Combining Manipulability and Fatigue for Multi-Metric Authority Arbitration of Physical Human-Robot Collaboration: A Proof of Concept

Álvaro Gil Andrés: 4949781

Tutors:
Prof. dr. ir. David Abbink
Dr. Luka Peternel
Dr. ir. Niek Beckers

Combining Manipulability and Fatigue for Multi-Metric Authority Arbitration of Physical Human-Robot Collaboration: A Proof of Concept

by

Álvaro Gil Andrés: 4949781

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday December 13, 2021 at 10:30 AM.

Student number:	4949781	
Thesis Tutors:	Prof. dr. ir. D.A. (David) Abbink,	TU Delft
	Dr. L. (Luka) Peternel,	TU Delft
	Dr. ir. N. (Niek) Beckers,	TU Delft

This thesis is confidential and cannot be made public until December 31, 2021.

TU Delft

Contents

Combining Manipulability and Fatigue for Multi-Metric Authority Arbitration of Physical Human-Robot Collaboration: A Proof of Concept

Álvaro Gil Andrés
Supervisors: Niek Beckers, David Abbink, and Luka Peternel

***Abstract*—Human-robot interaction is a growing field that aims to research and develop communication channels between humans and robots to enhance comfort, safety, and productivity in healthcare, the household, and the industry. Researchers have considered ergonomy-related metrics to compose these channels for physical human-robot collaborative scenarios. We refer to these communication channels as arbitration methods. Several of these metrics, such as human arm manipulability and muscle fatigue, have taken their turns in the literature to set the base for arbitration methods reaching promising results. Human arm force manipulability represents the transmission between joint torques in the joint space and end-point force in the task space depending on the configuration of the joint angles. Muscle fatigue keeps track of the muscle activation and builds up depending on the muscle activation level and previous fatigue value. The first one has predictive value. The other has a reactive value.**

Nevertheless, no work in the literature explores the power of combining both metrics into an arbitration method. Here we develop a multi-metric arbitration method that combines human arm force manipulability and muscle fatigue as input for a finite state machine (FSM) that translates the human multi-metric state to robot control level over a collaborative task. Although some modifications may be worth trying and evaluating to reach generalizability in physical human-robot collaborative tasks, the system reached satisfactory results. Moreover, as future steps, we should conduct human-factors research to compare the effect of the system on task performance.

***Index Terms*—IEEE, IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

OVER the last decades, the concept of robots and their relationship with humans has been evolving, aiming to tackle increasingly complex applications to enhance productivity and safety in most facets or our lives, including the industry, our households, and healthcare. To achieve such productive and safe application of robotics, designing for proper human-robot interaction is essential. Human-robot interaction aims to investigate intuitive means of interaction and communication between a robot and a human and maximize their performance, efficiency, and applicability as a coupled team [1]. Physical human-robot interaction concerns the interaction through physical contact.

In collaborative scenarios, the human and the robot work contribute to a common task, depending on their skillset [2]. Typically, humans are good at adapting to unknown and changing environments and task requirements, and have manual dexterity lacking in robots. Robots, on the other hand, are able to perform tasks with higher precision, while not being impacted by fatigue. Agravante et al. [3], for example, developed a framework for combining the strengths of a human and robot in for a collaborative tasks. In their task, the human and robot had to work together to keep a ball on a flat horizontal board. The human can choose where the board has to go or how to get there. The robot follows the human motion, calculating a trajectory that minimizes the required energy for the motion and, therefore, helps keeping the ball on the board.

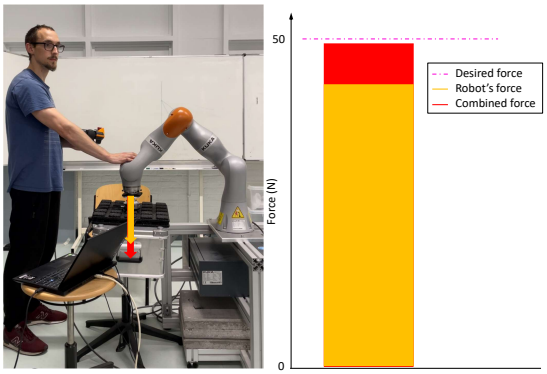

Fig. 1. Representation of the task: the human and robot collaboratively exert a constant force against a horizontal surface. To the right we can see a graph representing the desired force level of the task (pink), the robot's end-point force (orange), and the combined exerted force (red)

Arbitration is defined as the mechanism that assigns the control of the (part of the) task to either the human or the robot [1]. Arbitration determines the interaction strategies between human and robot, such as teacher-student, supervisor-subordinate, or leader-follower to fit the task and skills of the human and robot [4, 1]. In the board game example, the human is the primary leader and the robot the follower, yet the robot can increase its assistance to keep the ball on the board when needed. In such a case, the task is divided into sub-tasks, and the arbitration mechanism will interpret the sensed information from the human and robot, estimate the human's intention, and

Author and supervisors are with the department of Cognitive Robotics, faculty of Mechanical, Maritime, and Material Engineering, Delft University of Technology.

choose the robot's action and timing. i.e., *Now it is time to help in this way and to this grade*. Pehlivan et al. [5] developed a method for rehabilitation that also fits as an example for this kind of arbitration. Utilizing the interaction forces between the human and the rehabilitation robot when performing a predefined motion, they derived the user's capability at each time step. The user's capability is then used to arbitrate the amount of force the robot exerts to perform the motion. In other words, their method uses the user's capability estimation to adjust the level of assistance the robot provides.

How roles and tasks are arbitrated between human and robot can be based on many sets of metrics representing different kinds of information. The type of information often depends on the task. Of particular interest for industrial human-robot collaboration are ergonomy metrics in collaborative scenarios, such as human arm manipulability and muscle fatigue [6]. The velocity and force manipulability ellipsoids, introduced by Yokishawa [7] in robotics, represent a quantitative metric of the velocity and force, respectively, with which the end effector of an arm can move in every direction depending on the joint space configuration. Jacquier-Bret et al. [8] borrowed manipulability from the robotics field and applied it for human arm movement evaluation. Gopinathan et al. [9] introduced the concept in a physical human-robot interaction scenario. They studied the effects caused by the variation of task parameters calculating manipulability on different simplified human arm models.

Peternel et al. [10] made the step towards introducing manipulability in robot control. They developed a control method for a physical human-robot co-manipulation and handover scenario using a human body model for ergonomic optimization with human arm manipulability properties as a constraint. They set the manipulability limits so the algorithm would find positions that keep isometric manipulability while optimizing for minimum joint torques. i.e., the positions for which the manipulability ellipsoid has equal main axes and minimum joint torques. This way, the robot ensures to handle or deliver the manipulated object within a range of positions the human can optimally manipulate the object in whatever his intended direction is. Petrič et al. [11] used human arm force manipulability directly as a supervisor-subordinate arbitration metric for a human arm exoskeleton for power augmentation. They made the robot compensate for the minor ellipsoid axes, turning manipulability into isometric manipulability in a wide range of arm positions. [12] discovered that arm muscles recruitment and activation is strongly related to human arm manipulability, and so is muscle fatigue induction. [13, 14, 15] concluded that a fatigued muscle is more likely to get injured. Therefore, using human arm manipulability for arbitration is an approach to prevent the muscles from getting fatigued and risk injury.

Muscle fatigue represents an interesting metric with a significant presence in the literature, with varying computational models of fatigue. Some models use the externally applied force (e.g., on an object by the human) as input to calculate a fatigue-related measure [15]. Others developed a fatigue model that infers the force generated by human muscles based on physiological muscle motor unit behavior [16]. Maurice et al. [17] used fatigue for ergonomy evaluation in a human-robot collaboration scenario, exploring the utility of such a metric in the field of focus for this research. They developed a dynamic human model consisting of a rigid-body model that asses for postural risk, physical effort, and consumed energy. With this, they simulated the subjects performing the evaluated tasks and computed the joint-torque derived fatigue for every joint through the estimated torques. Peternel et al. [18, 19] used it as a supervisor-subordinate arbitration metric and computed it through muscle activity measured through electromyography (EMG) signals captured online, [18] for a system in which the robot would take over the task whenever the human fatigue reaches a predetermined threshold. In [19], the robot modifies and adapts the working configuration to the fatigue of the involved muscles. The EMG-based fatigue models proved to be advantageous over the force or torque-based models in many aspects. While both present similar dynamics, the EMG-based fatigue models allow for accurate muscle-specific fatigue estimation without the need for complex biomechanical models and dimensional reduction methods. Another advantage is that they do not require expensive force sensors.

In all the aforementioned studies (all but [16]), the fatigue model constitutes a first-order differential equation system, behaving like a (leaky) integrator of either force measurements or physiological signals (EMG) as input. Moreover, they are generally used for a reactive approach. The robot assists once one of the considered muscles gets to a predetermined level of fatigue, providing time to rest or guide towards a posture change. Therefore, the robot acts to reduce fatigue once fatigue occurs, in opposition to the manipulability-based existing approaches.

Nevertheless, both preventive and reactive approaches present significant limitations. Preventive approaches often fail to succeed since they do not explicitly consider and measure the event they are preventing. If we think about fatigue as the event to prevent, our preventive method will be based on the assumption that fatigue happens under certain conditions and our actions prevent it. On the other hand, reactive approaches can often be too late. As we mentioned above, fatigue is directly related to the probability of injury. The existing fatigue models used for physical human-robot collaboration constitute a practical tool, yet they still approximate the actual fatigue level and depend on the previous calibration process.

Given that muscle activation and recruitment are highly related to the manipulability ellipsoid, a combination of both approaches would be preferable. Thus, assisting in preventing fatigue, keeping track, and assisting when fatigue cannot be prevented from happening. Nevertheless, an approach with these features is missing in the literature. The problem is that combining these two metrics requires a method that can successfully arbitrate both while resulting in stable behavior. The arbitration method should behave sensibly to the considered metrics. However, it should not provide too variable assistance increasing the task difficulty from the user's perspective.

To address this gap, we propose a multi-metric arbitration method based on human arm force manipulability and muscle fatigue. The proposed method constitutes a finite state machine model with four states and assistance levels

that provide smooth transitions between states. We prove the concept with an abstract, simplified task: *collaboratively exerting a constant force a horizontal surface* and perform four experiments modifying the manipulability and fatigue conditions around the task. With this research, we set the first step towards developing a simple physical human-robot collaboration method suitable for the industry that exploits the advantages of preventive and reactive assistance.

We describe the arbitration algorithm, the manipulability model and its data acquisition, the fatigue model, and data acquisition in the Methods section. The Experiments section describes the designed functionalities of the system and the experiment data to prove them. It explains the variations of the main task that we introduce in the experiments and presents the achieved results. Finally, the discussion section develops the relationship between the results and the experiments, the conclusions, and the suggested future work.

II. Methods

The proposed method uses a collaborative task that involve exerting a constant force for an extended period, but which still requires the human-robot team to change the position of the applied force. For example, a real-world task would be a polishing task. The human and the robot would provide a percentage of the total desired force throughout the task execution, based on how the force production sharing is arbitrated (e.g. more by the human, or more by the robot). The force production arbitration is assigned online depending on the human's muscle fatigue and human arm force manipulability in the direction of the desired force.

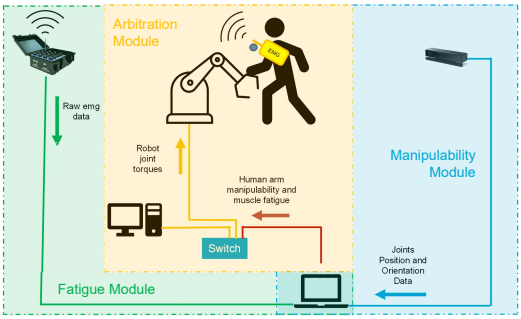

Fig. 2. Schematic representation of the developed network. The system consists of three main components, which communicate through UDP. (Yellow) The arbitration module. It incorporates the arbitration method, based on a finite state machine that translates the fatigue and manipulability values into a percentage of the total force exerted by the robot. (Green) The fatigue module. It incorporates the EMG data acquisition and processing and translates the signal into a fatigue value. (Blue) The Manipulability module. It integrates the joints' position and orientation data acquisition, and It calculates the manipulability value in the direction of the exerted force.

Muscle fatigue indicates the accumulated fatigue level of the selected muscle during the task execution. The human arm force manipulability indicates the transmission level from joint torques to the end-point force of the human arm, depending on the arm pose. As a result, the method will result in a human-robot arbitration in which the robot will reactively assist the human depending on the human fatigue, allowing the human to rest when needed, while still sharing the task when possible (minimal intervention of the robot). Assistance that depends on force manipulability constitutes a preventive approach; the robot assists when the human is in a posture that involves more effort and is less comfortable to prevent fatigue in the long-term. We combine these two metrics to exploit the advantages of preventive (manipulability based) and reactive (fatigue based) approaches. Thus, the robot keeps track of the fatigue level even though its assisting to prevent it, and assists based on fatigue when it cannot be prevented from happening.

An example of an eligible task would be collaboratively polishing a surface. Here, the robot's end-effector and the human would hold the polisher against the surface. Then, the human would swipe the polisher through the surface while exerting a percentage of the force. Depending on the arm pose the arm takes along the target surface and the fatigue level of the involved muscles, the robot would adapt, exerting different force percentages. The human can see through visual feedback the desired force level, and the combined exerted force. This way, the human adapts to the robot's levels of assistance. For this method, we performed an abstraction of this task. The selected task is *collaboratively exerting a constant force level on a horizontal surface*, see Fig 1.

This method consists of three main modules: (1) the multi-metric arbitration module, (2) the manipulability estimation module, and (3) the fatigue estimation module. The multi-metric arbitration module is based on a finite state machine. Depending on a normalized value of human arm force manipulability and muscle fatigue, the finite state machine smoothly converges towards an arbitration value $a \in [0, 1]$ that represents the robot's fraction of its required force production. The normalized human arm force manipulability value emerges from an estimate of the force manipulability ellipsoid, which is measured through human arm pose estimation. Selecting the direction of the applied force, we extract a single manipulability norm that we normalize based on a comparison with known manipulability poses and values. Finally, the fatigue estimate module computes a fatigue level based on EMG measurements of the muscle. The fatigue module distinguishes between two modes: fatigue mode, slowly converging to 1, and recovery mode, slowly converging to 0. We can see an schematic representation of the built system in figure 2.

In figure 2, we can see a scheme of the implemented network. The Kinect V2 and the Delsys Trigno System acquire and send the data to a PC (Lenovo Legion with USB 3.0 connection and Intel Core i7 9^{th} generation, windows 10) through the blue, green channels. The algorithm performs the estimations, coordinates the results, and sends them to the desktop computer through the red channel. The arbitration algorithm and robot control interpret and translate the metrics and communicate with Kuka iiwaa 7 R800 through the yellow

channel. The Kuka iiwaa 7 R800's control loop operates at 200Hz.

A. Arbitration Method

Manipulability and fatigue need to be translated into an arbitration variable a. The arbitration variable a is afterward integrated into an impedance control scheme that translates into the torques for the control system of Kuka iiwaa 7 R800.

The arbitration algorithm is a finite state machine (FSM) composed of five states, see Fig. 3: four states for multi-metric arbitration and a fifth that substitutes the two middle states for single-metric arbitration. The initial state is $a = 0$, when manipulability $m = 0$ and fatigue $f = 0$. The five other states that depend on which interval contains m and f: $[0, 0.3)$, $[0.3, 0.7)$ or $[0.7, 1]$. When m is in the low interval or f is in the high interval $a = 0.9$ meaning the robot exerts 0.9 times the desired force providing maximum assistance. When m and f are in the middle interval $a = 0.6$ providing a robot-dominated intermediate assistance. When m is in the high interval and f in the middle interval, or f is low and m is in the middle interval $a = 0.3$ providing a man-dominated intermediate assistance. When m is high and f is low, or for single-metric arbitration, when f is low or m is high, $a = 0.1$ providing minimum assistance. Finally, for single-metric arbitration, is one of the metrics is 0 and the other is in the middle interval, $a = 0.5$ providing intermediate assistance.

The algorithm contemplates two scenarios: only one input is relevant, or both of them are. If the system is going to be used for a single quick task such as drilling several holes in places that are not reachable in a confortable position, then setting the EMG sensors might be inconvenient, and manipulability-based assistance would be helpful. Three states contemplate the first scenario, leading a to converge to 0.1, 0.5 and 0.9. On the other hand, in an scenario were the considered time is worth setting up the sensors, combining preventive and reactive assistance is preferable. For the second scenario, the algorithm disregards the previous middle state, and two others are considered in its place: 0.3 and 0.6. In figure 3, we can see an scheme with the conditional logic expressions that lead towards each state, with the green color referring to f and the red color referring to m.

$$a(t+dt) = \begin{cases} a_t + (a_0 - a_t) \cdot \left(1 - \frac{1}{1+e^{E(t)}}\right) & \text{if } a(t) > a_t \\ a_0 + (a_t - a_0) \cdot \left(1 - \frac{1}{1+e^{E(t)}}\right) & \text{if } a(t) < a_t \\ a(t) & \text{if } a(t) = a_t \end{cases} \tag{1}$$

where:

$$E(t) = -\frac{\frac{a_0+a_t}{2} - \left(a(t) + \frac{a_t - a_0}{1000}\right)}{0.02} \tag{2}$$

State transitions are smoothed using a sigmoid function (Eq. 1) between the current (a_0) and the target state (a_t). In equation 2 we can see the expanded exponential for equation 1. Here, a_0 represents the initial arbitration, meaning the arbitration of the previous state, and a_t represents the target arbitration, meaning the arbitration to which the current state converges. The minimum time in each state after a transition is set to 5 seconds to avoid fast state transitions and to allow the human to adapt to the new state. The denominator at 0.02 was set to have the sigmoid function fit a 5s transition. Similarly, the denominator 1000 was set so the transition takes 5 seconds in total, since the frequency at which the robot runs is 200Hz. We can see the sigmoid transitions in Fig. 6.

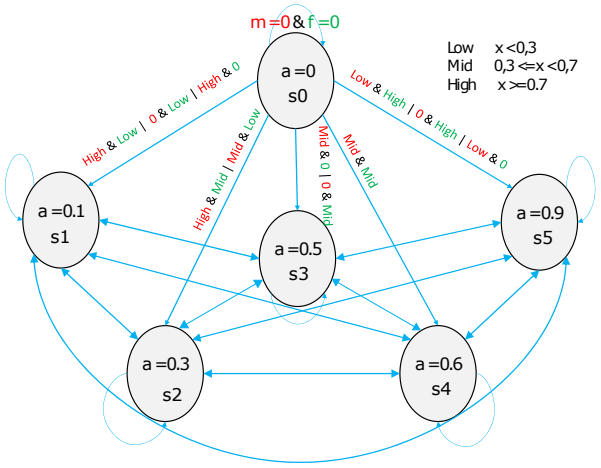

Fig. 3. Schematic representation of the arbitration finite state machine. On the top right corner we can see a legend explaining the intervals for the variables: manipulability m and fatigue f. By the arrows pointing out the top state, we can see the conditionals that lead to each of the states. In red we can see the values or intervals referring to manipulability m. In green we can see the values or intervals referring to fatigue f.

B. Robot Control

The robot uses an impedance controller. In equation 3, we can see the expression of our impedance controller, where $\mathbf{K}$ is the stiffness matrix of the robot, $\mathbf{x_{ref}(t)}$ is the reference end-point state, $\mathbf{x(t)}$ is the current end-point position, $\mathbf{D}$ is the damping matrix of the robot, $\mathbf{v(t)}$ is the end-point velocity and $\mathbf{F_{end}}$ is the end-point force. The control algorithm transforms $\mathbf{F_{end}}$ to joint torques for the torque control of the robot.

$$\mathbf{F_{end}(t)} = \mathbf{K} \cdot (\mathbf{x_{ref}(t)} - \mathbf{x(t)}) + \mathbf{D} \cdot \mathbf{v(t)} \tag{3}$$

Finally, in equation 4 we can see how the control algorithm integrates the arbitration variable $a(t)$ (result of the current state of the FSM) into the impedance controller, adjusting F_{end} accordingly. The control algorithm modifies the z coordinate of the end-point reference position to a maximum of 20 centimeters. Depending on the robot stiffness we select, that modification translates to a determined force when the robot is in contact with a horizontal surface.

$$\mathbf{x_{ref(t)}} = \begin{bmatrix} 0 & -0.5657 & 0.0590 - 0.2 \cdot a(t) & 0 & 0 & 0 \end{bmatrix}^T \tag{4}$$

C. Human Arm Manipulability

The end-point motion of an arm in the task space is the result of joint angle variations. The arm's endpoint manipulability ellipsoid represents this kinematic relationship, capturing

how efficiently joint angle variations propagate towards end-point variations in every direction within the task space. The force manipulability ellipsoid is orthogonal to the velocity manipulability ellipsoid and, similarly, represents how efficiently joint torques are propagated towards end-point forces in every direction within the task space [7, 20].

The manipulability calculation requires the human arm joints configuration in the joint space. Provided the position and orientation of the human arm joints in the task space, several steps lead to the projection from task space to joint space to human arm manipulability. Figure 4 shows these steps: task space data acquisition, human arm triangle space projection [21], joint space projection, and human arm manipulability calculation.

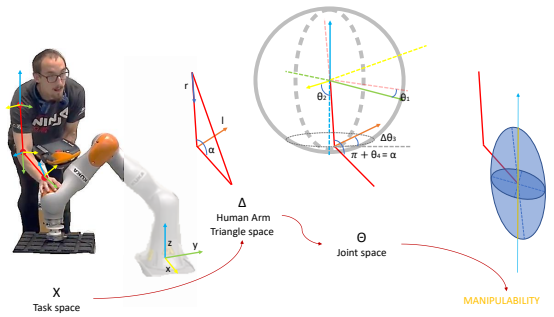

Fig. 4. Scheme of the data transformations connecting the human arm position and orientation data in the task space and the human arm manipulability in the task space. [21]

Given the human arm configuration in the joint space, the jacobian $J(q)$ connects the transmission of the joint velocities and end-point velocities, and $J(q)^{-T}$ connects the joint torques to end-point force transmission. We can see in Eq. 5 the expression for force manipulability ellipsoid for a redundant arm, and in the last term, the matrices resulting from a singular value decomposition.

$$\mathbf{f_M} = (\mathbf{J(q)J(q)}^T)^{-1} = \mathbf{U\Sigma V}^* \tag{5}$$

Here, $\mathbf{\Sigma} \in R^{m\times n}$ is a diagonal matrix composed of the singular values of the force manipulability matrix $f_M \in \mathbb{R}^{m\times n}$, where $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_n \geq 0$. In Eq. 6 we can see a general development of $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$. $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices. In Eq. 7 we can see a general development of $\mathbf{U} \in \mathbb{R}^{m\times m}$, where each entry represents an unitary vector [7, 11].

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \sigma_3 & \\ 0 & & & \sigma_m \quad 0 \end{bmatrix} \in \mathbb{R}^{m\times n}, \tag{6}$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u_1} & \mathbf{u_2} & \mathbf{u_3} & \mathbf{u_m} \end{bmatrix} \in \mathbb{R}^{m\times m} \tag{7}$$

The principal axes of $fM \in R^{mxn}$ are $\sigma_1\mathbf{u_1}, ...\sigma_m\mathbf{u_m}$. $\sigma_1\mathbf{u_1}$ represent the direction in which the end-effector can exert the highest force. Similarly, $\sigma_m\mathbf{u_m}$ represents the direction for the lowest force [7, 20, 11]. Jacquier-Bret et al. [8], when applying the manipulability concept to the human arm, identifies three singular axes. Equation 8 shows the expression of the projections of the axes of the ellipsoid in the global reference frame.

$$\mathbf{U\Sigma}^{\frac{1}{2}} = \begin{bmatrix} u_{1,x}\sqrt{\sigma_1} & u_{2,x}\sqrt{\sigma_2} & u_{3,x}\sqrt{\sigma_3} \\ u_{1,y}\sqrt{\sigma_1} & u_{2,y}\sqrt{\sigma_2} & u_{3,y}\sqrt{\sigma_3} \\ u_{1,z}\sqrt{\sigma_1} & u_{2,z}\sqrt{\sigma_2} & u_{3,z}\sqrt{\sigma_3} \end{bmatrix}, \tag{8}$$

For this research, exerting a constant force against a horizontal surface is the task that serves as a proof of concept. In figure 4, the last data transformation graphically explains the selection of the manipulability values of interest for such a task—the projection of the manipulability ellipsoid in the z-axis of the global reference frame.We can see the expression for such projection in equation 9. The m value is then normalized using a sigmoid function with 0.5 value on m = 1.0283.

$$m = u_{1,z}\sqrt{\sigma_1} + u_{2,z}\sqrt{\sigma_2} + u_{3,z}\sqrt{\sigma_3}, \tag{9}$$

1) Sensor for manipulability estimation: The Kinect V2 sensor acquired the position and orientation arm data in the task space. In figure 5, we can see a picture acquired using the Kinect Studio v2.0 software. It represents the skeleton the Kinect embedded software estimation and user body segments at a sampling frequency of 30Hz. This skeleton was accessed online using the Kinect C++ package. The joint angles needed to calculate the Jacobian are extracted from the Kinect software.

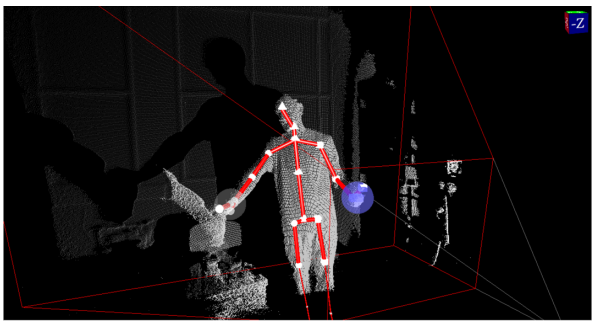

Fig. 5. Graphic representation of the body segment skeleton extracted using the Kinect Studio v2.0 software.

D. Fatigue model

We found the fatigue estimation model by Peternel et al.[18] the most appropriate for the application, because such a model allows for a muscle-specific fatigue estimation through wearable light EMG sensors that do not interfere with task execution. It is suitable for every muscle involved in the task. For instance, this research tests different experiments with different involved muscles providing proof of versatility. In equation 10, we can see the first-order system representing this research's fatigue model [18].

$$\frac{dV_i(t)}{dt} = \begin{cases} (1 - V_i(t))\frac{A_i(t)}{C_{Fi}} & if \quad A_i(t) \geq A_{th} \\ -V_i(t)\frac{R}{C_{Fi}} & if \quad A_i(t) < A_{th} \end{cases}, \tag{10}$$

Here, each equation represents the fatigue increase at the top and recovery at the bottom. The fatigue index of the muscle i $V_i(t)$ increases when the muscle activation level i $A_i(t)$ is above an activation threshold A_{th}. Similarly, the muscle fatigue index decreases when the mentioned muscle activation is below such threshold at a recovery rate R. The speed to which the fatigue induction or recovery happens also depends on the fatigue-related capacity of muscle i C_{Fi}.

$$0 \leq A_i(t) = \frac{EMG_i(t)}{MVC_i} \leq 1, \tag{11}$$

Eq. 11 represents how we obtain the activation level $A_i(t)$ of muscle i, from its EMG signal. First, the fatigue estimation algorithm processes the raw EMG signal. First, the EMG signals are filtered using a second-order Butterworth high-pass filter with a cut-off frequency at 20 Hz. The resulting signal is then rectified and low-pass filtered using a second-order Butterworth low pass filter at 1 Hz, yielding the EMG envelope $EMG_i(t)$. Finally, the EMG envelope is normalized using the maximum-voluntary-contraction of muscle i MVC_i, resulting in the muscle activation $A_i(t)$. The MVC is obtained by exerting as much force with that muscle as possible by pushing against an object and recording its muscle activity.

Equation 12 expresses the fatigue-related capacity of muscle i. We used 0.3 times the MVC_i [18] as the reference effort G_{ref}. T_{end} represents the time the user can endure exerting the reference effort G_{ref}. In order to obtain the MVC_i and T_{end}, we integrated a preliminary experiment in a Jupyter notebook that allows calibrating these values in a straightforward way. Equation 12 also assumes the muscle reaches total capacity at fatigue index $V_i(t) = 0.993$ [18].

$$C_F = -\frac{G_{ref} \cdot T_{end}}{log(1 - 0.993)}, \tag{12}$$

1) Sensor for fatigue estimation: We used the wireless Delsys Trignos System for the data collection of the muscle activity $A_i(t)$ using the muscle's EMG signal.

III. Experiments

Aiming to use the potential of the selected metrics: human arm force manipulability and muscle fatigue; and explore the preventive assistance allowed by the former metric and the reactive assistance allowed by the latter, we designed the proposed method to have three main functionalities: *Modularity*, *Fatigue hysteresis*, and *Conservative behavior*.

Modularity is the ability to correctly function using either of the arbitration metrics: manipulability and fatigue, on their own, and the ability to function with both metrics simultaneously. *Fatigue hysteresis* allows the human to rest for longer when the fatigue level reaches a high threshold. Once the muscle is fatigued, the proposed method disregards the manipulability value and takes over 90% of the force until muscle fatigue decreases below a lower threshold, thus skipping the intermediate states in the multi-metric arbitration. This way, the robot takes a reactive approach and gives time to the human to modify his/her position and rest. *Conservative behavior* prioritizes the metric that is in the worst condition. Thus, when the fatigue level reaches a high threshold, the robot takes over 90% of the force no matter the manipulability value. Similarly, if the manipulability value is below the low threshold, the robot takes over 90% of the force and disregards fatigue. In the latter case, the robot takes a preventive approach. If manipulability is low, the robot aims to minimize the induction of fatigue.

We designed and performed five experiments to validate the modalities of the proposed method. Before getting in contact with the robot, we checked the functionalities in a simulated experiment or *pre-experiment*. The pre-experiment consisted of several simulations with pre-designed values for the arbitration metrics. Then a chain of three experiments tested the manipulability module, the fatigue module, and the combination of both modules. The latter experiment implemented the proof of concept task: *the user and the robot collaboratively exert a constant force against a horizontal surface*. While keeping a constant combined force, the human modifies the human arm posture to test the robot's adaptability to the manipulability value and fatigue accumulation, see Fig. 1. The *high manipulability configuration experiment*, proves one of the affirmations that motivated this research: *even while keeping a good posture, exerting a force induces fatigue*.

A. Pre-experiment

The pre-experiment consisted of 6 simulations with different predefined arbitration metrics combinations and 2 simulations with variations of only one metric. Figure 6 is composed of the simulations performed with one of the metrics increasing from 0 to 1 and decreasing back to 1, while the other metric is constant within two state limits. The column on the left shows the results of simulations in which the manipulability value was constant and within the different intervals, and fatigue increases uniformly from 0 to 1 in 50 seconds. The last row of the first column keeps the manipulability value set to 0 and increases fatigue. Similarly, in the column on the right, fatigue keeps a constant value within the different intervals, and manipulability increases.

B. Manipulability experiment

Once we had validated the arbitration algorithm on the pre-experiment, the *manipulability experiment* took place. For this experiment, the subject stood in front of the Kinect v2 sensor at a distance of 1.7 meters and with the line of the shoulders at an angle of approximately 45º with respect to the baseline of the sensor. In this position the subject performed a sequence of three poses. These poses are representative of high manipulability, medium manipulability, and low manipulability. We previously tested the poses exerting a vertical force without the robot to have a qualitative perception of the manipulability in each pose. The reasoning behind the manipulability levels is explained in the *multi-metric experiment* subsection. The *high manipulability pose* consists of an orthostatic body position, with the right arm pointing downward and no shoulder rotation. We can see it in figure 8, in the top row, being the high manipulability pose at the first, fifth, and last picture. We can see the *medium manipulability pose* in figure 8, in

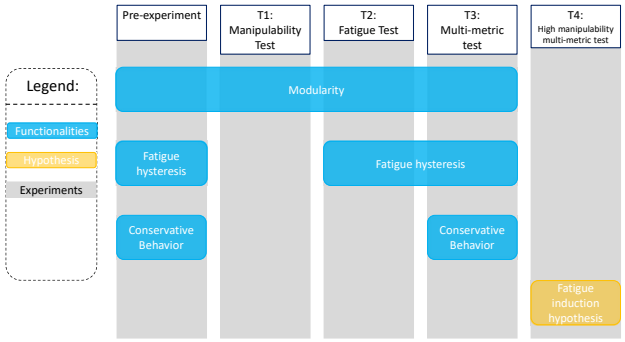

Fig. 7. Scheme of the performed experiments and tested functionalities. (Blue) Functionalities. (Orange) Hypothesis. We performed five experiments to test the proposed method.(Pre-experiment) First, a simulated experiment or pre-experiment proved the designed functionalities (modularity, fatigue hysteresis and conservative behavior) to work correctly before performing tests in direct contact with the robot. (T1) Then, a manipulability test in which the user performs a sequence of movements to test the manipulability module. (T2) Then, a fatigue test is conducted in which the user evaluates the fatigue module and the fatigue hysteresis. (T3) This is followed by a multi-metric test that assesses the multi-metric module, the fatigue hysteresis functionality, and the conservative behavior functionality. (T4) Finally, a multi-metric test with constant high manipulability is performed to check the hypothesis that motivates the proposed method:*even while keeping a high manipulability pose, exerting a force induces fatigue*.

the top row, the second, and the fourth picture. It consists of an orthostatic pose with 90º forward shoulder flexion and 90º internal shoulder rotation and 90º elbow flexion. The *low manipulability pose* consisted of an orthostatic pose with 90º shoulder flexion and 90º internal shoulder rotation with the extended elbow. We can see this in figure 8 in the third and sixth pictures in the top row.

The experiment took two minutes per repetition and a total of eight repetitions. We can also see in figure 8 some of the results of the described experiment. The second row represents the manipulability value m, the fatigue value f, and the arbitration value. The third row represents the calculated robot exerted force, and the measured exerted force.

C. Fatigue experiment

Then, we validated the fatigue module. For this, we designed an experiment that consisted of intermittently lifting a load for 5 and 10 seconds to achieve two different levels of fatigue. As a weight, we used an aluminum frame from the laboratory and previously tested that the required effort to lift it was high enough to trigger fatigue. We also developed an online fatigue visualization algorithm through the pyqtgraph python library to see the effect of different motions and efforts on the fatigue level and check its correct behavior.

We chose the anterior deltoid muscle for the fatigue experiment. The subject first performed a short calibration pre-experiment to fit the model to the electrical behavior and features of the chosen muscle. The experiment took 4 repetitions of 5 minutes. We set the recovery rate R parameter to 0.1 for two repetitions and 0.5 for the rest. This way, the user could pick the recovery rate that felt more natural for the muscle.

In figure 9 we can see some of the results of this experiment, performed with the selected recovery rate $R = 0.1$. In the

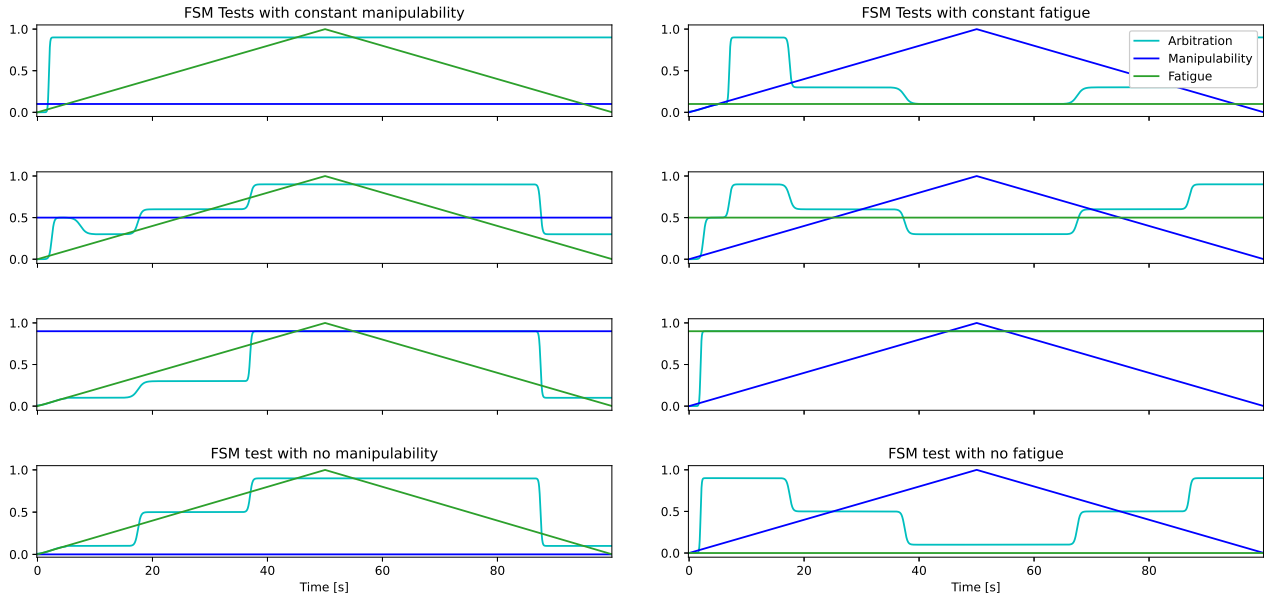

Fig. 6. Arbitration value when (0,0) manipulability $m < 0.3$ and fatigue f is uniformly increasing and decreasing. (0,1) $0.3 < m < 0.7$ and f is uniformly increasing and decreasing. (0,2) $m > 0.7$ and f is uniformly increasing and decreasing. (0,3) $m = 0$ and fatigue is uniformly increasing and decreasing. (1,0) $f < 0.3$ and m is uniformly increasing and decreasing. (1,1) $0.3 < f < 0.7$ and m is uniformly increasing and decreasing. (1,2) $f > 0.7$ and m is uniformly increasing and decreasing. (1,3) $f = 0$ and m is uniformly increasing and decreasing.

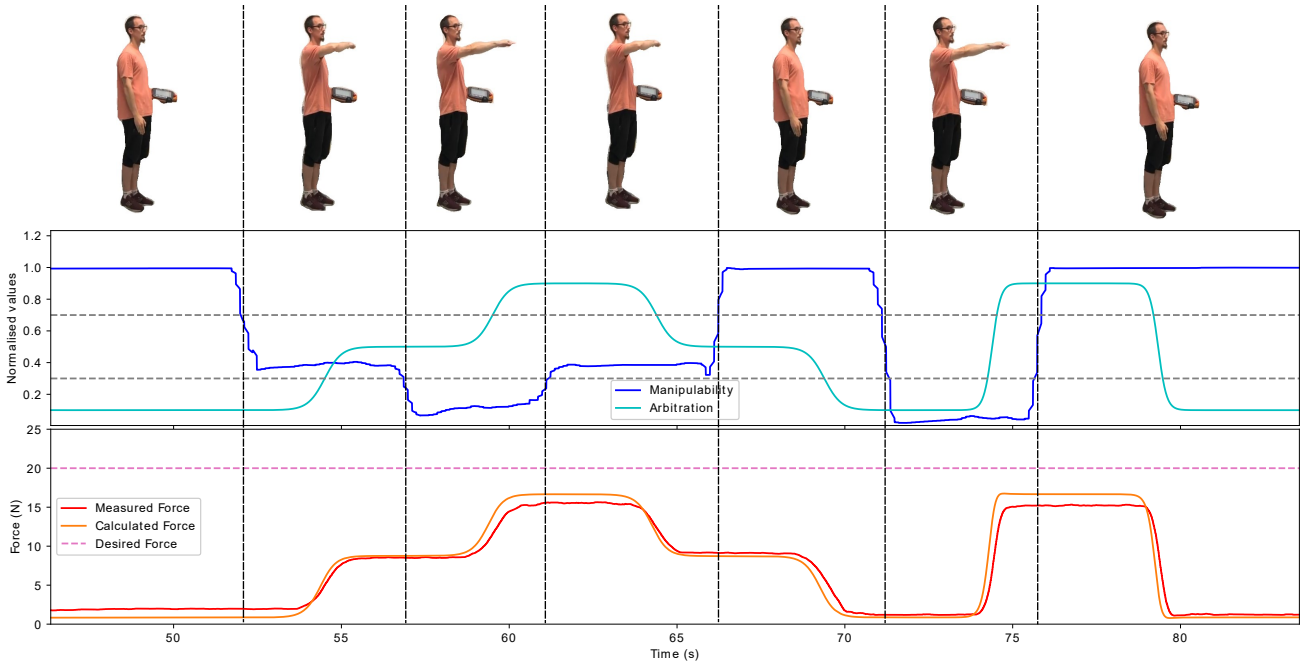

Fig. 8. Results plot for the manipulability experiment together with images of the poses that were performed in each moment of the test. (0,0) high manipulability pose (HMP), medium manipulability pose (MMP), low manipulability pose (LMP), MMP, HMP, LMP and HMP. (1,0) Manipulability m, fatigue f, and arbitration a vs time (s). (2,0) Calculated robot force and Measured force vs time (s).

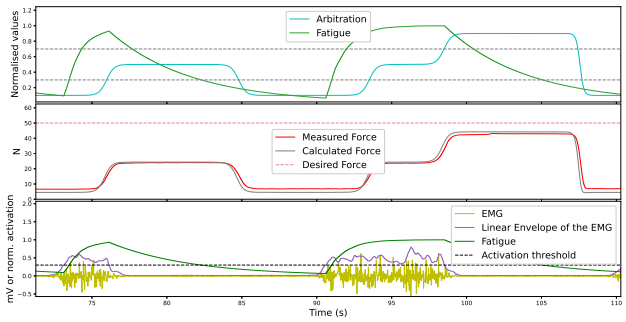

Fig. 9. Results plot for the fatigue experiment. (0,0) Fatigue value f and arbitration value a vs time(s).(1,0) Calculated robot force and measured exerted force vs time (s). (2,0) Raw EMG signal, normalised EMG's linear envelope and fatigue f vs time(s).

first row, we can see the arbitration value a and the estimated fatigue value f. In the second row, we can see the calculated robot force and the measured exerted force. Finally, in the third row, we can see represented the raw EMG signal and processed normalized linear envelope of the EMG signal and the estimated fatigue f.

D. Multi-metric experiment; exerting a constant force

The *multi-metric experiment* consisted of the subject changing poses of known manipulability value around the robot while collaboratively exerting a force of 50N. We placed the EMG sensor on the latissimus dorsi muscle of the subject and tuned the fatigue model parameters to $MVC = 0.09mV$ and $T_{end} = 40$ seconds. For this test, we did not perform a calibration pre-experiment. Instead, we tuned the model to be more sensitive than usual to achieve a faster system response and check the state transitions quicker online. The subject repeated the experiment 8 times, with 4 minutes per repetition. Same as for the *fatigue experiment*, we set the recovery rate $R = 0.5$ for half of the takes and $R = 0.1$ for the other half and $R = 0.1$ was preferred.

The succession of poses kept similar criteria to the *manipulability experiment*. While exerting a vertical force with the right hand on the back of the robot's end-effector, the subject went through a *low manipulability pose*, two middle manipulability poses (*medium manipulability pose* and *sideways medium manipulability pose*), and a *high manipulability pose*. We can see the sequence of poses in the first raw of figure 10.

The *low manipulability pose* consisted of the user in an orthostatic position in front of the robot. Due to the height difference between the subject and the robot, the user's arm only flexes slightly on the elbow while pushing the end-effector downward. In this position, the difference on the principal axes of the manipulability ellipsoid is significant, meaning there is a very high manipulability in the long axis direction. However, the moment the user exerts force in a different direction, the manipulability quickly drops. The middle manipulability poses consisted of the user squatting on a chair by the robot.

The *medium manipulability pose*, as we can see in the second picture of the first row in figure 10, was a squat facing the robot's end-effector, with the elbow 90º flexed and the shoulder 90º rotated inward. With the 90º elbow flexion, the difference between the principal axes of the manipulability ellipsoid decreases. The ability to produce force is similar in

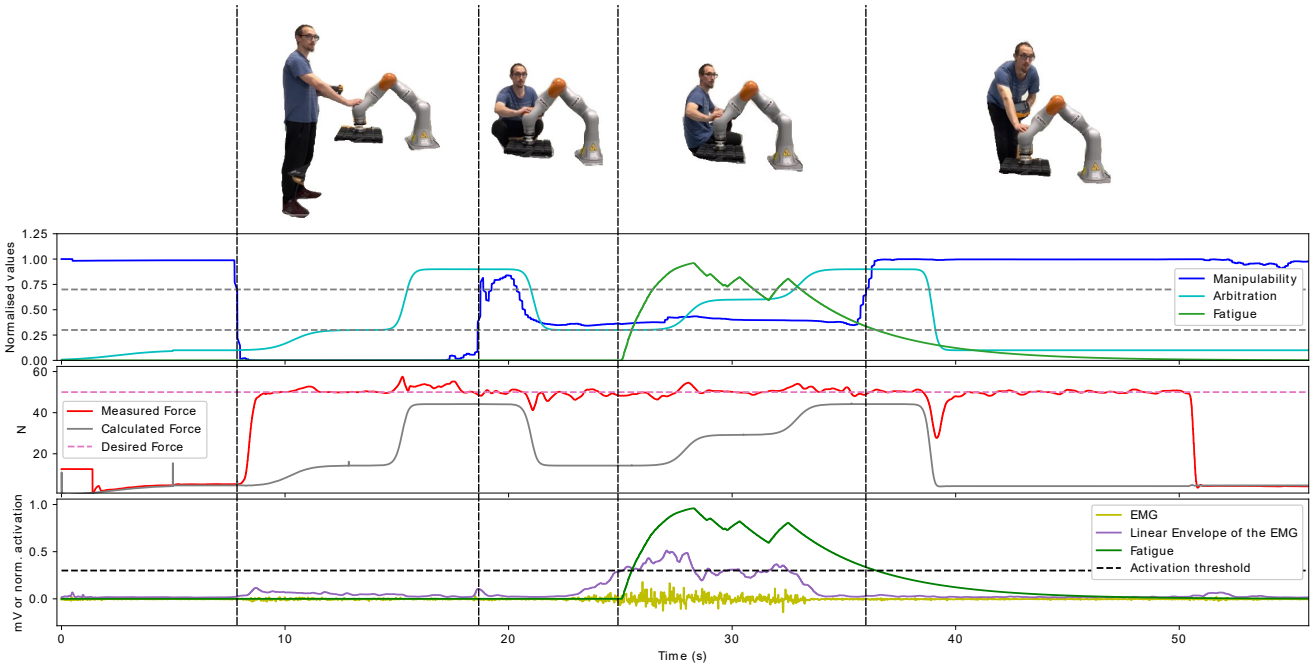

Fig. 10. Results plot for the multi-metric experiment.(0,0) Pictures of the poses that compose the experiment: Low manipulability pose, middle manipulability pose 1, middle manipulability pose 2 and high manipulability pose.(1,0) Fatigue value f, manipulability value m and arbitration value a vs time (s).(2,0) Calculated robot force and measured exerted force vs time (s). (3,0) Raw EMG signal, normalised EMG's linear envelope and fatigue f vs time(s).

every direction and oscillates around m = 0.5. In the *sideways medium manipulability pose*, the user keeps the squat but faces to his/her left. This way, the manipulability level keeps invariable, but the latissimus dorsi exerts the effort.

Finally, the *high manipulability pose* occurs with the user standing on the chair by the robot, leaning forward and pushing downward with the straight arm. As explained above, for the low manipulability position, in this case, we are using a position with a significant difference of principal axes. However, the force direction is much closer to that of the longest axis. Thus the manipulability level is close to 1.

Figure 10 shows the results for one of the loops of one of the repetitions of the *multi-metric experiment*. In a similar fashion to the fatigue experiment results, the first row displays the fatigue value f, the manipulability value m, and the arbitration value a; the second row displays the calculated robot force and the measured combined force; and the third row shows the raw EMG signal, the normalized EMG's linear envelope, and the fatigue value f.

E. High manipulability multi-metric experiment

The last of the experiments had the purpose of checking one of the affirmations set as motivation for this research: *even while keeping a good posture, exerting a force induces fatigue*. Even when performing a task in a high manipulability setting, the involved muscles get fatigued. Thus, allowing the robot to consider fatigue as an arbitration metric together with manipulability may prove beneficial and compensate for the faults of the corresponding separate modules on their own, by faults meaning, the inability of manipulability as a metric to keep historical track of the physical activity; And the inability of fatigue to instantly reflect or predict the effort involved or to be involved in the execution of a force or movement.

With the above-explained reasoning in mind, we set an experiment consisting of performing the proof of concept task: *collaboratively exerting a constant force against a horizontal surface*; but constantly keeping a high manipulability pose: the high manipulability position described in the multi-metric experiment, that we can see in the last picture of the first row of figure 10. We considered the triceps muscle for fatigue calculation, and the user performed a short calibration experiment previous to the task. The resulting $MVC = 0.17$ mV, and the resulting T_{end} = 94.6 seconds.

The experiment took 8 repetitions of 3 minutes each, 4 of them with R = 0.5 and 4 with R = 0.1. In this case, we selected R = 0.5 since it allowed us to see more transitions in the system. We can see in figure 11 the results of one of the repetitions. In a similar fashion as the previous test, it displays arbitration, manipulability, and fatigue in the first row; the calculated robot's end-point force and the measured combined force in the second row; and the raw EMG signal, the normalized EMG's linear envelope and fatigue in raw 3.

IV. Discussion

We developed a method to arbitrate the force contributed by each agent (a human and a robot) in a collaborative task. The proposed method finally resulted in an arbitration algorithm that maps the "ergonomic level" of the human arm (the human arm force manipulability and muscle fatigue) into predefined levels of force assistance (i.e., the robot exerts predefined percentages of the desired total required task force). Here,

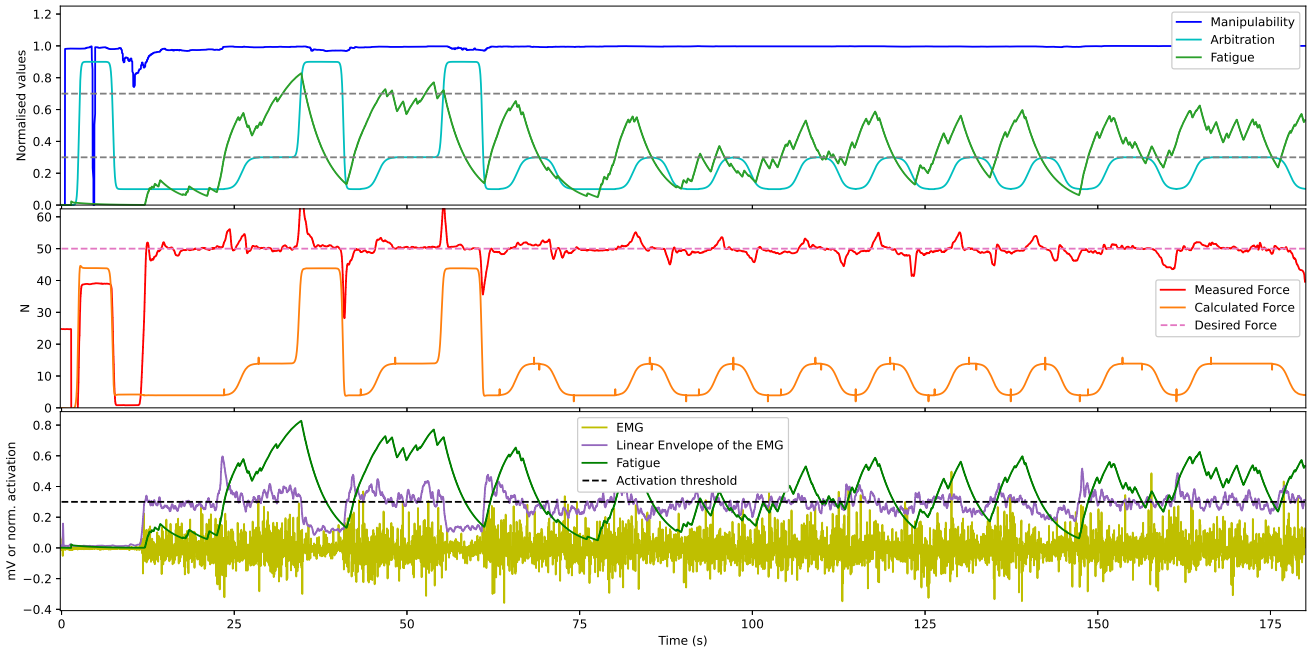

Fig. 11. Results plot for the high manipulability multi-metric experiment.(0,0) Fatigue value f, manipulability value m and arbitration value a vs time(s).(1,0) Calculated robot force and measured exerted force vs time (s). (2,0) Raw EMG signal, normalised EMG's linear envelope and fatigue f vs time(s).

we discuss the design choices of the proposed method. Then, designed functionalities: *modularity*, *fatigue hysteresis* and *conservative behavior*. Then, the limitations of the proposed method are explained, and finally, a future work subsection states the future modifications and research projects that should sharpen the proposed method.

Collaborative robots have the ultimate goal of decreasing workload to prevent injuries in the workplace and increase task performance. Nevertheless, decreasing the cognitive load below the individual skill of the user could lead to boredom [22]. Boredom is highly related to a detriment in vigilance [23], which in an industrial setting can be problematic in terms of safety. We, therefore, want to decrease the physical workload keeping user engagement. Most of the methods in the literature (all but [18] and [19]) propose authority arbitration methods between the human user and the automated system. Even though they increase the cognitive workload compared to a fully-automated system, the automated agents are task-dependent, limiting their applicability.

The methods in [24, 25] solves this issue by using machine learning techniques to teach the robot the required task by demonstration. However, they still require to be trained prior to the task execution, and when the robot performs the task, it is fully autonomous, removing the cognitive load from the task. [11], on the other hand, developed their method for power augmentation with exoskeletons. They altogether remove the automation, reducing the physical load and leaving the cognitive load for the user. However, actuated exoskeletons might be inconvenient. Gull et al. [26] concluded that in most of the currently developed systems, the human and robot interaction had not been addressed satisfactorily, pointing towards further research to achieve good interaction in terms of robot compliance.

We have developed a system following a similar framework to that of power-augmentation systems. The system is designed such that the cognitive work is on the human. The assistance is provided as needed depending on the ergonomic state of the human arm. We have also developed the proposed method on a collaborative robot, so the human biomechanics does not have to be explicitly considered for the correct functioning of the robot.

A. *Reactive versus preventive assistance*

Assistance can be provided reactively or preventively. In reactive assistance, the robot helps the human when a particular incident occurs, even when the incident is unforeseen or unmodelled. Nevertheless, reactive assistance can often be too late. e.g., if the considered incidence is an injury, preventing the incidence would be preferable. Thus, assistance should be preventive. The robot helps the human to avoid a specific incidence. However, the incidence is not explicitly considered for the assistance to be provided. Because the robot assists the human before the incidence takes place, it is based on the assumptions of the incidence happening under certain conditions and the assistive action preventing the incident.

If we consider muscle fatigue as the incident, reactive assistance can get to be insufficient. [13, 14, 15] concluded that a fatigued muscle is more likely to get injured. [12] discovered that arm muscles recruitment and activation is strongly related to human arm manipulability, and so is muscle fatigue induction. Thus, we decided to provide a combined approach (preventive and reactive) by using the manipulability metric to provide assistance that prevents fatigue and use

fatigue to provide assistance when fatigue cannot be prevented from happening.

Previous methods that exploit these two metrics individually and successfully achieve preventive or reactive behavior of the robot. The method in [18] employed fatigue for binary single-metric arbitration. The robot takes over the task when a certain level of muscle fatigue is reached. This way, the robot successfully provides rest when needed. The approach in [11], on the other hand, used manipulability for single-metric arbitration in a power-augmentation exoskeleton. This way, the exoskeleton provides assistance compensating for the difficulty in exerting force in every possible position. This way, they managed to reduce the muscle activation required for performing physical tasks. They reduced the induced fatigue reaching a fatigue-preventive behavior. However, method that can combine both reactive and preventive techniques was missing.

We filled this gap by successfully developing a multi-metric arbitration method. Our method can assists dependent on manipulability, aiming to reduce the probability of getting fatigued and dependent on muscle fatigue to react to fatigue when this cannot be prevented. We can see in figure 11 a scenario in which manipulability is high. Here the preventive assistance is not provided since manipulability is in a good state. Thus, fatigue slowly builds up, and the robot provides assistance reacting to the fatigue level. Interestingly, we can see that the fatigue level only reaches the maximum state in two loops. Afterward, it only gets to the intermediate state in every loop. This finding provides a good sign of the utility of combining preventive and reactive behavior. Nevertheless, further research should be conducted.

B. Continuous versus binary arbitration

Arbitration methods map the selected arbitration metrics' values into an arbitration value. This arbitration level is the value that represents the percentage of the desired force the robot is exerting. In the literature, we have found out there are two main approaches to take: the binary mapping [18, 24, 27], and the continuous mapping[25, 28]. The main advantage of binary mapping is its simplicity. Nevertheless, the changes between states are too significant for the user to adapt to them quickly. In the case of [24] they performed a collaboratively polishing task in which the robot and the human had to exert a constant force against the surface. They successfully performed the task since their arbitration method did not aim to improve performance but to allow for the human to supervise a robot learning process. Nevertheless, their results show difficulty for the human to compensate for the sudden changes in robot behavior. [29] already tried to overcome this issue by implementing a finite state machine that smoothly transitions between robot control and human control. They state an improvement in this aspect. Nevertheless, they concluded that there is a trade-off between accuracy and workload reduction with such a framework.

On the other hand, the continuous mapping translates the metrics state more accurately into the control command. The robot can adapt without the need to have predefined states. This way, continuous mapping aims to have a more natural robot behavior and act highly sensitive to the selected metrics. The work in [25] successfully implemented a continuous arbitration method for control transfer during a machine learning process in which a human teaches a robot standing balance. Here the method results are satisfactory since the human does not directly feel the arbitration. The robot listens to the human commands to more or less grade depending on the arbitration value. [28] successfully implemented a continuous arbitration method for a pick-and-place task in telemanipulation. Similarly, the human does not directly feel the arbitration since the robot arm and the human are not in direct contact. In our case, the arbitration value is translated directly into the interaction forces between the robot and the human. Thus, the human needs to adapt to the robot forces and the binary method produces significant force changes that humans can find problematic. In the case of continuous metrics, having a continuously varying force involves a big increase in the task's cognitive load.

Considering human adaptability and cognitive load, we decided to exploit the advantages of both by finding a middle-ground: the simplicity and cognitive easiness of the binary methods and the sensitivity of the continuous methods. We used a finite state model with 4 states for multi-metric arbitration and three states for single-metric arbitration. We can see a representation of the behavior of the arbitration method for every combination of the metrics in figure 6. Looking in the second graphs at figures 10 and 11 we can see that the proposed method successfully performed the task: *collaboratively exerting a constant force against a horizontal surface.*

C. Modularity

Using multi-metric arbitration in the industry is not always possible. In some scenarios, the visibility of the user from the motion capture system might be compromised. In other cases, the user might want to use the system for a short task like moving a heavy load and then move on to a task that does not require physical assistance. Then, the user will prefer not to set the EMG sensors up. Thus, we considered it necessary to make the algorithm work for single-metric arbitration and multi-metric arbitration. As we have already seen, the literature only contains methods for single-metric arbitration with ergonomic metrics [18, 19, 11]. Nevertheless, there are a few multi-metric arbitration methods with other metrics as well. The method in [28] use confidence in automation and confidence in the human user as arbitration metrics for telemanipulation. Even though their arbitration method is suitable for working with either metric separately, they did not contemplate such a scenario due to the different requirements of their application.

To address the above issue, we have developed a modular arbitration method. The proposed method can be used with either metric for single-metric arbitration or both metrics for multi-metric arbitration. We can see in figure 6 proof for this. In both columns, the three first graphs correspond to the mapped arbitration value from different multi-metric combinations. The last graph in both columns represents the

arbitration value for single-metric arbitration. An example of modularity working is shown in figures 8 and 9. We also aimed to provide preventive-reactive assistance when only fatigue is available as a metric, as we can see in figure 9 by adding an intermediate assistance state when fatigue is moderate for when manipulability cannot be used.

D. Fatigue hysteresis

We face two main problems with the fatigue module: The fatigue model presents two modes, fatigue, and recovery. When the activation level of the muscle oscillates around the predefined threshold, the transition between these two models is too sharp and can lead to fast state transitions. Also, when high fatigue is reached, the performance is already compromised, and the risk of injury is high. In the literature, the fatigue models that have been used do not consider the fatigue-recovery sharp transitions. [18, 15, 17]. The work in [18] developed a binary method with a single fatigue threshold that determines if the model is in human-control or robot-control.

To avoid sharp and rapid switching between fatigue and recovery modes, we have designed the proposed method to have *fatigue hysteresis*. The fatigue single-metric arbitration module is composed of three states and two fatigue thresholds. When fatigue reaches a high level, the intermediate state and high fatigue thresholds are disregarded, so the system provides extra resting time that the user can use to change to a more comfortable position or resting. Moreover, when the system is performing multi-metric arbitration, manipulability is also disregarded in a high fatigue state. The robot provides maximum assistance until the muscle is rested. We can see in figure 6 proof of this feature. In the left columns, when the fatigue value is decreasing, the system provides maximum assistance level for an extended period in every case, disregarding manipulability. We can see a similar result in figure 10.In the third posture, high fatigue level is reached. Then, when the user changes to a high manipulability pose in the fourth posture, the assistance level is maximum until the muscle is rested, providing additional 5 seconds of rest that are enough for the user to choose a better posture.

E. Conservative behavior

Even though multi-metric arbitration seems beneficial, it can also be problematic. When the muscle is fatigued, having the system transition to different states depending on manipulability would add cognitive load without achieving real benefits. An increase in the required load means an increase in muscle activation. Similarly, the muscle activity required to increase force is inversely proportional to the manipulability level. When manipulability is very low, the possibility of the system adapting to fatigue and transitioning state would be detrimental rather than beneficial. In the literature, this issue has not been directly analyzed. However, [24] proposed a hybrid mapping framework of ergonomic metrics for communicating ergonomic work conditions. They combined a binary map and a continuous map, filtered out the configurations in which at least one of the ergonomic metrics was not sufficiently good. For the configurations in which all the ergonomic metrics were sufficiently good, they proposed continuous mapping for tasks that are not highly paced and changeable, such as ours.

Aiming to avoid the human from having to adapt when either the posture is not good enough, or the fatigue level is too high, We have designed a functionality called *conservative behavior*. This system provides maximum assistance either if the manipulability is not sufficiently high or the fatigue level sufficiently low, in this case exploiting *fatigue hysteresis* to provide extra rest. Nevertheless, when both metrics are sufficiently good, the proposed method provides three assistance levels as a middle point between the binary framework and continuous frame that [6] recommend in their hybrid mapping. The reasoning behind this is in the fourth paragraph of this discussion. We can see proof of *Conservative behavior* in the positions (0, 0) and (2, 1) of figure 6 and figure 10.

F. Limitations and future work

For the sake of simplicity, the proposed method presents some limitations. It is designed to work in a hard-coded predefined direction. In the case of the proof of concept task, it is vertical or z-direction. Also, we have only considered one muscle for muscle fatigue estimation. Moreover, the proposed method is suitable for lengthy steady tasks. In order to make it robust to the noise introduced by the Kinect V2 sensor, we have set the transition time between states too long for highly paced tasks. Nevertheless, the proposed method sets a good start towards further exploiting multi-metric arbitration with ergonomy metrics.

As future work, we propose several ideas:

1) More muscles should be considered for the fatigue estimate.
2) It would be interesting to exploit the IMU functionality of the Delsys sensors. Even though the Kinect V2 sensor is easy to implement and use, it would be interesting to evaluate a more accurate motion capture system in the scenarios where the users set up the EMG sensors.
3) The proposed method should be tested on actual collaborative tasks: drilling, polishing, and co-manipulating. An interface to easily select the direction of the applied force and the desired force should be developed, integrating different tasks.
4) Human-factors research to evaluate the effect of the proposed method in performance versus state-of-the-art methods should be conducted.

V. Conclusions

We have developed a satisfactory multi-metric arbitration method applying the assist-as-needed approach for a physical human-robot collaboration scenario. It is suitable for tasks where end-point assistance force is mainly required in one direction, and quick changes of the human arm position are not required: polishing, drilling, or co-manipulating. Nevertheless, more work is worth doing towards the generalization of the proposed method. More importantly, human-factors research is required to quantitatively know the effect on the performance of the proposed method versus state-of-the-art methods.

REFERENCES

[1] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," 2018.

[2] S. Haddadin and E. Croft, "Physical human-robot interaction," in *Springer Handbook of Robotics*. Springer International Publishing, jan 2016, pp. 1835–1874. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-32552-1_69

[3] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar, "Collaborative human-humanoid carrying using vision and haptic sensing," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., sep 2014, pp. 607–612.

[4] N. Jarrassé, V. Sanguineti, and E. Burdet, "Slaves no longer: Review on role assignment for human-robot joint motor action," *Adaptive Behavior*, vol. 22, no. 1, pp. 70–82, feb 2014.

[5] A. U. Pehlivan, D. P. Losey, and M. K. Omalley, "Minimal Assist-as-Needed Controller for Upper Limb Robotic Rehabilitation," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 113–124, feb 2016.

[6] L. Peternel, D. T. Schøn, and C. Fang, "Binary and Hybrid Work-Condition Maps for Interactive Exploration of Ergonomic Human Arm Postures," *Frontiers in Neurorobotics*, vol. 14, p. 114, jan 2021.

[7] T. Yoshikawa, "Manipulability of Robotic Mechanisms."

[8] J. Jacquier-Bret, P. Gorce, and N. Rezzoug, "The manipulability: a new index for quantifying movement capacities of upper extremity," *http://dx.doi.org/10.1080/00140139.2011.633176*, vol. 55, no. 1, pp. 69–77, jan 2011.

[9] S. Gopinathan, S. K. Ötting, and J. J. Steil, "A user study on personalized stiffness control and task specificity in physical human-robot interaction," *Frontiers Robotics AI*, vol. 4, no. NOV, p. 58, nov 2017.

[10] L. Peternel, W. Kim, J. Babic, and A. Ajoudani, "Towards ergonomie control of human-robot co-manipulation and handover," *IEEE-RAS International Conference on Humanoid Robots*, pp. 55–60, dec 2017.

[11] T. Petrič, L. Peternel, J. Morimoto, and J. Babič, "Assistive arm-exoskeleton control based on human muscular manipulability," *Frontiers in Neurorobotics*, vol. 13, p. 30, 2019.

[12] P. K. Artemiadis, P. T. Katsiaris, M. V. Liarokapis, and K. J. Kyriakopoulos, "On the effect of human arm manipulability in 3D force tasks: Towards force-controlled exoskeletons," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 3784–3789.

[13] S. D. Mair, A. V. Seaber, R. R. Glisson, and W. E. Garrett, "The role of fatigue in susceptibility to acute muscle strain injury," *American Journal of Sports Medicine*, vol. 24, no. 2, pp. 137–143, apr 1996.

[14] L. Ma, D. Chablat, F. Bennis, and W. Zhang, "A new simple dynamic muscle fatigue model and its validation," *International Journal of Industrial Ergonomics*, vol. 39, no. 1, pp. 211–220, jan 2009.

[15] R. Ma, D. Chablat, F. Bennis, and L. Ma, "Human Muscle Fatigue Model in Dynamic Motions," *Latest Advances in Robot Kinematics*, pp. 349–356, jan 2012. [Online]. Available: https://link.springer.com/chapter/10.1007/978-94-007-4620-6_44

[16] T. Xia and L. A. Frey Law, "A theoretical approach for modeling peripheral muscle fatigue and recovery," *Journal of Biomechanics*, vol. 41, no. 14, pp. 3046–3052, oct 2008.

[17] P. Maurice, V. Padois, Y. Measson, and P. Bidaud, "Experimental assessment of the quality of ergonomic indicators for dynamic systems computed using a digital human model," *International Journal of Human Factors Modelling and Simulation*, vol. 5, no. 3, pp. 190 – 209, oct 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01383415 https://hal.archives-ouvertes.fr/hal-01383415/document

[18] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani, "Robot adaptation to human physical fatigue in human–robot co-manipulation," *Autonomous Robots*, vol. 42, no. 5, pp. 1011–1021, jun 2018.

[19] L. Peternel, C. Fang, N. Tsagarakis, and A. Ajoudani, "A selective muscle fatigue management approach to ergonomic human-robot co-manipulation," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 69–79, aug 2019.

[20] P. Chiacchio, "New dynamic manipulability ellipsoid for redundant manipulators," *Robotica*, vol. 18, no. 4, pp. 381–387, jul 2000. [Online]. Available: https://www.cambridge.org/core/journals/robotica/article/new-dynamic-manipulability-ellipsoid-for-redundant-manipulators/BBCA0F3168D7E1A48A66273BAEC9C8E0

[21] X. Ding and C. Fang, "A novel method of motion planning for an anthropomorphic arm based on movement primitives," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 624–636, 2013.

[22] M. L. Cummings, F. Gao, and K. M. Thornburg, "Boredom in the Workplace: A New Look at an Old Problem," *Human Factors*, vol. 58, no. 2, pp. 279–300, mar 2016.

[23] J. D. Eastwood, A. Frischen, M. J. Fenske, and D. Smilek, "The Unengaged Mind: Defining Boredom in Terms of Attention," *Perspectives on Psychological Science*, vol. 7, no. 5, pp. 482–495, sep 2012. [Online]. Available: /record/2012-24579-005

[24] L. Peternel, E. Oztop, and J. Babič, "A shared control method for online human-in-the-loop robot learning based on Locally Weighted Regression," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem. Institute of Electrical and Electronics Engineers Inc., nov 2016, pp. 3900–3906.

[25] L. Peternel and J. Babic, "Humanoid robot posture-control learning in real-time based on human sensorimotor learning ability," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp.

5329–5334.

[26] M. A. Gull, S. Bai, and T. Bak, "A review on design of upper limb exoskeletons," *Robotics*, vol. 9, no. 1, p. 16, mar 2020. [Online]. Available: https://www.mdpi.com/2218-6581/9/1/16/htm https://www.mdpi.com/2218-6581/9/1/16

[27] A. Kucukyilmaz, T. M. Sezgin, and C. Basdogan, "Intention recognition for dynamic role exchange in haptic collaboration," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 58–68, 2013.

[28] S. Li, M. Bowman, and X. Zhang, "A General Arbitration Model for Robust Human-Robot Shared Control with Multi-Source Uncertainty Modeling," *arXiv*, mar 2020. [Online]. Available: http://arxiv.org/abs/2003.05097

[29] S. O. Oguz, A. Kucukyilmaz, T. M. Sezgin, and C. Basdogan, "Haptic negotiation and role exchange for collaboration in virtual environments," in *2010 IEEE Haptics Symposium, HAPTICS 2010*, 2010, pp. 371–378.

Appendix A: Extended figures

This appendix contains the extended version of the figures in the manipulability experiment (Fig 1), the fatigue experiment (Fig 2), and the multi-metric experiment(Fig 3). More repetitions of these experiments were performed, but we considered the ones displayed here representative enough. Notice that the figure and equation numbering is reset for the appendices to differentiate this section from the research paper.

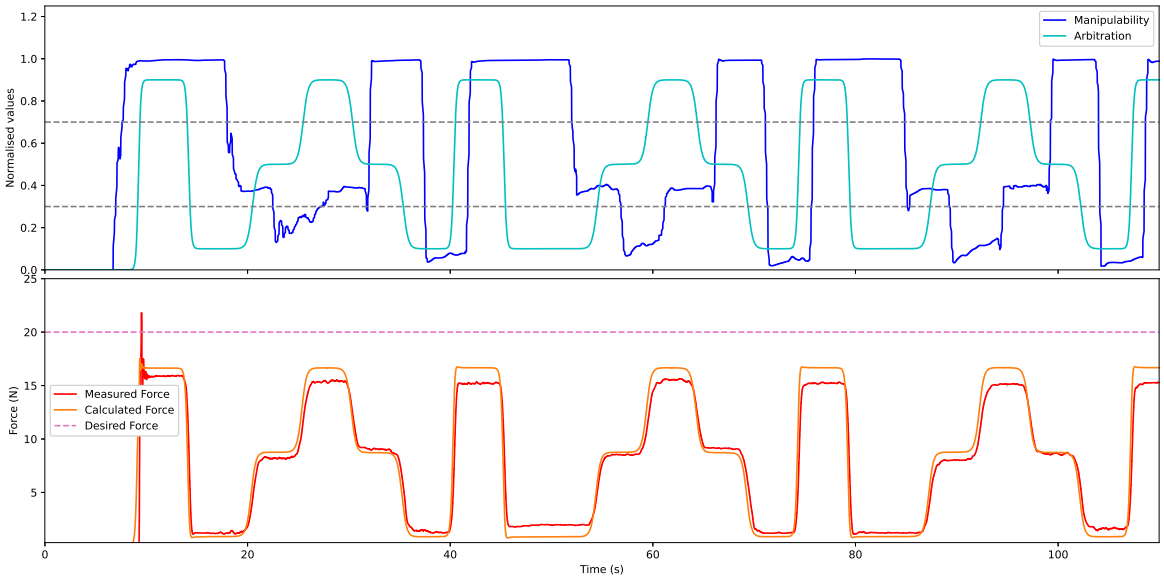

Figure 1: Long results plot for the manipulability experiment together with images of the poses that were performed in each moment of the test. (0,0) Manipulability m, fatigue f, and arbitration a vs time (s). (1,0) Calculated robot force and Measured force vs time (s).

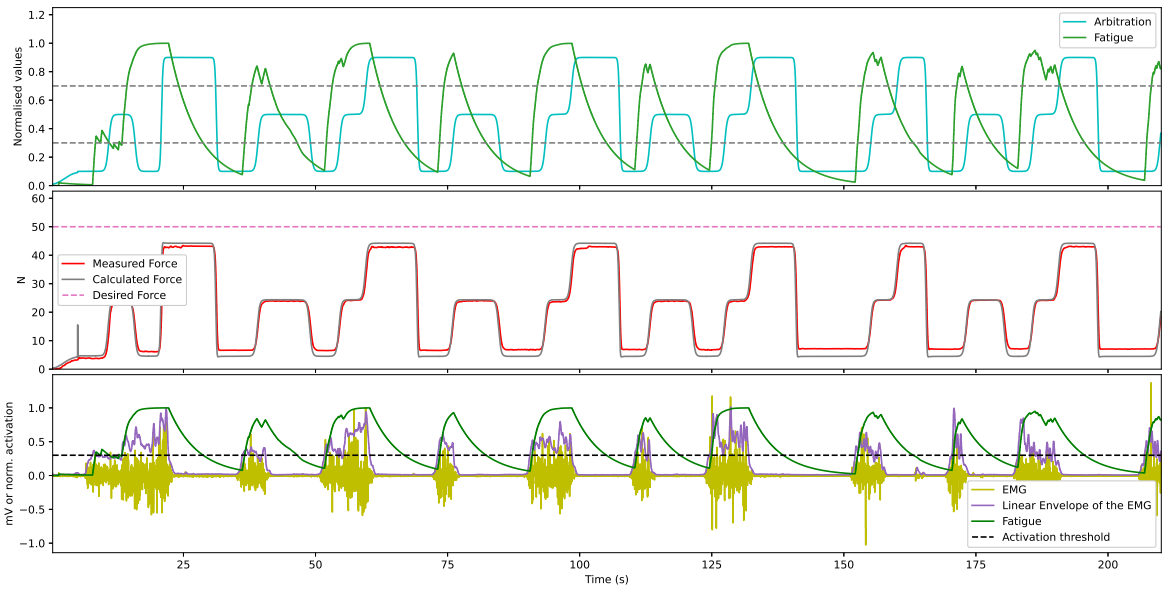

Figure 2: Long results plot for the fatigue experiment. (0,0) Fatigue value f and arbitration value a vs time(s).(1,0) Calculated robot force and measured exerted force vs time (s). (2,0) Raw EMG signal, normalised EMG's linear envelope and fatigue f vs time(s).

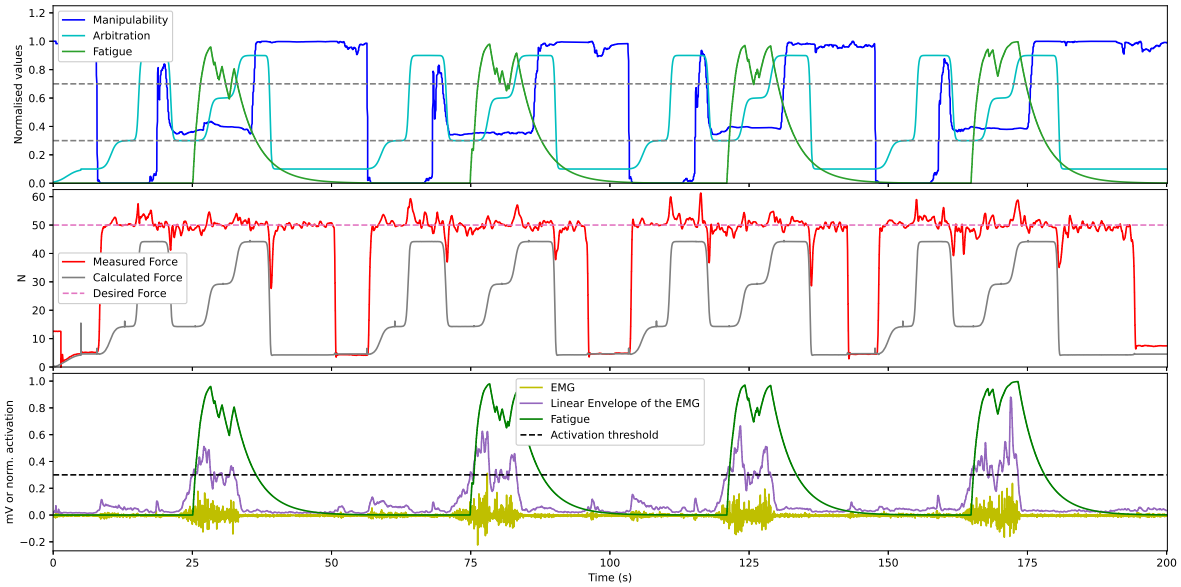

Figure 3: Long results plot for the high manipulability multi-metric experiment.(0,0) Fatigue value f, manipulability value m and arbitration value a vs time(s).(1,0) Calculated robot force and measured exerted force vs time (s). (2,0) Raw EMG signal, normalised EMG's linear envelope and fatigue f vs time(s).

Appendix B: Manipulability module

This appendix presents in-depth information about the motion capture method and data processing for manipulability estimation. As mentioned in the *method* section, we use Kinect V2 for this purpose. Implementing the Kinect SDK C++ library, we extracted right away the estimates the Kinect V2 does on the right wrist, elbow, and shoulder positions. Then, we process the data, change the reference frames to fit the Robot's reference frame, and correctly interpret manipulability information. Once this is achieved, the calculations to transform task-space data into joint-space data take place to estimate the manipulability ellipsoid finally and extract the direction of interest.

.1. Position data capture and processing

The Kinect SDK library allows access to the orientation and position data of the joints in the-task space. Nevertheless, these data are noisy and represented concerning the reference frame of Kinect V2. A moving window of 300 manipulability values is averaged and updated for the former issue in every iteration.

For the latter issue, we modified the reference frame of the shoulder to be coincident with that of Kuka for a fixed positioning of the user. Nevertheless, since the task was being performed exerting force only in the *z-direction*, this is not particularly relevant in this case, so the Kinect did not have to be perfectly aligned in *x* and *y* and its location could be varied between experiments. Notice that if the task-required applied force involves any other direction (not one of the reference axis), the location and orientation of the Kinect need to be defined and kept throughout the use of the proposed method. Then, the rest of the local frames were expressed concerning the new shoulder reference frame.

$$\mathbf{T} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & (dis - s)\cdot\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & (dis + s)\cdot\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 & -h \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

In Eq 1 we can see the transformation matrix that was applied to make the aforementioned reference frame transformation. Here, dis, s and h represent the $y - coordinate$, $x - coordinate$ and $z - coordinate$ respectively in the Kinect reference frame of the shoulder position, which is extracted online.

Appendix C: Fatigue

For the fatigue module, the Delsys Trigno system was used. We set the EMG sensors' frequency to 1500Hz. This way, by sending to the robot one every 50 samples, we matched the Kinect frequency of 30Hz for the arbitration. Thes raw EMG data were processed before applying the fatigue model. Moreover, we designed a fast, easy-to-implement Jupyter notebook to calibrate the model to the used muscle. Finally, an online visualization system was implemented for the user to know his/her fatigue level and, this way, judge if the model's behavior meets the user's qualitative sensations.

.2. Signal collection and processing

A python library called pytrigno was used to define the settings and access to the EMG channels of the sensors. Then, the signal was processed through python before calculating the fatigue value. First, the signal is rectified through the application of a 2^{nd} order high pass Butterworth filter with cut-off frequency at 20Hz. The absolute value of the resultant signal was computed, and then a second-order low-pass Butterworth filter with cut-off frequency at 2 Hz was applied. The signal was then normalised against the Maximum Voluntary contraction of the selected muscle.

.3. Calibration experiment

The fatigue model needs to be calibrated to the user and selected muscles. For this purpose, we designed a brief and easy-to-implement calibration exercise in a Jupyter notebook. First, the maximum voluntary contraction (MVC) was calculated. For this, the user is asked to exert maximum force in a body posture that maximally engages the selected muscle. For example, the user presses upwards with the extended arm against a mechanical resistance for the anterior deltoid. An example of this is represented in Fig 4. The user is asked to perform this exercise 3 times, exerting maximal force for 3 seconds each. These 3-second signals are processed as described in the previous section. Then the maximum values are selected and averaged between the three signals to extract the MVC value.

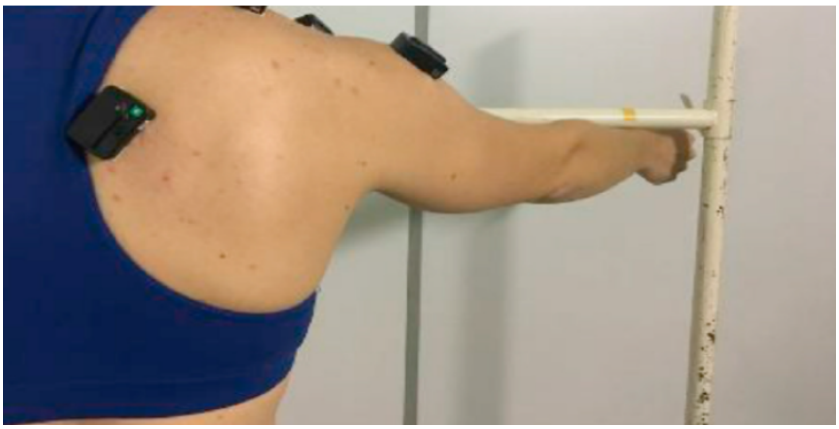

Figure 4: Representation of the MVC calibration exercise for for the fatigue model to use with the anterior deltoid muscle.

Once we know the MVC value, the user performs a second exercise to get the T_{end} or endurance time, also required for the fatigue model. This second exercise consists of exerting 0.3 times the MVC for as long as the effort can be stably exerted. We, therefore, programed an online visualization plot using pyQtgraph that shows the target value and a region in which the user needs to keep the maximum value of the linear envelope of their EMG signal. When the linear envelope goes below the low or high threshold, the system saves the time the user endures. The exercise can be performed several times for practice. Then three 'official' takes are finally recorded and averaged. The thresholds were finaly set to +0.1 and -0.1 after several trials. We can see an screenshot of the visualization plot in Fig 5

.

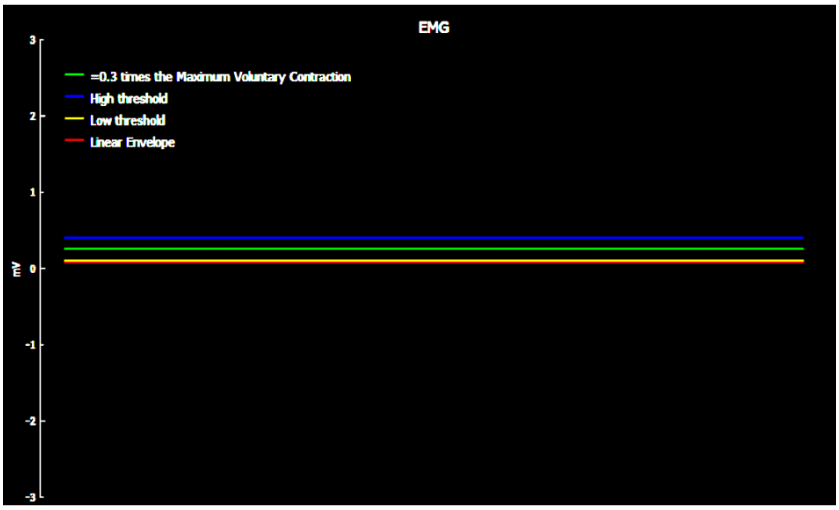

Figure 5: Screenshot of the online visualization plot of the T_{end} calibration exercise for for the fatigue model. (Green) 0.3 times the MVC. (Blue) High threshold or limit of the target zone of the experiment. (Yellow) Low threshold or limit of the target zone of the experiment. (Red) Linear Envelop of the EMG signal measured from the user.

.4. Online fatigue feedback

We programmed an online visual feedback plot to online evaluate the fatigue model. For this, we also used pyQtgraph. We can see in Fig 6 a screenshot of this visualization system.

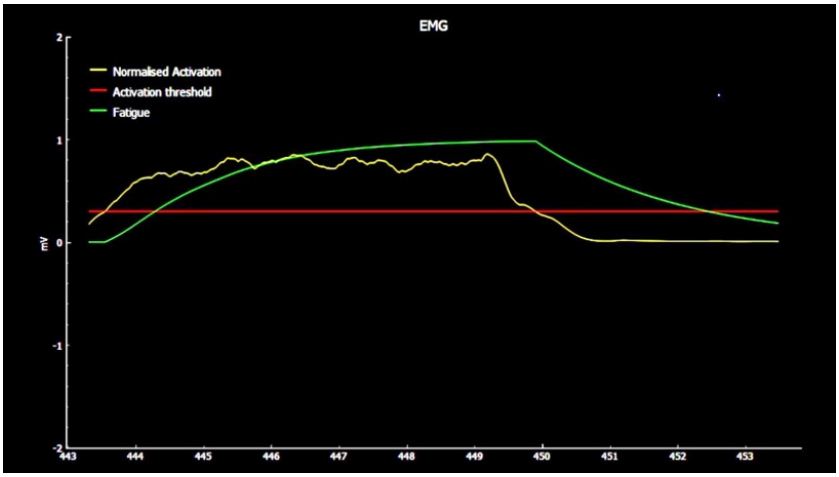

Figure 6: Screenshot of the online feedback plot of the fatigue model. (Yellow) Normalised linear envelope of the EMG signal called here normalised activation measured from the user. (Red) Activation threshold for the fatigue model. (Green) Calculated fatigue value.

Appendix D: Force Feedback

To successfully perform the task in the proposed method, a force feedback is needed showing the target force or desired force and plotting the force that is being collaboratively exerted. The exerted force was measured online through a Schunk FTS-Delta SI-330-30 load cell. The force feedback was also programmed using the pyQtgraph library. We can see a screenshot of it in Fig 7.

Figure 7: Screenshot of the online force feedback visualization during the multi-metric experiment. (Red) Desired force set at 50N. (Yellow) Collaborative measured force.