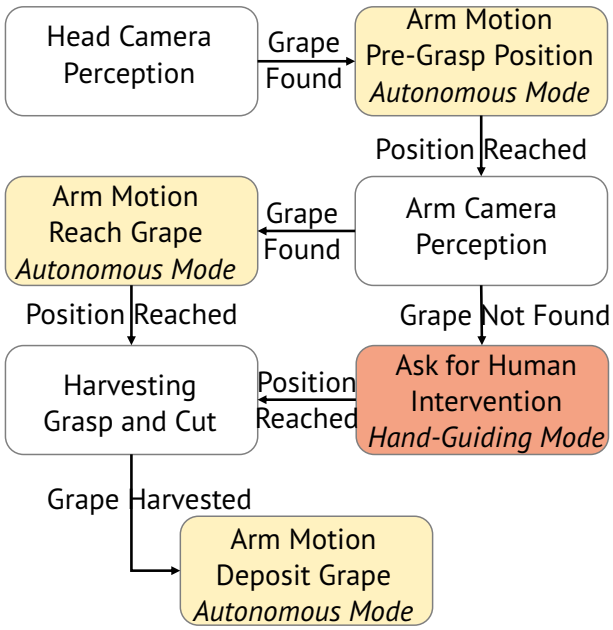

Figure 14: State Machine describing the sequence of operations for human-robot collaboration in grape harvesting. The colors of the blocks encode the robot control mode. Yellow: *autonomous*, orange: *hand-guiding*.

6.2.3 Experimental results

In the following, we report the results obtained from a harvesting experiment conducted within an indoor environment to validate and test the described cooperative harvesting operation. Accordingly to the Finite State Machine shown in Figure 14, at the beginning, the robot works in *autonomous* mode, and the first operation that it performs is the identification of the peduncles of the bunches detected with the aid of the head camera. In this case, due to some occlusions, the robot is not able to identify the peduncle of the detected bunch, so it switches to *hand-guiding* mode. This mode allows a human operator to freely move the end-effector and help the robot in performing its task. After bringing the end-effector to the designed position, which in this case is the cutting point of the peduncle of the detected bunch, the human operator stops interacting with the robot. After a few seconds of absence of interaction, the robot comes back to work in *autonomous* mode and continues to perform the remaining operations necessary to complete the

This project is funded by the European Union G.A No 101016906

harvesting procedure.
Figure 15 summarizes the results of this experiment. It shows the time evolution of the interaction wrench of the left end-effector during the experiment. From this figure it is possible to notice that from $t \approx 40$s to $t \approx 60$s the robot is subject to an external wrench, which is applied by the human operator to lead the robot end-effector to the cut-point of the peduncle of the detected bunch. As can be seen, from $t > 60$ the norm of the interaction wrench is approximately null; this means that after leading the end-effector to the desired pose, the human releases it and stops interacting with the robot. The robot detects this condition by comparing the norm of the interaction forces and moments with given thresholds f_{min} and μ_{min}, which have been set equal to 3N and 1Nm, respectively. When the $||\boldsymbol{f}_L|| \leq f_{\text{min}}$ and $||\boldsymbol{\mu}_L|| \leq \tau_{\text{min}}$ for more than 3s it switches back to *autonomous* mode. From that point on, the robot autonomously continues the harvesting procedures with the admittance parameters as is Eq. (83). In particular, the two-finger gripper first grasps the peduncle (at $t \approx 70$s); then, the Trajectory Generation module plans a trajectory to release the grape on the front part of the robot (at $t \approx 130$s). It is worth noticing that, during this phase, the measured interaction wrench is mainly represented by gravity due to the grape weight (yellow line in the top plot in Fig. 15). Figure 16 shows some screenshots of the video of the experiment, which can be found in Section 8 (Cooperative_harvesting.mp4).

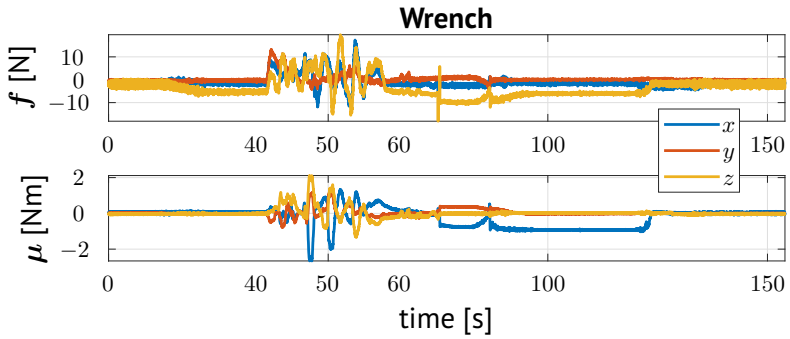

Figure 15: Interaction wrench of the left end-effector over time during the cooperative harvesting experiment.

This project is funded by the European Union G.A No 101016906



Figure 16: Screenshot of the video showing the cooperative harvesting experiment. Top-left: the robot recognizes the grape and brings the end-effector close to it in order to have a more accurate estimation of the peduncle position. Top-right: the grape is occluded by leaves and the robot asks for the human operator's help. Bottom-left: the human operator manually moves the end-effector towards the peduncle. Bottom-right: the robot autonomously continues the harvesting procedure.

6.2.4 Shared control

Results are here provided to validate the approach proposed in Section 3.3. At the same time, it is shown the effectiveness of the approach devised in Section 4 concerning the distributed multi-robot scenario depicted in Figure 4.

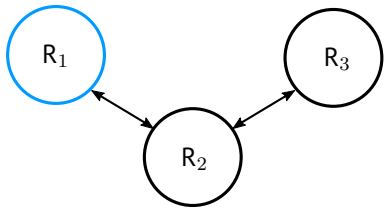

Figure 17: Communication graph. Leader robot is highlighted with a blue node.

In detail, in the depicted scenario, the arms are rigidly grasping a cylindrical object as in Figure 4. As already known, the Farming Robot is redundant (we consider, for the sake of simplicity, that, for each farming robot, only one arm is grasping the object, thus keeping the other unused); moreover, the reduced regressor matrix $\boldsymbol{Y}_i \in \mathbb{R}^{6\times 56}$ ($n_{\pi_i} = 56$) in (4) is considered and a uniformly distributed 5% uncertainty is assumed concerning the robots' dynamic parameters $\boldsymbol{\pi}_i, i = 1, 2, 3$. For the sake of the implementation on a real setup, an initial estimation of the robot dynamic model might be obtained by resorting to one of the approaches mentioned in [18]. Furthermore, it is also required the robots agree on a common reference frame in which the object trajectory and the estimation of global variables are specified and/or computed. However, this requirement

Parameter	Value	Parameter	Value
$\boldsymbol{M}_v$ (eq. (33))	$\text{diag}\{2\boldsymbol{I}_3\text{kg}, \boldsymbol{I}_3\text{kgm}^2\}$	$\bar{e}$ (eq.(42))	0.5
$\boldsymbol{D}_v$ (eq. (43))	$\text{diag}\{10\boldsymbol{I}_3\text{Ns/m}, 8\boldsymbol{I}_3\text{Nms}\}$	T (eq.(42))	0.001s
$\boldsymbol{K}_\gamma$ (eq. (44))	$0.3\boldsymbol{I}_6$	Δ (eq.(42))	0.2s
t_h (eq. (44))	1	$\boldsymbol{K}_\theta$ (eq. (55))	$800\boldsymbol{I}_6$
$\boldsymbol{R}_h$ (eq. (34))	$0.5\boldsymbol{I}_3$	γ (eq. (62))	300
$\boldsymbol{R}_v$ (eq. (34))	$0.1\boldsymbol{I}_6$	$\mu_{v,1}$ (eq. (66))	30
$\boldsymbol{Q}_{r,d}$ (eq. (34))	$10\ \text{diag}\{250\boldsymbol{I}_3, \boldsymbol{I}_3, 250\boldsymbol{I}_3, \boldsymbol{I}_3\}$	k_f (eq. (68))	0.5
$\boldsymbol{W}_{h,d}$ (eq. (35))	$\boldsymbol{I}_6$	$\mu_{v,2}$ (eq. (66))	25
λ (eq.(40))	0.998	k_c (eq. (69))	1
$\boldsymbol{P}_0$ (eq.(40))	$10000\boldsymbol{I}_9$	k_p (eq. (70))	1
α (eq. (41))	0.1	$\boldsymbol{K}_s$ (eq. (72))	1

Table 1: Simulation parameters.

is common to almost all distributed algorithms and, in the case of mobile manipulators, implies the robots to be able to localize in a common frame; this can be obtained, for example, by using passive markers to be detected by on-board vision sensors. The latter can also be used to estimate the point at which the object is grasped by the human operator if such information is not known beforehand.

Concerning the communication graph used to model the information exchange (see Section 4.2), shown in Figure 17, bi-directional communication links are considered between robots 1 and 2 and between robots 2 and 3, while robot 1, for example, is set as leader. Finally, the manipulated object is represented by a cylinder with mass $m_o = 5$ kg, radius $r_o = 1$ m and height $h_o = 0.05$ m. In order to simulate the overall system dynamics, the Udwadia-Kalaba equation of motion for constrained systems is adopted, which, as done in [30], allows to compute the constraint wrenches $\boldsymbol{h}$ arising from the robot rigid grasping constraint. In addition, uncertainty on both the manipulated object model and initial human parameters is introduced. In particular, for each robot, a uniformly distributed 3% of error is generated regarding the object dynamic model, while, concerning the initial human parameters, an uncertainty with normal distribution $\mathcal{N}(0, 0.3\pi_{h,i})\, \forall i \in \{1, ..., 9\}$, with $\pi_{h,i}$ the i th element of vector $\boldsymbol{\pi}_h$ in (12), is introduced.

The objective of the simulation is to achieve the shared control described in Section 3.3 of the object motion where, as shown in Figure 18, the robots task is to perform a circular trajectory in xy-plane with radius $r_c = 1.5$ m, center $\boldsymbol{c} = [0 \quad 0 \quad 0.94]^{\text{T}}$m and period $T_c = 3.8$ s, while keeping the orientation constant, whereas the human desired configuration coincides with the circumference center, i.e. $\boldsymbol{p}_{h,d} \equiv \boldsymbol{c}$. Note that, as assumed, for example, in [29], a constant human desired motion is typically considered for human reaching motions. The human operator force is modeled according to eq. (11) where stiffness $\boldsymbol{K}_h$ and damping $\boldsymbol{D}_h$ matrices are increased during the simulation time (as discussed below and shown in Figure 19 with solid lines) to dynamically modify the object trajectory and make the human leading the motion. Finally, zero internal wrenches are required, i.e. the stacked vector of the desired internal wrenches $\boldsymbol{h}^d_{int} = [\boldsymbol{h}^{d\,\text{T}}_{int,1}\ \boldsymbol{h}^{d\,\text{T}}_{int,2}\ \boldsymbol{h}^{d\,\text{T}}_{int,3}]^{\text{T}} \in \mathbb{R}^{18}$ is $\boldsymbol{h}^d_{int} = \boldsymbol{0}_{18}$.

Simulation parameters are reported in Table 1. Among them and concerning the human parameters estimation discussed in Section 3.3.2, the initial covariance matrix $\boldsymbol{P}_0$ is set to a high value to denote a high uncertainty on the initial human parameters, while the values of T and Δ are selected as in [14]. In addi-

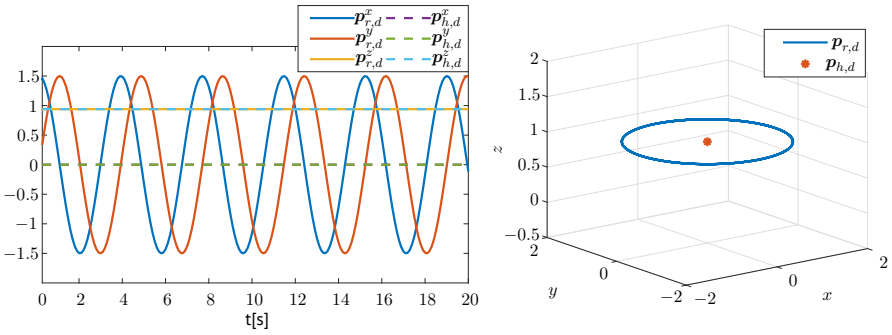

Figure 18: Representation in 2D (left) and 3D (right) of the robots' desired position (solid lines on the left and blue line on the right) and the human desired one (dashed lines on the left and red star on the right).

tion, the values of the weighting matrix $\boldsymbol{Q}_{r,d}$ are defined so as to make the contribution associated with the robots desired trajectory comparable with respect to the one associated with the human operator. Finally, the rationale behind the gain matrix $\boldsymbol{K}_\theta$ is to set it to a higher value than the low-level control law gains in order to estimate the respective quantity sufficiently fast. The same reasoning applies for the choice of the gain γ.

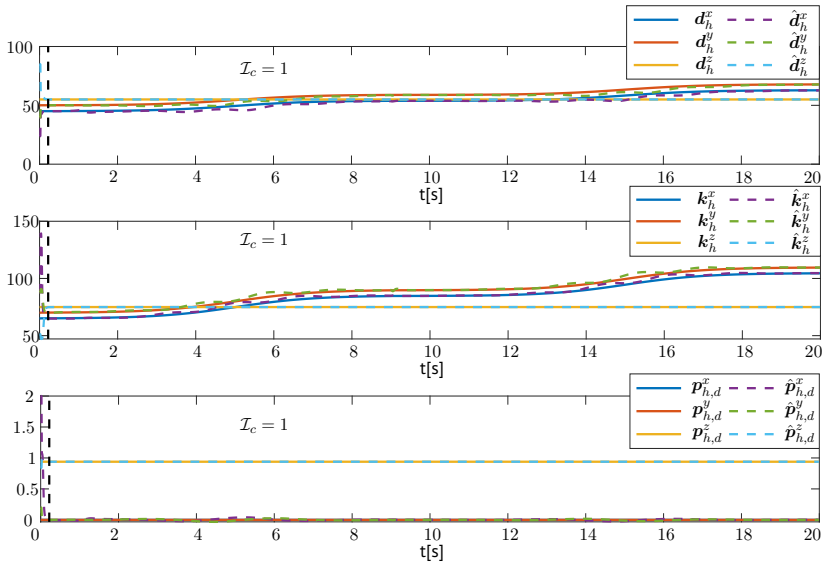

Figure 19: Evolution of the human parameters $\boldsymbol{d}_h$, $\boldsymbol{k}_h$ and $\boldsymbol{p}_h^d$ (from the top, solid lines), respectively, along x, y and z directions and the respective estimates (dashed lines); the time instant $t_c \approx 0.22$ s in which the confidence index $\mathcal{I}_c$ in (42) becomes 1 is highlighted by the black vertical line.

Simulation results are reported in Figures 19-28. More specifically, Figure 19 shows the evolution of the human parameters (solid lines) along x, y, and z directions, as well as the respective estimates (dashed lines) computed by the leader robot. In particular, the human stiffness $\boldsymbol{K}_h$ (middle plot) increases along x and y during the time intervals $[2, 8]$ s and $[12, 18]$ s in order to increase the leading action of the human.

Similarly, an increase of the human damping (top plot) is simulated during the same intervals as a stabilizing and slowing down action. In addition, the time instant $t_c \approx 0.22$s, in which the estimate is assessed to be reliable according to the confidence index in (42), is highlighted (vertical black line).

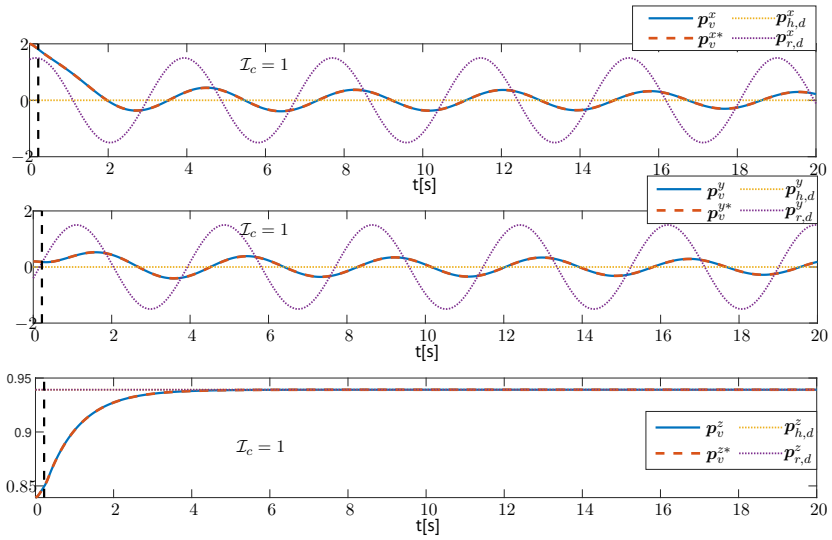

Figure 20: Evolution of the position of the virtual model (blue solid lines) along x, y and z directions compared with the ideal one obtained with the real human parameters $\boldsymbol{\pi}_h$ (red dashed lines); for the sake of completeness, human (yellow dotted lines) and robot (purple dotted lines) desired positions and time instant when it holds $\mathcal{I}_c = 1$ (black line) are also shown.

Figure 20 reports the position components $\boldsymbol{p}_v$ of the desired object trajectory (solid bold lines) resulting from the devised strategy (orientation variables are constant). In particular, as long as the estimate is considered not reliable ($t < t_c$), the desired object motion is generated according to the admittance model in (43) and is driven towards the desired position of the human; then, starting from time t_c, the optimal policy in (36) for the shared control is adopted which leads to performing a circular trajectory with time-varying radius ($< r_c$) and time-varying period ($< T_c$). In detail, the trajectory dynamically adapts according to the human behavior: the higher the human stiffness, the more the motion amplitudes reduce, being the human desired position coincident with the center of the trajectory, i.e. $\boldsymbol{p}_{h,d} = [0\ 0\ 0.94]^{\mathrm{T}}$ m, as also reported in the figure (yellow dotted lines); moreover, the higher the human damping, the slower the resulting motion is. For the sake of comparison, the robots desired position $\boldsymbol{p}_{r,d}$ is also shown in the figure (purple dotted lines). In addition, in order to show the effect of human arm parameter estimation on the overall behavior of the system, the object trajectory obtained in the ideal case of zero human parameter estimation error, i.e. with $\boldsymbol{\pi}_h = \hat{\boldsymbol{\pi}}_h$, is reported (dashed line, denoted with $\boldsymbol{p}_v^*$) and is shown to be equivalent to the one derived with the estimated parameters in Figure 19.

Figure 21 (top plot) shows the error $\|{}^l\hat{\boldsymbol{h}}_h - \boldsymbol{h}_h\|$ between the real human wrench $\boldsymbol{h}_h$ and the one estimated according to Lemma 2. The figure shows that the estimated wrench converges to the real one and in finite time according to the theory. Therefore, each agent is adopting a consistent estimate for computing the local control strategy. In the same way, the bottom plot shows that the internal wrench $\boldsymbol{h}_{int}$ converges to the desired value $\boldsymbol{h}_{int}^d$ since the error $\|\boldsymbol{h}_{int} - \boldsymbol{h}_{int}^d\|$ converges to the origin and required in Problem 1. A

zoom of the evolution of both quantities up to 0.05 s is provided in the plots.
The purpose of Figure 22 is to show the performance of the two-layer architecture presented in Figure 2. As explained in previous sections, the high-level is in charge of defining the object reference trajectory $\boldsymbol{x}_v$, which is the solution of the optimal problem defined in (34). This trajectory represents the desired trajectory of the object $\boldsymbol{x}_o$ to be tracked by the robots. Therefore, Figure 2 shows that the position $\|\boldsymbol{x}_v - \boldsymbol{x}_o\|$ (top) and velocity $\|\dot{\boldsymbol{x}}_v - \dot{\boldsymbol{x}}_o\|$ (bottom) tracking errors converge to the origin, which means that the real object trajectory converges to the desired one.

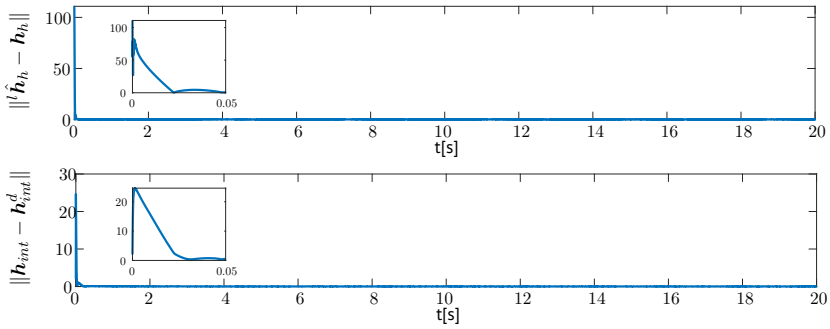

Figure 21: Evolution of the human forces estimation error (top) and of the internal forces error (bottom).

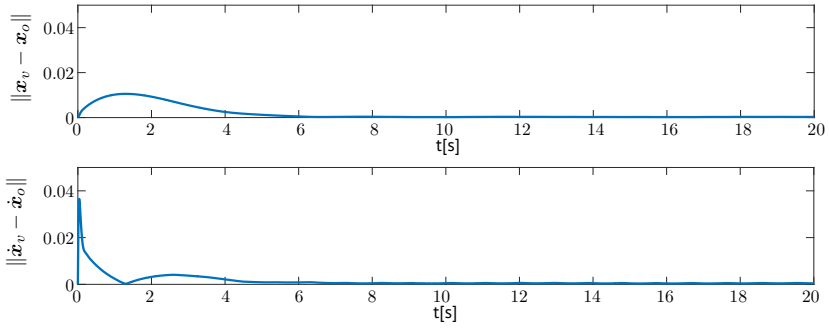

Figure 22: Evolution of trajectory tracking error $\|\boldsymbol{x}_v - \boldsymbol{x}_o\|$ (top) and its derivative $\|\dot{\boldsymbol{x}}_v - \dot{\boldsymbol{x}}_o\|$ (bottom).

Finally, the effects of some main parameters of the framework have been analyzed. More specifically, the influence of the weight matrices $\boldsymbol{Q}_{r,d}$ and $\boldsymbol{R}_v$ in the cost function (34) as well as the inertia matrix $\boldsymbol{M}_v$ of the virtual model (33) has been considered. No plot of the internal wrench is provided in the following analysis since it is not affected by the considered matrices and for the sake of space.
Concerning the weight matrices $\boldsymbol{Q}_{r,d}$, it regulates how much relevance is given to tracking the robots trajectory and its effect has been investigated in regard to the resulting shared motion and human force. In particular, by expressing $\boldsymbol{Q}_{r,d}$ as $\boldsymbol{Q}_{r,d} = \text{diag}\{Q_{r,p}\boldsymbol{I}_3, Q_{r,o}\boldsymbol{I}_3, Q_{r,p}\boldsymbol{I}_3, Q_{r,o}\boldsymbol{I}_3\}$, Figures 23 and 24 report the object motion and human force, respectively, obtained by varying $Q_{r,p}$ in the set $\{10, 500, 2500, 10000\}$. For the sake of comparison, the human's desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown in Figure 23. The figures make evident that, as the weight $Q_{r,p}$ decreases, the resulting object trajectory approaches the human desired configuration $\boldsymbol{p}^d_h$, and, accordingly, the human effort

This project is funded by the European Union G.A No 101016906

decreases since the person leads the control. On the contrary, the higher the value, the more the resulting trajectory gets closer to the robots desired circular motion, thus leading to a major human effort. It follows that the role of human and robots in the shared control is driven both by the tuning of the static weighting matrices in the cost function, such as $\boldsymbol{Q}_{r,d}$, and dynamically, by the time-varying parameters of the human arm. No analysis of the weight matrices $\boldsymbol{Q}_{h,d}$ and $\boldsymbol{R}_h$ has been reported since their effect is dual to the one of $\boldsymbol{Q}_{r,d}$, i.e. the higher norm, the more the human desired trajectory is pursued with respect to the robot one and the human force minimized.

As far as the weight $\boldsymbol{R}_v$ is concerned, it regulates how much relevance is given to the minimization of the robots effort $\boldsymbol{u}_v$. Figures 25 and 26 report the norm of the overall robot efforts and the resulting object motion, respectively, when varying R_v. More specifically, by denoting with $\boldsymbol{u} = [\boldsymbol{u}_1^{\mathrm{T}} \ \boldsymbol{u}_2^{\mathrm{T}} \ \cdots \boldsymbol{u}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Np}$ the stacked vector of the robots inputs, Figure 25 shows how by increasing the weight R_v, the robots control effort, i.e. $\|\boldsymbol{u}\|$, reduces. This is achieved, as reported in Figures 26, by slowing down the trajectory and decreasing the amplitude of the oscillations. It thus follows that the higher R_v, the more the control effort is reduced at the expense of lower tracking performance of the desired robot trajectory. As for the previous parameters, the human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown in Figure 26 for the sake of completeness. Note that the same trajectory is recorded for all parameters for $t < t_c$ since the weight matrices are only used in the cost function and do not affect the admittance controller in eq. (43) that is used when the human estimation is not reliable.

Regarding the inertia matrix, let us formulate it as $\boldsymbol{M}_v = \mathrm{diag}\{M_{v,p}\boldsymbol{I}_3, M_{v,o}\boldsymbol{I}_3\}$. The following set of values matrix has been tested for analyzing its influence: $M_{v,p} = \{2, 5, 10, 50, 500\}$. Figures 27 and 28 report the resulting object motion and the norm of the overall robots' effort, respectively, when varying $M_{v,p}$. In particular, the latter influences the bandwidth of the virtual dynamics, implying that the higher $M_{v,p}$, the slower the system's response will be. This behavior is evident from the figures where a more dampened motion (see Figure 27), with reduced acceleration, is obtained as $M_{v,p}$ grows. This also implies that a lower robots control effort is required when increasing $M_{v,p}$.

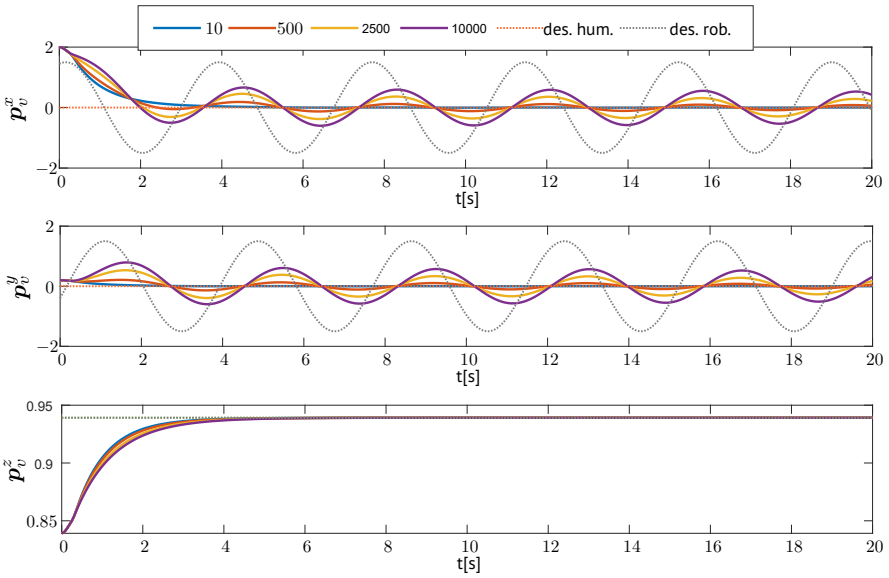

Figure 23: Evolution of the object motion (bold lines) along x, y and z directions by varying the robots weight $Q_{r,p} \in \{10, 500, 2500, 10000\}$ in the cost function. The human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown.

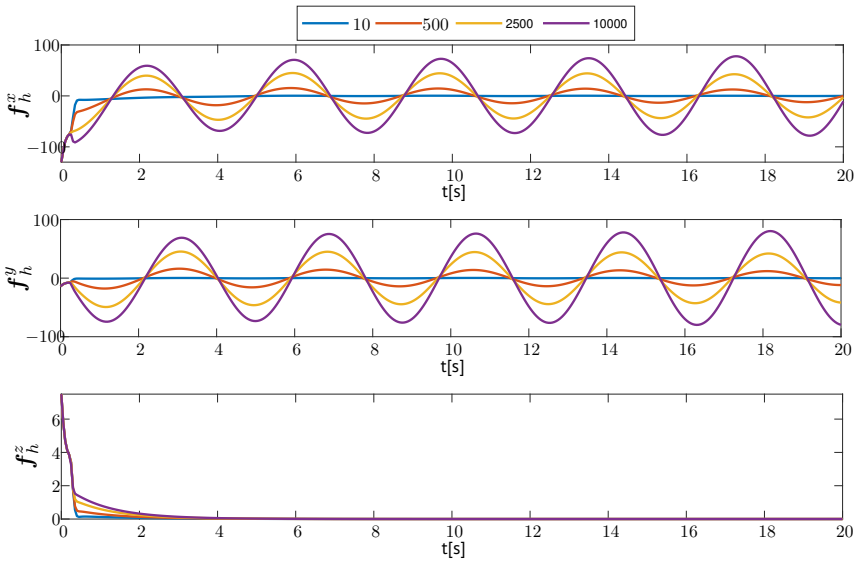

Figure 24: Evolution of the human force along x, y and z directions by varying the robots weight $Q_{r,p} \in \{10, 500, 2500, 10000\}$ in the cost function.

This project is funded by the European Union G.A No 101016906

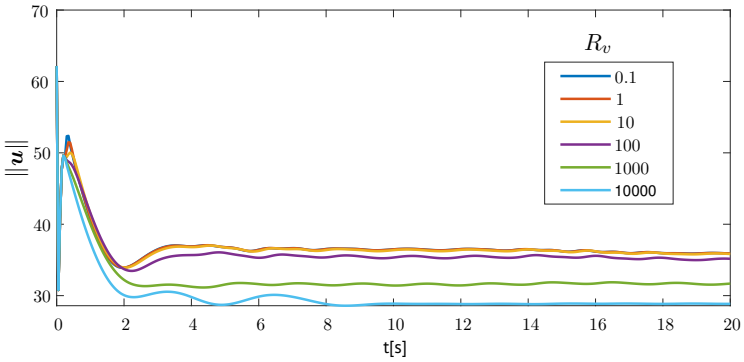

Figure 25: Evolution of the norm of the overall robots control input by varying the robots effort weight $R_v \in \{0.1, 1, 10, 100, 1000, 10000\}$ in the cost function.

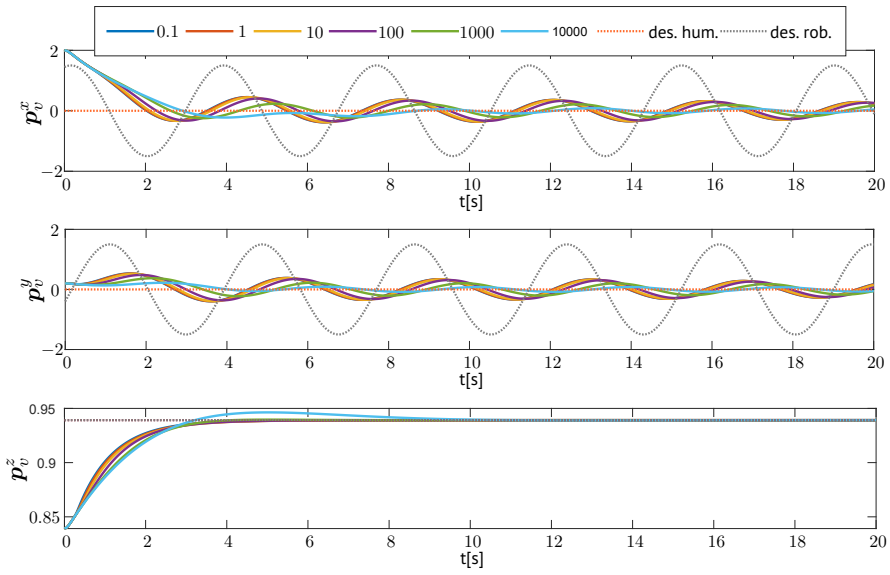

Figure 26: Evolution of the object motion along x, y and z directions by varying the robots effort weight $R_v \in \{0.1, 1, 10, 100, 1000, 10000\}$ in the cost function. The human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown.

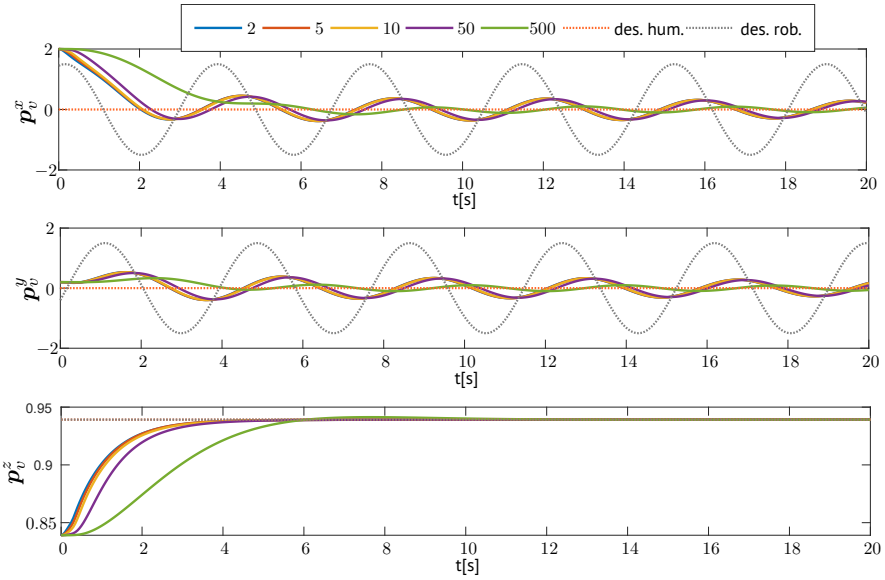

Figure 27: Evolution of the object motion along x, y and z directions by varying the virtual inertia $M_{v,p}$. The human desired motion (dotted orange line) and the robots desired one (dotted gray line) are also shown.

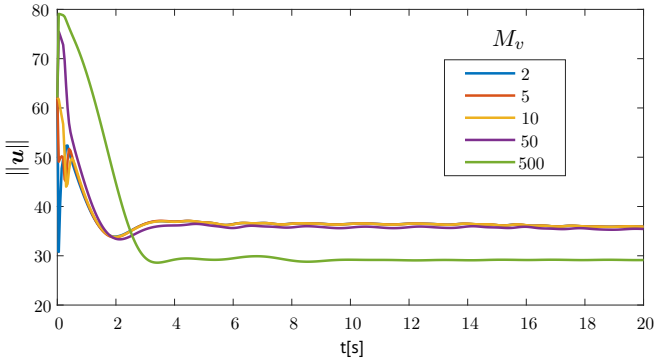

Figure 28: Evolution of the norm of the robots control input by varying the virtual inertia $M_{v,p}$.

6.3 Neural-network contact detection and classification

6.3.1 Experiments on a Kinova Jaco^2 robot

We describe here experimental results obtained by employing the contact classification method devised in Section 5.2.1 on a Kinova Jaco^2 robot. It is worth noticing that this robot is equipped with torque sensors for each joint, and after a proper identification procedure, we were able to estimate the matrices of the dynamic model of the robot as in Section 2.2.
The system setup, illustrated in Figure 5, is composed of a Kinova Jaco^2 robotic arm with 7 DOFs and a Microsoft Kinect One RGB-D sensor. All the algorithms related to the contact classification and to the control run on Matlab 2020 and exchange information with Robot Operating System (ROS). A vision-based localization component (green block in 5.2.1) is also included in ROS. In detail, a human skeleton tracking algorithm[1] is adopted, which, combined with the robot configuration information, allows identifying the point P along the robot structure where human contact occurs. Concerning the robot arm, the following collaborative tasks are considered. It is worth mentioning that the considered tasks are related to pick and place operations of common objects that can be exploited to carry out agronomic operations as well, as, for example, grape transportation in boxes, or agronomic tools (hand over) in human-robot collaboration scenarios.

Object pick and place. The robot can manipulate bottles and mugs. To prevent the liquid from spilling out of the objects, we consider the following orientation constraints:

$$\begin{aligned} \phi^l_{o,i} &= \varphi_i - (\varphi_{d,i} - \Delta\varphi_i) \geq 0 \\ \phi^u_{o,i} &= (\varphi_{d,i} + \Delta\varphi_i) - \varphi_i \geq 0 \end{aligned} \quad i = 1, 2, 3 \tag{84}$$

which can be seen as Control Barrier Functions (CBFs), being $\boldsymbol{\varphi}_d \in \mathbb{R}^3$ a desired orientation expressed as Euler angles, $\Delta\varphi_i$ a positive tolerance, and $(\cdot)_i$ the i th component of the respective vector. The above constraints have a relative degree equal to 2, which are handled as in [3]. More details about the inclusion of CBFs-based in the HQP control framework are given in Deliverable D5.1.

Pouring task. This task foresees the robot holding a mug and the human filling it. The nominal wrench profile due to the filling is approximated with $K_{\mathcal{T}} = 5$ Gaussians and by using 5 randomly selected demonstrations. The GMM parameters for the EM algorithm are initialized by resorting to a K-means clustering algorithm. Also in this case, the end effector orientation is constrained as in (84).

Table cleaning The robot executes a periodical motion on the table surface holding a cylindrical sponge (see Figure 5). The force exerted on the table is approximated through GMMs by using 5 randomly selected demonstrations with, as before, $K_{\mathcal{T}} = 5$ and K-means algorithm for initialization.
Note that the human is always allowed to physically interact with the robot end effector, changing its configuration, or with the robot structure, changing its internal joint configuration. In this way, the user can help the robot to accomplish the task or to avoid possible collisions with obstacles not detected by the vision system.

[1] `https://github.com/mcgi5sr2/kinect2_tracker`

6.3.1.1 Behavior selection. The Finite State Machine (FSM) in Figure 29 regulates the activation of the appropriate behavior. More specifically, the robot task is carried out as long as no human interaction is detected, or if a force is voluntarily applied along the robot structure to internally reconfigure it. Then, in the case an intentional contact is recognized at the end effector, the admittance control strategy in Section 3.1 is selected, which enables the human to adjust the robot task. The robot persists in this state until the human exerts a wrench at the contact point. Finally, if an accidental contact is recognized, the robot selects the avoidance behavior to increase the human safety field F. The robot persists in this state as long as a safety condition is not restored, i.e. as long as $F < F_{min}$, with $F_{min} > F_d$ a scalar threshold.

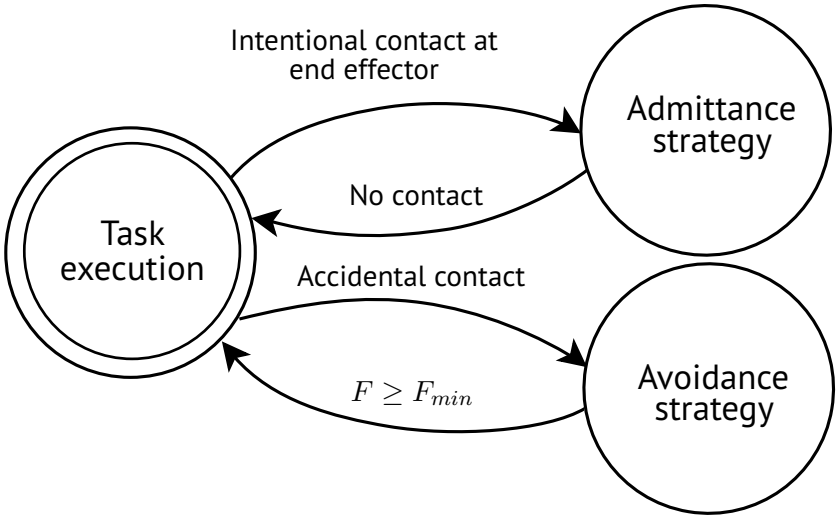

Figure 29: FSM for robot's behaviors activation.

6.3.1.2 Classification results. Here we show the classification results of the two RNNs, for which 100 hidden units have been considered. In particular, the detection NN determines if a datapoint in the time series belongs to the class *no contact* (nc) or *with contact* (wc), while the recognition network establishes if, in the case of contact, a datapoint belong to the class *intentional contact* (ic) or *accidental contact* (ac).
The training dataset of the detection (recognition) NN consists of 43487 (15647) time samples, representing 869.7 s (312.9 s) with time step $T = 0.02$ s, equally distributed in their respective two classes. Concerning the test set, we evaluated the accuracy both with the user that was involved in the collection of the training dataset, referred to as *training user*, and with other four users *not* involved in the training process, referred to as *novel users*. Table 2 reports the confusion matrices of the classifiers for the training user (top part) and the novel users (bottom part). We observe that all the RNNs achieve good overall accuracy, which is, for the training user, $\approx 94\%$ for the detection NN and $\approx 90\%$ for the recognition NN. The lower accuracy of the latter is motivated by the intrinsic greater complexity of the problem of recognizing the type of contact rather than identifying *any* possible contact. Remarkably, the results also confirm the generalization capabilities of the chosen NNs, which allow classification also with novel users. In detail, an overall accuracy equal to $\approx 90\%$ and $\approx 82\%$ is obtained for the detection and recognition classifications, respectively, implying a decrease of performance of only $\approx 4\%$ and $\approx 8\%$, compared to the training user. Finally, note that the results in Table 2 are obtained with a sample-by-sample evaluation; however, a certain delay for detecting and classifying a

This project is funded by the European Union G.A No 101016906

Training user

Detection NN

	nc	wc	
nc	4697	373	92.6%
wc	139	2930	95.5%
	97.1%	88.7%	93.7%

Recognition NN

	ic	ac	
ic	3979	135	96.7%
ac	555	1826	76.7%
	87.8%	93.1%	89.3%

Novel users

	nc	wc	
nc	6120	456	93.1%
wc	2836	19745	87.4%
	68.3%	97.7%	88.7%

	ic	ac	
ic	21041	2647	88.8%
ac	3669	7016	65.7%
	81.2%	72.6%	81.6%

Table 2: Confusion matrices of the RNNs with data acquired by the user involved in the training dataset (top part) and with different users not involved in the training dataset (bottom part).

contact always occurs as the networks obviously need some samples before being able to correctly classify. This motivates the classification inaccuracies in Table 2, but, as demonstrated in the following case studies, it does not undermine the human-robot collaboration.

6.3.1.3 Case studies and experimental results. We consider two case studies for the experimental validation which involve different tasks and human interactions. In both case studies, we used the following parameters: $\boldsymbol{K} = 50\boldsymbol{I}_7$, $\boldsymbol{K}_d = 5\boldsymbol{I}_6$, $\boldsymbol{K}_p = 6\boldsymbol{I}_6$, $\boldsymbol{M}_d = 5\boldsymbol{I}_6$, $\boldsymbol{D}_d = 100\boldsymbol{I}_6$, $k_d = 5$, $k_p = 6$, $\boldsymbol{Q} = \boldsymbol{I}_7$. Moreover, we set $F_d = 10$ to ensure a minimum distance ≈ 0.4 m (see [31] for details) between every point of the robot and the human operator, and $F_{min} = 11$ in the FSM. A video of the experiments can be seen found in Section 9 (Collision_detection_jaco.mp4).

Case study **1**. In this case study, we let the robot execute three different tasks while the human interacts with it. At the beginning, the robot takes the cylindrical sponge and starts a cleaning task, during which the human intentionally interacts with its end effector to change its configuration. Once the cleaning task is completed, the robot starts a bottle pick and place operation. During this task execution, the human voluntarily interacts along the robot structure to modify the joint configuration while preserving the end effector position. Finally, a pick operation of a mug and a pouring task are performed. During each task, the respective constraints and possible expected environment interaction defined at the beginning of the section are taken into account.
Figure 30 summarizes the results of this case study. In detail, proceeding from the top to the bottom, it reports the robot end effector trajectory $\boldsymbol{x}$, the nominal interaction wrench $\hat{\boldsymbol{h}}_{\mathcal{T}}$, the norm of estimated human torque $\|\hat{\boldsymbol{\tau}}_h\|$ and the classification output of the RNNs (in blue) compared to the Ground Truth (GT, in green). Initially, the robot autonomously performs the cleaning task (highlighted with red boxes in the plots) and ex-

ecutes the desired periodic motion (first plot), which generates interaction wrenches with the table. The use of the nominal wrench profile (second plot) modeled via GMM allows to estimate the human torque (third plot) and to properly recognize that, in the initial phase, only interaction with the environment is occurring while no human contact is present (last plot). Once the human intentionally interacts with the end effector at $t \approx 21$ s, it is recognized by the RNNs (with a delay < 1 s), and the admittance behavior is activated. This makes the robot end effector compliant towards the human wrench, and its trajectory is modified, as shown in the first plot. At the end of the interaction, the execution of the desired tasks is restored, and the pick and place operations are carried out (highlighted with green boxes) from $t \approx 27$ s to $t \approx 100$ s. In this phase, an intentional contact on link 4 of the robot is recognized via the localization component at $t \approx 70$ s, which leads the robot to change its joint configuration while preserving its end effector position, as shown in the top plot. Orientation variables are also limited according to (84) with $\boldsymbol{\varphi}_d = [-1.4,\ 1.6,\ -2.9]^{\mathrm{T}}$ and $\Delta\varphi_i = 0.2\,\forall i$. Finally, the pouring task (blue box in the plots) is executed starting from $t \approx 100$ s during which the respective GMM model is exploited and no contact with the human is detected.

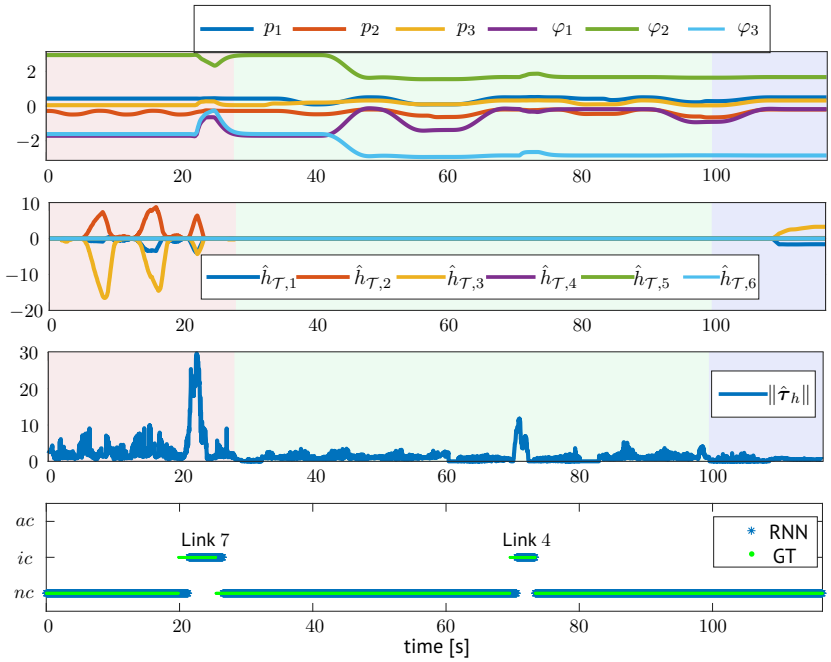

Figure 30: Case study 1. From the top: end effector trajectory, nominal task interaction wrench, norm of the estimated human torques and classification results compared to ground truth. Cleaning, pick and place and pouring phases are denoted in red, green and blue.

Case study **2**. This case study aims to prove the effectiveness of the solution also with accidental contacts. It is structured as follows: the human first intentionally interacts with the robot end effector during a pick-and-place task, and then an accidental collision happens.
Figure 31 shows, from the top, the robot end effector trajectory $\boldsymbol{x}$, the norm of the estimated human torques $\|\hat{\boldsymbol{\tau}}_h\|$ and the classification results (in blue), compared to ground truth values (in green). Based on the estimated human wrench (second plot) and on the localization component, an intentional contact at the end

effector is recognized by the RNNs (third plot) at $t \approx 9$ s; then, the admittance strategy is activated, leading to a modification of the robot trajectory. The robot task execution is then restored when the interaction terminates, and an accidental contact is detected on link 7 at $t \approx 33$ s. This activates the avoidance behavior, which drives the robot away from the human operator in order to increase the safety.

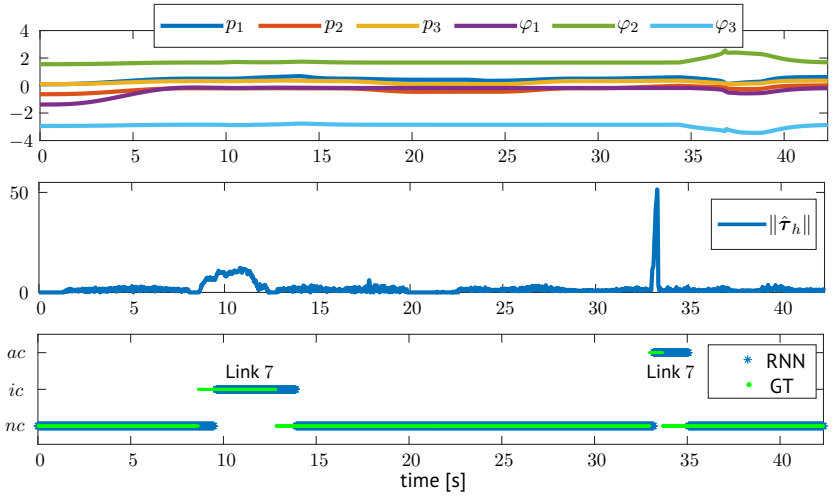

Figure 31: Case study 2. From the top: end effector trajectory, estimated human torques and classification results compared to ground truth.

6.3.2 Experiment on the Canopies farming robot

In this section, we describe an experiment conducted on the CANOPIES farming robot by making use of the contact classification algorithm presented in Section 5.2.2. In detail, the use case scenario taken into account is a typical harvesting operation, in which we foresee the presence of a human operator that might physically interact with the robot both accidentally and intentionally.
The Finite State Machine (FSM) in Figure 32 regulates the activation of the appropriate behavior. More specifically, the robot starts executing the admittance control strategy in Section 3.1, performing the harvesting operation. When an accidental contact is detected, the robot switches to the hand-guiding strategy in Section 3.2 in order to avoid damaging its own structure and hurting the human operator. It stays in this state until no contact is detected for N_s seconds. When this condition is met, the state switches back to admittance control, allowing the robot to resume the harvesting operation. It is worth mentioning that, differently than strategy in Section 6.3.1, here no collision avoidance behavior is introduced, since a different approach is adopted to ensure the human safety. Indeed, the hand-guiding mode keep the end-effector still when no force is applied, while moves the robots in the direction of the exerted force (and, then, in the opposite direction of the human operator) when a contact occurs.

6.3.2.1 Classification results: This section contains the results of the training process performed on a model of a *cubic KNN* classifier aimed at determining if a data-point in a time series belongs to the *no contact* class (*NC*), the *accidental contact* class (*AC*) or the *intentional contact* class (*IC*).

This project is funded by the European Union G.A No 101016906

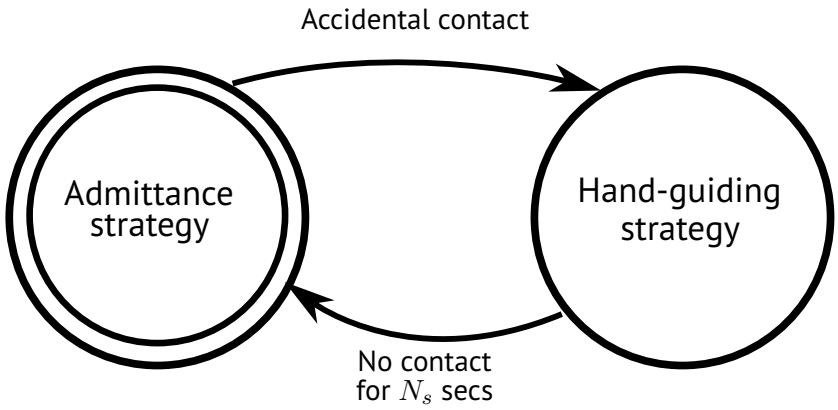

Figure 32: FSM for robot's behaviors activation.

nc	93.1%	0.9%	6.0%	93.1%	6.9%
ac		75.0%	25.0%	75.0%	25.0%
ic	2.8%	1.3%	95.9%	95.9%	4.1%
	nc	ac	ic	trp	fnr

Table 3: Confusion matrix of the classifier resulting from the training performed via a 10-fold cross-validation scheme using the data collected by performing voluntarily and accidentally interactions with the robot.

The training has been performed with the aid of the Classification Learner App tool of Matlab 2023, and as mentioned in Section 5.2.2, it has been performed via a validation scheme to prevent the overfitting of the model. In detail, the scheme chosen for evaluating the predictive accuracy of the model and preventing its overfitting during the training process is the *k-fold cross-validation*, with $k = 10$.
The dataset used for training the model is made of 657 sets of features and has been derived from a set of 62 time series, which have been collected with a time step T =0.01s by voluntarily and accidentally interacting with the robot. In detail, it has been derived by performing a features extraction process on time windows of 1 second duration.
Table 3 contains the confusion matrix of the resulting three-class classifier, where trp and fnr denote the columns of the *true-positive rates* and *false-negative rates*, respectively. As can be seen from the figure, the resulting model can recognize the class nc with an accuracy over 93% (93.1%), the class ac with an accuracy of 75% and the class ic with an accuracy over 95% (95.9%).

6.3.2.2 Case study and experimental results: The experimental validation of the classification algorithm has been performed on the CANOPIES farming robot. In detail, in the case study taken into account a human operator interacts, first intentionally and then accidentally, with the farming robot while it is performing a harvesting operation.
According to the Finite State Machine shown in Table 32, the default interaction strategy employed by the

This project is funded by the European Union G.A No 101016906

robot, i.e. when there is no contact detected, is the admittance control strategy. At the beginning of the experiment, the robot brings the end-effector of its left arm to a pre-grasp position. During this phase, a human operator intentionally interacts with the end-effector of the arm. This contact is correctly classified as *intentional* by the robot, and it does not cause variations in the robot's behavior. After reaching the pre-grasp position, the robot brings the end-effector to the grasp position and then to the pre-release position. During this latter phase, the operator interacts with the robot again. However, this time, he interacts with it simulating an *accidental* contact. The robot classifies it as *accidental* and switches from admittance to hand-guiding control. Finally, after a predefined amount of time in which the robot does not recognize any contact, the robot switches back to admittance control and performs the remaining part of the harvesting procedure, i.e., the release of the harvested bunch on the grey box mounted on the front part of the mobile base. Figure 33 shows, in the top part, the time evolution of the norm of the human interaction forces $\|\boldsymbol{F}\|$ and, in the bottom part, the output of the classification algorithm. From this figure, it is possible to notice that the robot is subject to external forces twice: the first one takes place from $t \approx 32$s to $t \approx 40$s and the second one from $t \approx 63$s to $t \approx 65$s. From the bottom part of the figure, instead, it is possible to notice that the output of the classification algorithm is equal to 2 during the first interaction and to 1 during the second interaction, which means that the classifier has correctly classified as *intentional* the first interaction and as *accidental* the second one. The results obtained by the above-described experiment highlight that, despite the high and not-negligible false positive rate of class ac (see Table 3), the classifier, in the case study taken into account, is able to accomplish its work, i.e., distinguishing accidental contacts from intentional ones, in a satisfactory way.
A video of the experiment can be found in Section9 (Collision_detection_Farming_Robot.mp4).

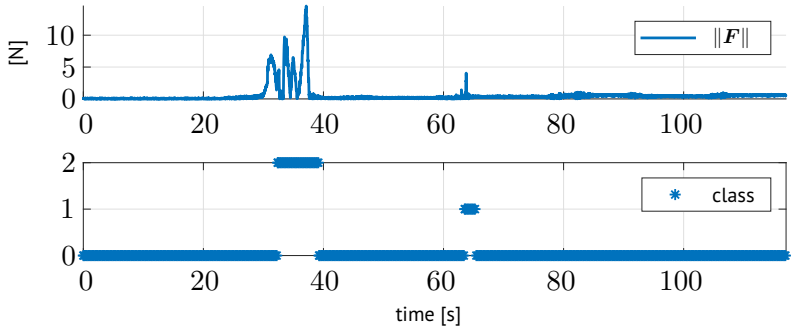

Figure 33: From the top: norm of the human interaction forces detected by the wrench sensor mounted on the left wrist of the robot and output of the classification algorithm. In both plots, the sections associated with an absence of contact, intentional contact, and accidental contact have been highlighted in green, blue and red, respectively.

This project is funded by the European Union G.A No 101016906

7 Conclusions

In this deliverable, the design and testing of the physical interaction functionalities developed for the CANOPIES Farming Robot were described. Several basic interaction strategies were designed by taking into consideration the low level constraints of the platform at hand and the safety of the human operator. To the scope, a method for classifying the type of contact between *intentional* and *accidental* based on machine learning was presented. The envisioned basic physical interaction strategies and the high-level policies in which they are employed were validated via dedicated simulations and indoor experiments on two different platforms. Further experiments in the field are envisaged as part of Task 9.2.

8 Appendix

8.1 Proof of Theorem 1

To prove Theorem 1, let us first derive the closed loop dynamics or robot i. By folding (72) in (4), it holds

$$\boldsymbol{M}_i\dot{\boldsymbol{s}}_i = -\boldsymbol{C}_i\boldsymbol{s}_i - \boldsymbol{K}_s\boldsymbol{s}_i - \Delta\boldsymbol{u}_i + \boldsymbol{h}_i + \boldsymbol{Y}_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\tilde{\boldsymbol{\pi}}_i \tag{85}$$

Now, let us consider the following Lyapunov function

$$V = \frac{1}{2}\sum_{i=1}^{N}\left(\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{M}_i\boldsymbol{s}_i + \frac{1}{k_f}\hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}\hat{\Delta}\boldsymbol{u}_{f,i} + \tilde{\boldsymbol{\pi}}_i^{\mathrm{T}}\boldsymbol{K}_\pi\tilde{\boldsymbol{\pi}}_i\right) \tag{86}$$

By virtue of (85) and Property 1, the time derivative of V is

$$\begin{aligned}\dot{V} = &\sum_{i=1}^{N}\Big(\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{M}_i\dot{\boldsymbol{s}}_i + \frac{1}{2}\boldsymbol{s}_i^{\mathrm{T}}\dot{\boldsymbol{M}}_i\boldsymbol{s}_i + \hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}\tilde{\boldsymbol{h}}_{int,i} - \tilde{\boldsymbol{\pi}}_i^{\mathrm{T}}\boldsymbol{K}_\pi\dot{\hat{\boldsymbol{\pi}}}_i\Big) = \sum_{i=1}^{N}\Big(-\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{K}_s\boldsymbol{s}_i + \frac{1}{2}(\dot{\boldsymbol{M}}_i - 2\boldsymbol{C}_i)\boldsymbol{s}_i + \boldsymbol{s}_i^{\mathrm{T}}(\boldsymbol{h}_i - \Delta\boldsymbol{u}_i)\\ &+ \hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}\tilde{\boldsymbol{h}}_{int,i} - \tilde{\boldsymbol{\pi}}_i^{\mathrm{T}}(\boldsymbol{K}_\pi\dot{\hat{\boldsymbol{\pi}}}_i - \boldsymbol{Y}_i^{\mathrm{T}}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\boldsymbol{s}_i)\Big) = \sum_{i=1}^{N}\Big(-\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{K}_s\boldsymbol{s}_i + \hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}\tilde{\boldsymbol{h}}_{int,i} - \boldsymbol{s}_i^{\mathrm{T}}(\boldsymbol{e}_{int,i} + k_f\hat{\Delta}\boldsymbol{u}_{f,i} + \frac{1}{N}{}^{i}\tilde{\boldsymbol{h}}_h)\\ &- \kappa_i(t)\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{s}_i - \tilde{\boldsymbol{\pi}}_i^{\mathrm{T}}(\boldsymbol{K}_\pi\dot{\hat{\boldsymbol{\pi}}}_i - \boldsymbol{Y}_i^{\mathrm{T}}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\rho}_i, \dot{\boldsymbol{\rho}}_i)\boldsymbol{s}_i)\Big)\end{aligned} \tag{87}$$

Given the parameters update law in (74), (87) simplifies to

$$\dot{V} = \sum_{i=1}^{N}\Big(-\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{K}_s\boldsymbol{s}_i + \hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}\tilde{\boldsymbol{h}}_{int,i} - \kappa_i(t)\|\boldsymbol{s}_i\|^2 - (\tilde{\boldsymbol{\zeta}}_i + \hat{\Delta}\boldsymbol{u}_{f,i})^{\mathrm{T}}(\tilde{\boldsymbol{h}}_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i}) + \frac{1}{N}\boldsymbol{s}_i^{\mathrm{T}i}\tilde{\boldsymbol{h}}_h\Big) \tag{88}$$

Thus, by choosing

$$\kappa_i(t) > \frac{\|\tilde{\boldsymbol{\zeta}}_i\|}{\|\boldsymbol{s}_i\|^2}\|\tilde{\boldsymbol{h}}_{int,i} + \hat{\Delta}\boldsymbol{u}_{f,i}\|$$

it holds

$$\dot{V} \leq \sum_{i=1}^{N}\big(-\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{K}_s\boldsymbol{s}_i - \hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}\hat{\Delta}\boldsymbol{u}_{f,i} + \frac{1}{N}\|\boldsymbol{s}_i^{\mathrm{T}i}\|\|\tilde{\boldsymbol{h}}_h\|\big)$$

From Lemma 2, $\|\tilde{\boldsymbol{h}}_h\|$ converges to the origin after a finite time T_h; then after this time it holds

$$\dot{V} \leq \sum_{i=1}^{N}\big(-\boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{K}_s\boldsymbol{s}_i - {}^{i}\hat{\Delta}\boldsymbol{u}_{f,i}^{\mathrm{T}}{}^{i}\hat{\Delta}\boldsymbol{u}_{f,i}\big)$$

which implies that $\dot{V}$ is semi-negative definite and, consequently, that V is bounded. By leveraging the boundedness of V and, then, of $\boldsymbol{s}_i$, $\hat{\Delta}\boldsymbol{u}_{f,i}$ and $\tilde{\boldsymbol{\pi}}_i$, it can be easily shown the $\ddot{V}$ is bounded as well. Thus, by virtue of Barbalat's lemma, $\dot{V}$ is uniformly continuous and converges to the origin, as well as $\boldsymbol{s}_i$ and $\hat{\Delta}\boldsymbol{u}_{f,i} = k_f\int_{t_0}^{t}\tilde{\boldsymbol{h}}_{int,i}\,d\tau$ (and, therefore, by definition $\tilde{\boldsymbol{h}}_{int,i}$). The main implication of the latter is that, since

This project is funded by the European Union G.A No 101016906

because of Lemma 2, $\tilde{\boldsymbol{h}}_{int,i}$ converges to the origin in finite time (that is the internal wrenches estimated via observer converges to the real one), then also $\boldsymbol{e}_{int,i}, \forall\, i$, converges to the origin.
In view of the expression of $\boldsymbol{s}_i$ in (70) and since $\hat{\Delta}\boldsymbol{u}_{f,i}$ converges to the origin, it follows

$$\dot{\boldsymbol{e}}_{x,i} + k_p \boldsymbol{e}_{x,i} = -\hat{\Delta}\boldsymbol{u}_{f,i}$$

which represents an asymptotically stable system (in the state variable $\boldsymbol{e}_{x,i}$) with vanishing input $-\hat{\Delta}\boldsymbol{u}_{f,i}$. Therefore, $\boldsymbol{e}_{x,i}$ asymptotically converges to the origin. Based on the expression of $\boldsymbol{e}_{x,i}$ in (69), it asymptotically holds

$$({}^i\hat{\boldsymbol{x}}_v - \boldsymbol{x}_i) + k_c \int_{t_0}^{t} \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_i) d\tau \to \boldsymbol{0}_p \tag{89}$$

Let us now introduce the object trajectory estimate error ${}^i\tilde{\boldsymbol{x}}_v = \boldsymbol{x}_v - {}^i\hat{\boldsymbol{x}}_v \in \mathbb{R}^p$ and the object trajectory tracking error $\boldsymbol{e}_{v,i} = \boldsymbol{x}_v - \boldsymbol{x}_i \in \mathbb{R}^p$. From (89), it asymptotically holds

$$(\boldsymbol{x}_v - \boldsymbol{x}_i) + k_c \int_{t_0}^{t} \sum_{j \in \mathcal{N}_i} (\boldsymbol{x}_j - \boldsymbol{x}_v - \boldsymbol{x}_i + \boldsymbol{x}_v) d\tau = -{}^i\tilde{\boldsymbol{x}}_v$$

which can be rewritten as

$$\boldsymbol{e}_{v,i} + k_c \int_{t_0}^{t} \sum_{j \in \mathcal{N}_i} (-\boldsymbol{e}_{v,j} + \boldsymbol{e}_{v,i}) d\tau = -{}^i\tilde{\boldsymbol{x}}_v \tag{90}$$

By denoting with $\tilde{\boldsymbol{x}}_v \in \mathbb{R}^{Np}$ and $\boldsymbol{e}_v \in \mathbb{R}^{Np}$ the stacked vectors of the errors ${}^i\tilde{\boldsymbol{x}}_v$ and $\boldsymbol{e}_{v,i}$, respectively, (90) leads to

$$\boldsymbol{e}_v(t) = -k_c (\boldsymbol{L} \otimes \boldsymbol{I}_p) \int_{t_0}^{t} \boldsymbol{e}_v \, d\tau - \tilde{\boldsymbol{x}}_v \tag{91}$$

in which, from (67), $\tilde{\boldsymbol{x}}_v$ converges to the origin in finite-time. Finally, since the communication graph is connected, the immediate consequence of (91) is that $\int_{t_0}^{t} \boldsymbol{e}_{v,i} = \int_{t_0}^{t} \boldsymbol{e}_{v,j}, \forall\, i, j$ which, based on (69) and (89), implies that $\boldsymbol{e}_{v,i} = \boldsymbol{0}_p \; \forall\, i$. This completes the proof.

This project is funded by the European Union G.A No 101016906

9 Reference to web content

Title	Description	URL
Admittance.mp4	Video showing the admittance experiment described in Section 6.1.1	Link
Hand_guiding.mp4	Video showing the hand-guiding experiment described in Section 6.1.2	Link
Cooperative_harvesting.mp4	Video showing the cooperative harvesting experiment described in Section 6.2	Link
Collision_detection_jaco.mp4	Video showing the collision detection experiment described in Section 6.3.1	Link
Collision_detection_Farming_Robot.mp4	Video showing the collision detection experiment described in Section 6.3.2	Link

10 References

Id	Title	Date
[D2.1]	Requirements, Specifications and Benchmarks	2021-04-30
[D2.2]	Specifications and KPIs for the two CANOPIES robot prototypes	2021-06-30
[D2.3]	Dimensioning of the Real-World Scenario for Final Demo	2021-10-29
[D3.2]	First Release of Farming and Logistic Robot Prototypes	2023-01-24
[D4.1]	Agronomic target modelling, detection and localization	2023-06-30
[D5.1]	Cooperative arms and mobile base functionalities	2023-02-28

References

[1] A. Albini and G. Cannata. Pressure distribution classification and segmentation of human hands in contact with the robot body. *Int. J. Robot. Res.*, 39(6):668–687, 2020.

[2] P. K. Artemiadis, P. T. Katsiaris, M. V. Liarokapis, and K. J. Kyriakopoulos. Human arm impedance: Characterization and modeling in 3d space. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3103–3108, 2010.

[3] Erlend A. Basso and Kristin Y. Pettersen. Task-priority control of redundant robotic systems using control lyapunov and control barrier function based quadratic programs, 2020. 21st IFAC World Congress.

[4] A. Bhole, F. Ficuciello, A. Mashayekhi, S. Strano, M. Terzo, L. Villani, and B. Siciliano. Online estimation of impedance parameters for a variable impedance controlled robotic manipulator. In *Advances in Italian Mechanism Science*, pages 267–274. Springer International Publishing, 2019.

[5] N. Briquet-Kerestedjian, A. Wahrburg, M. Grossard, M. Makarov, and P. Rodriguez-Ayerbe. Using neural networks for classifying human-robot contact situations. In *European Control Conf.*, pages 3279–3285, 2019.

[6] Etienne Burdet, David W Franklin, and Theodore E Milner. *Human robotics: neuromechanics and motor control*. MIT press, 2013.

[7] S. Calinon, T. Alizadeh, and D. G. Caldwell. On improving the extrapolation capability of task-parameterized movement models. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 610–616, 2013.

[8] F. Chen, Y. Cao, and W. Ren. Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Trans. Autom. Control*, 57(12):3169–3174, 2012.

[9] G. Cheng, E. Dean-Leon, F. Bergner, J. Rogelio Guadarrama Olvera, Q. Leboutet, and P. Mittendorfer. A comprehensive realization of robot skin: Sensors, sensing, control, and applications. *Proceedings of the IEEE*, 107(10):2034–2051, 2019.

[10] R. Colbaugh, H. Seraji, and K. Glass. Direct adaptive impedance control of robot manipulators. *J. Robot. Syst.*, 10(2):217–248, 1993.

[11] G. Coll-Ribesa, I. J. Torres-Rodríguezа, 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*, 2023.

[12] A. De Luca and F. Flacco. Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In *IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, pages 288–295, 2012.

[13] F. Dimeas and N. Aspragathos. Online stability in human-robot cooperation with admittance control. *IEEE Trans. Haptics*, 9(2):267–278, 2016.

This project is funded by the European Union G.A No 101016906

[14] M. S. Erden and A. Billard. Hand impedance measurements during interactive manual welding with a robot. *IEEE Trans. Robot.*, 31(1):168–179, 2015.

[15] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

[16] Junjie Fu and Jinzhi Wang. Observer-based finite-time coordinated tracking for general linear multi-agent systems. *Automatica*, 66:231–237, 2016.

[17] M. Geravand, F. Flacco, and A. De Luca. Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. In *IEEE Int. Conf. Robot. Autom.*, pages 4000–4007, 2013.

[18] G. Golluccio, G. Gillini, A. Marino, and G. Antonelli. Robot dynamics identification: A reproducible comparison with experiments on the kinova jaco2. *IEEE Robot. Autom. Mag.*, 2020.

[19] S. Haddadin, A. De Luca, and A. Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Trans. Robot.*, 33(6):1292–1312, 2017.

[20] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Inc., 1996.

[21] Young Jin Heo, Dayeon Kim, Woongyong Lee, Hyoungkyun Kim, Jonghoon Park, and Wan Kyun Chung. Collision detection for industrial collaborative robots: a deep learning approach. *IEEE Robotics and Automation Letters*, 4(2):740–746, 2019.

[22] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robot. Autom.*, 3(1):43–53, 1987.

[23] Sanghyun Kim, Keunwoo Jang, Suhan Park, Yisoo Lee, Sang Yup Lee, and Jaeheung Park. Continuous task transition approach for robot controller based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 4(2):1603–1610, 2019.

[24] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 25(5):581–603, 2011.

[25] A. Kouris, F. Dimeas, and N. Aspragathos. A frequency domain approach for contact type distinction in human–robot collaboration. *IEEE Robot. Autom. Lett.*, 3(2):720–727, 2018.

[26] B. Lacevic, P. Rocco, and A. M. Zanchettin. Safety assessment and control of robotic manipulators using danger field. *IEEE Trans. Robot.*, 29(5):1257–1270, 2013.

[27] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[28] Y. Li and S. S. Ge. Human–robot collaboration based on motion intention estimation. *IEEE/ASME Trans. Mechatronics*, 19(3):1007–1014, 2014.

[29] Yanan Li and Shuzhi Sam Ge. Force tracking control for motion synchronization in human-robot collaboration. *Robotica*, 34(6):1260–1281, 2016.

This project is funded by the European Union G.A No 101016906

[30] M. Lippi and A. Marino. Cooperative object transportation by multiple ground and aerial vehicles: Modeling and planning. In *IEEE Int. Conf. Robot. Autom.*, pages 1084–1090, 2018.

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

[32] A. De Luca and R. Mattone. Sensorless robot collision detection and hybrid force/motion control. In *IEEE Int. Conf. Robot. Autom.*, pages 999–1004, 2005.

[33] O. P. Malik, G. S. Hope, and S. J. Cheng. Some issues on the practical use of recursive least squares identification in self-tuning control. *Int. J. of Control*, 53(5):1021–1033, 1991.

[34] A. Marino. Distributed adaptive control of networked cooperative mobile manipulators. *IEEE Trans. Control Syst. Technol.*, 26(5):1646–1660, 2018.

[35] Alessandro Marino and Francesco Pierri. A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots. *Robotics and Autonomous Systems*, 103:122–133, 2018.

[36] M. Mesbahi and M. Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[37] H. Modares, I. Ranatunga, F. L. Lewis, and D. O. Popa. Optimized assistive human–robot interaction using reinforcement learning. *IEEE Trans. Cybern.*, 46(3):655–667, 2016.

[38] Md. Mozasser Rahman, Ryojun Ikeura, and Kazuki Mizutani. Investigation of the impedance characteristic of human arm for development of robots to cooperate with humans. *JSME Int. J. Series C Mechanical Systems, Machine Elements and Manufacturing*, 45(2):510–518, 2002.

[39] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

[40] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, and B. Siciliano. Haptic-based shared-control methods for a dual-arm system. *IEEE Robot. Autom. Lett.*, 3(4):4249–4256, Oct 2018.

[41] Abdel-Nasser Sharkawy and Nikos Aspragathos. Human-robot collision detection based on neural networks. *Int. J. Mechanical Engineering and Robotics Research*, 7(2):150–157, 2018.

[42] Abdel-Nasser Sharkawy, Panagiotis N Koustoumpardis, and Nikos Aspragathos. Neural network design for manipulator collision detection based only on the joint position sensors. *Robotica*, 38(10):1737–1755, 2020.

[43] N. Rao Sripada and D. Grant Fisher. Improved least squares identification. *Int. J. of Control*, 46(6):1889–1913, 1987.

[44] T. Tsumugiwa, R. Yokogawa, and K. Hara. Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task. In *IEEE Int. Conf. Robot. Autom.*, volume 1, pages 644–650 vol.1, 2002.

This project is funded by the European Union G.A No 101016906

[45] Qilong Yuan and I-Ming Chen. Localization and velocity tracking of human via 3 imu sensors. *Sensors and Actuators A: Physical*, 212:25–33, 2014.