UNIVERSITY OF GENOVA

MASTER OF SCIENCE IN ROBOTICS ENGINEERING

A Method for Dynamic Human-Robot Role Allocation based on Ergonomic Risk Prediction

by

Elena Merlo

December 2021

Supervisors:

Prof. Fulvio Mastrogiovanni — *University of Genova (Italy)*

Dr. Arash Ajoudani — *Italian Institute of Technology (Italy)*

Co-supervisors:

Dr. Edoardo Lamon — *Italian Institute of Technology (Italy)*

Dr. Marta Lorenzini — *Italian Institute of Technology (Italy)*

Dr. Alessandro Carfì — *University of Genova (Italy)*

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

A Nonna Rita, la mia luce.
Mi manchi.

Ringraziamenti

Un sincero grazie al Professor Mastrogiovanni e ad Alessandro per avermi aiutato ad orientare la mia ricerca verso il tema sviluppato in questa tesi.
Grazie ad Arash, Edo e Marta per avermi accompagnato in questi mesi, per i suggerimenti, le critiche, le opportunità e per avermi fatto crescere attraverso la loro esperienza.
Grazie a tutto l'HRII lab per avermi accolto fin da subito e avermi fatto sentire parte del gruppo per tutto il tempo. Conclusa questa esperienza ho nuovi amici. Grazie in particolare a Marta, Albi, Emir, Francesco e Soheil, con cui ho vissuto quattro intense giornate a Roma in occasione del Maker-Faire.
Inoltre, vorrei ringraziare tutta la mia famiglia e i miei amici: mi sento estremamente amata e fortunata. Ringrazio, in particolare, mamma e papà, perchè mi hanno sempre incoraggiato e sostenuto nei momenti di buio, perchè credono nelle mie capacità molto di più di quanto riesca a farlo io. Grazie perchè mi hanno insegnato che i traguardi si raggiungono un passo alla volta, con fatica e con il sorriso, sempre.
Grazie a Bambo e Nenno, più fratelli che cugini: con voi sento sempre il profumo di casa.
Grazie a Fabi e Ari, le mie amiche di sempre. Grazie per condividere con me le risate fino al mal di pancia, i discorsi seri e impostati, i pianti, le insicurezze, i progetti, le paure. Quello che abbiamo costruito in questi anni è davvero potente e sono sicura sarà per sempre.
Grazie a Luca (LT), il mio migliore amico. Il "classico esempio" di persona con cui si può parlare di tutto, un'amicizia speciale che resiste ai km di distanza. Grazie perchè questi anni di Magistrale non sarebbero stati gli stessi senza il tuo aiuto e la tua compagnia.
Grazie ad Andrea, perchè mi vuole bene per come sono e non desidera cambiarmi. Stare con te mi rende felice.
Grazie a Chiara e Daniel, che aprono le porte di casa e del cuore ogni volta che qualcuno ne ha bisogno. Grazie a Chiara, Gaia, Marta e Silvia, perchè quando sono con loro mi sento sempre al posto giusto.
Grazie a Lalli, un concentrato di energia, creatività e speranza che mi ispira fin da bambina.
Grazie a Michi e Silvia "per tutta la luce".

Grazie a Sofia e a tutti i ragazzi con cui ho condiviso innumerevoli sabati pomeriggio e domeniche sera, perchè mi hanno insegnato l'importanza del prendersi cura e la responsabilità di essere guida.
Infine, anche se mi risulta difficile, vorrei ringraziare me stessa: col sorriso, ad maiora!

Abstract

Human Robot Collaboration (HRC) applications can bring several benefits when applied in industrial environments, since they allow to exploit the skills of both humans and robots involved in the same team. Collaborative robots (cobots) are convenient especially for small-sized enterprises, characterized by flexible and agile manufacturing requirements: compared to the highly specialized platforms, cobots are fast re-configurable and adaptable to the new product demand. A recent trend in collaborative robotics research consists in integrating ergonomic principles in the design of the HRC strategy to mitigate both physical and cognitive risks for the human operators, which would affect the entire process in terms of time and costs. The aim of this thesis is to investigate a new method for an ergonomic and fruitful HRC. In particular, the idea is to exploit task features and some quantitative measurements, that are computed online, encoding the (physical or cognitive) human status, to optimize the collaboration at the team-level. By acting on the roles distribution among the workers within the human-robot team, we aim to allocate to the human workers only those tasks that do not pose an ergonomic risk for them. The developed framework is composed by two main modules. The first provides the model used to characterise the HRC task, considering actions and workers features. We exploited AND/OR Graph (AOG)s, that we adapted to solve also the role allocation problem. The other module provides the human ergonomic risk evaluation during the actions performance, updating a risk indicator that contains the history of the human "weariness" from the beginning of the collaboration. The ergonomics information are embedded in the AOG arcs costs, that change during the task performance, guiding an optimal graph search to assign to human workers only actions that are not hazardous. The proposed framework is not paired with any specific metrics, but it remains a valid approach by using different risk indicators, chosen according to the task features: it could convey information about the physical human status or about the cognitive one. To validate our framework we arranged a proof-of-concept assembly, made of light-weight pieces, and we designed a kinematic risk indicator, *kinematic wear*, to evaluate potentialities and limitations of the proposed method. After testing the two modules separately by means of two different pilot experiments, we conducted a multi-subject preliminary experiment to evaluate the

overall framework. By using the validated NASA questionnaire we were able to compare the HRC task load with the effort required for executing the same task without the robot aid. Then, with a custom questionnaire we tried to investigate the efficiency of the ergonomic role allocation during the collaborative performance, trying to evaluate also the usability of such a framework in industrial contexts. Even though the number of involved subjects is limited, the results look promising: the more risky actions are recognized and they are not allocated to the human. This leads subjects to complete the HRC task less fatigued and frustrated.

Table of contents

List of figures viii

List of tables xiii

1 Introduction 1
- 1.1 Motivations . . . 1
- 1.2 State of the Art . . . 3
  - 1.2.1 Multi-Agent Role Allocation . . . 3
  - 1.2.2 Ergonomics in Human-Robot Collaboration . . . 9
- 1.3 Contribution . . . 12
- 1.4 Framework . . . 13

2 AND/OR Graphs for Role and Task Planning 15
- 2.1 AOG for Industrial Task Planning . . . 15
- 2.2 AOG for Role Allocation . . . 19
- 2.3 AO* for the Optimal Search . . . 21

3 Integrating Ergonomics in Human-Robot Role Allocation 25
- 3.1 Risk Prediction for hyper-arcs cost updating . . . 25
- 3.2 Ergonomic Risk Indexes . . . 29
- 3.3 Kinematic Wear . . . 33

4 Software Architecture and Performances 39
- 4.1 Software Architecture . . . 39
- 4.2 Computational Complexity Evaluation . . . 41

5 Experiments 44
- 5.1 Pilot Experiment: Dynamic Role Allocation testing . . . 46

5.2 Pilot Experiment: Kinematic Wear testing 51
5.2.1 Kinematic Wear-based Action Description 53
5.2.2 Overview of the results 76
5.3 Multi-subject Experiment 79

6 Conclusions **88**

Appendix A Questionnaires **90**

References **95**

List of figures

1.1 Human and Robot skills according to Fitts (1951). Original illustrations. . . 2
1.2 Visual representation of the three axes of Gerkey and Matarić's taxonomy (Korsah et al. (2013)) 4
1.3 Engagement levels in human-robot interaction (Bauer et al. (2016)) 6
1.4 Capability-aware Role Allocation Lamon et al. (2019) 8
1.5 (left) Lightweight robotic arm Kuka LWR 4+, (right) Lightweight robotic arm Franka Emika Panda equipped with its standard gripper. 9
1.6 Result of REBA calculation for the trunk, neck, upper arms and lower arms (Busch et al. (2018)) 10
1.7 Framework scheme. 13

2.1 AND/OR Graph for a simple product De Mello and Sanderson (1990). . . . 17
2.2 Disassembly plan using AOG De Mello and Sanderson (1990). 19
2.3 AOG according to Johannsmeier and Haddadin (2016). 20
2.4 General AOG showing the duplication of hyper-arcs for introducing in such structure the role allocation problem. Each of these additional arcs has a different cost according to the suitability of each agent in performing the corresponding action. 21

3.1 An arbitrary LTI system in discrete time, with an exogenous input u, a disturbance e, and an output y. The two input signals are filtered with transfer functions $G(z)$ and $H(z)$, respectively. 26
3.2 COP and vertical GRF variations without and with interaction forces. At the joint level $\Delta\tau$ describes the overloading torque (Kim et al. (2017b)) 31
3.3 RULA scores for shoulder postures in sagittal plane 35
3.4 RULA scores for postures in sagittal plane of elbow (top), wrist (second), trunk (third), neck (bottom) 36

3.5 Charge and discharge trend of wrist *kinematic wear* $V_{wrist}(t)$ if joint angles remain in the same risk area ($G = 1$, $G = 3$, $G = 5$, and $G = 7$). The higher the risk (greater G), the faster the charge and discharge time. 38

4.1 Software Architecture of the designed framework. The circles represent ROS nodes, while the squares are ROS topics. The arrows describe the data flow. 39
4.2 AO* computational time (in log scale) obtained by increasing the number of pieces to be assembled (top) and workers (bottom). 42

5.1 The four pieces used for the proof-of-concept assembly task. The three aluminum profiles (L, $S1$, $S2$) and the corner joint (J). 44
5.2 The complete assembly. 45
5.3 Experimental setup. The human worker and the robot co-worker shared the workbench where the assembly pieces were placed. A monitor informs the human worker of allocation results. 47
5.4 Action 1: pick (left) and place (right) the corner joint J 47
5.5 Action 2: pick (left) and assemble (right) the aluminum profile S_1 48
5.6 Action 3: pick (left) and assemble (right) the aluminum profile S_2 48
5.7 Action 4: pick (left) and assemble (right) the aluminum profile L 48
5.8 Action 5: pick (left) and place (right) the entire assembly 48
5.9 (Top) Snapshots of the experiment. (Bottom) *cumulRiskScoreH* value across the five actions (blue), the resulting values in case a damaging action would be assigned to the human (dashed red line), the robot hyper-arcs cost C_R (black) working as a threshold on *cumulRiskScoreH*. 50
5.10 Experimental setup. The workstation consisted in a workbench in front of the human operator and in a shelf at the human's right. The corner joint J was on top of the shelf, while the aluminum profiles were on the workbench. The human wore an Xsens suit as motion-capture system. The human operator assumed the minimum risk position (right), i.e. RULA score was about 1 for all the monitored joints. 52
5.11 X-sens human model. The red squares highlights the considered frames. . . 54
5.12 Action 1: pick and place J on the workbench 56
5.13 Action 1: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant. 57
5.14 Action 1: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant. 58

5.15 Action 1: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant. 58
5.16 Action 1: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant. 59
5.17 Action 1: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant. 60
5.19 Action 2: pick and assemble the aluminum profile L 60
5.18 The *kinematic wear* trend for all the monitored joints during the performance of a_1, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_1 (bottom). 61
5.20 Action 2: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant. 62
5.21 Action 2: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant. 62
5.22 Action 2: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant. 63
5.23 Action 2: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant. 63
5.24 Action 2: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant. 64
5.26 Action 3: pick and assemble the aluminum profile S_1 64
5.25 The *kinematic wear* trend for all the monitored joints during the performance of a_2, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_2 (bottom). 65
5.27 Action 3: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant. 66
5.28 Action 3: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant. 66
5.29 Action 3: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant. 67
5.30 Action 3: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant. 67
5.31 Action 3: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant. 68
5.33 Action 4: pick and assemble the aluminum profile S_2 68

5.32 The *kinematic wear* trend for all the monitored joints during the performance of a_3, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_3 (bottom). 69
5.34 Action 4: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant. 70
5.35 Action 4: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant. 70
5.36 Action 4: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant. 71
5.37 Action 4: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant. 71
5.38 Action 4: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant. 72
5.40 Action 5: pick and place the entire assembly on the shelf 72
5.39 The *kinematic wear* trend for all the monitored joints during the performance of a_4, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_4 (bottom). 73
5.41 Action 5: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant. 74
5.42 Action 5: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant. 74
5.43 Action 5: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant. 75
5.44 Action 5: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant. 75
5.45 Action 5: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant. 76
5.46 The *kinematic wear* trend for all the monitored joints during the performance of a_5, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_5 (bottom). 77
5.47 The *kinematic wear* trend for each joint during the performance of a cooperative assembly. The increasing trend occurred when the action was allocated to the human operator, the decrease when the action was allocated to the robot. 78

5.48 Experimental Setup. The workstation had a "L" shape, the assembly pieces were placed at the operator's left as well as the robot co-worker; in front of the human, the assembly area and the monitor to read the allocations were located. The human wore the Xsens to record the joint angles. 79
5.49 Pick and place object *J* action assigned to the human worker. 81
5.50 Pick and place object *J* action assigned to the cobot. 82
5.51 Results by collecting the subjects' answers to the NASA (Task Load indeX) TLX questionnaire. The bars height correspond to the mean value of the subjects score for each task aspect: Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Performance (P), Effort (E), Frustration (F). The two matched bars are for the experiment in *H only* (blue one) and *HRC* (red one) mode. 85
5.52 How many times each action has been allocated to the robot during the cooperation. Considering all the subjects (twelve), each action was executed either by the human or by the robot for 120 times in total. On the top of each bar the number of allocation to the robot for each action in percentage is expressed. 85
5.53 Results by collecting subjects' answers to the custom questionnaire. On the x-axis the question number, on the y-axis, the average score. A positive score means a positive answer and vice versa. Questions were designed to evaluate: granted Ergonomics, Role Allocation efficiency and Usability of such a framework. 87

List of tables

2.1 Number of nodes in the Directed Graph and in the AOG in case of weakly connections among assembly parts. 18

3.1 Definition of *mental effort* and *stress level* factors by Lagomarsino et al. (2021). 33

5.1 Action costs for the human worker and robot 49

5.2 From *i*-th joint configuration to the corresponding segment movements. The two columns "+" and "-" indicate the movement described by a positive and negative rotation around the corresponding axis. 56

Chapter 1

Introduction

The introduction of collaborative robots (cobots) in the industry dates back to the end of the 90s (Akella et al. (1999)) and envisions humans and robots not only to coexist in a fenceless environment but also to share tasks, working simultaneously in the same workspace. This new scenario overcomes the old industrial robot paradigm, describing a *Smart Production*, one of the Industry 4.0 mainstays. The study of collaborative processes in which human and robot agents work together to achieve shared goals in safety is known as Human Robot Collaboration (HRC).

This first chapter has the purpose of introducing the thesis topic, starting with the motivations behind this research (section 1.1). Some background is given by presenting the state of the art (section 1.2), the main contributions of this work are stated in section 1.3, while in section 1.4 an overview of the developed framework is given together with the outline of the thesis.

1.1 Motivations

The research trend in collaborative robotics is motivated by different considerations. First, while the human can adapt his/her action to the task requirements and can handle unexpected events, robots are very precise, can tolerate heavy loads and their performance does not decrease over time. Therefore, by mixing humans and robots in a heterogeneous team, it is possible to exploit the skills of all the agents to maximize team performance. One of the first attempts to distinguish between human and robot skills was proposed by Fitts (1951). He made a list of what man can do better than a machine and vice versa: detection, perception, judgement, induction, improvisation, and long term memory were considered human virtues; speed, power, computation, replication, simultaneous operations, and short term memory

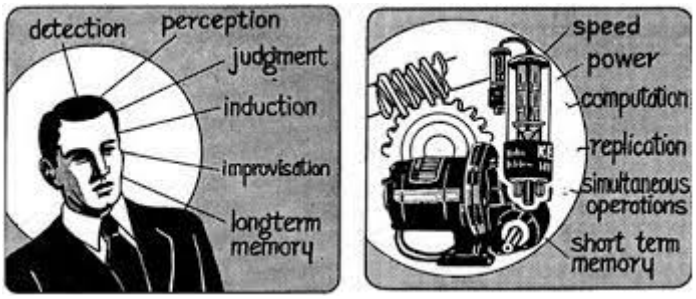


Figure 1.1 Human and Robot skills according to Fitts (1951). Original illustrations.

denoted, instead, as machine features. Although this list is outdated and some assumptions might not be true anymore, the basics concepts are still valid (see Figure 1.1).

Moreover, such robotic platforms capable to collaborate with humans are convenient especially in scenarios of small and medium-sized enterprises (SMEs), characterized by flexible and agile manufacturing requirements, since cobots, compared to the highly specialized platforms, are fast re-configurable and adaptable to the new product demand.

Even though cobots have design characteristics that make them suitable to stay closer to the human, such as the lightweight construction materials, the soft architecture, the speed and torque limitations, these robots are not intrinsically safe. An essential requirement of a robot programmed for direct interaction with human users or uncertain environments is that it must in no case pose a threat to the human: the problem of safety in pHRI and the related open issues have been extensively discussed in literature (Haddadin et al. (2009); Santis et al. (2008)).

A recent trend in collaborative robotics research is devoted to the design of collaboration strategies that take into account the ergonomics requirements typical of industrial applications (Maurice et al. (2017)): such a designed strategy would have the potential to free human operators from risky tasks, avoiding work-related musculoskeletal disorders (WMSDs), which still represent the biggest problem in terms of absenteeism, and, hence, lost productivity among workers in industries (Govaerts et al. (2021)).

Despite collaborative solutions have high potential in bringing several benefits in the manufacturing and logistic processes, their exploitation in the workcell deployment is limited. Some reviews about the already conducted studies and on the future challenges in the HRC realm are written by Ajoudani et al. (2018); Cherubini et al. (2016); Maurtua et al. (2017); Villani et al. (2018): researchers have tried to close the gap between the industrial requirements and the HRC paradigm working on human–robot interfaces, robot control

modalities, robot intelligence, and robot perception of the human physical status to make ergonomic the collaboration.

This thesis purpose is to pursue in this direction, in particular by investigating the development of a new method to ensure an ergonomic and fruitful HRC. The main idea is to profit from quantitative measurements, encoding the human worker status, and from task characteristics to optimize the collaboration at the team-level, i.e. intervening on the role distribution within the mixed team.

Therefore, hereafter some background is given about multi-agent role allocation, one of the main problems in the design of a collaboration strategy. The second part of the state of the art, instead, shows the most famous ergonomics assessment methods in literature, and how some of the previous studies integrate ergonomics principles in their HRC frameworks.

1.2 State of the Art

In the scenario of cooperative multi-agent systems a lot of work has been already done, in particular in the case of **multi-robot systems**. When the researchers' interest shifted from single robot systems to multi-robot systems, the coordination of the whole team became a priority. Many studies regarding this problem have been conducted in fields as exploration, monitoring, transportation, robotic soccer (Farinelli et al. (2004)). An important aspect of the team strategy consists in determining which agents should execute which tasks to accomplish the desired final objective. This problem in literature is referred to as role allocation.

1.2.1 Multi-Agent Role Allocation

Treating the role allocation problem requires, first of all, a definition of the concept of role. In literature, the distinction between *role* and *task* allocation problems (Campbell and Wu (2011)) is a source of debate. Krieger and Billeter (2000) define the task as what specifies "what has to be done" while role as "the task assigned to a specific individual within a set of responsibilities given to a group of individuals". Gerkey and Matarić (2003) recognize a difference in the significance of the two words, but state that the two problems are equivalent.

Having defined roles in multi-agent systems has great importance for at least two reasons. First of all, in this way, agents can become experts in the behaviors for which they are responsible. For example, it could be possible to design many simple robots able to perform a limited number of tasks for which they specialized, rather than constructing an expensive, monolithic single-robot system. Moreover, if agents can learn through their experience,

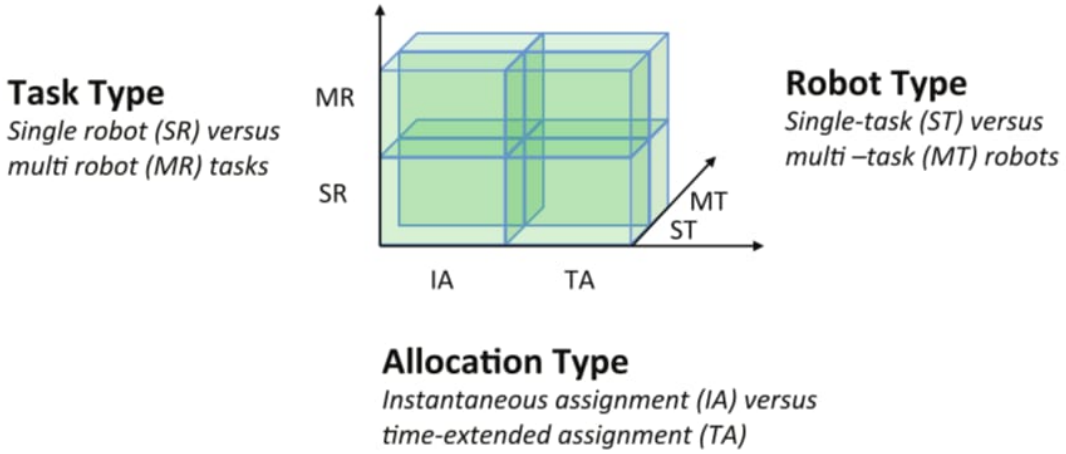


Figure 1.2 Visual representation of the three axes of Gerkey and Matarić's taxonomy (Korsah et al. (2013))

the repeated execution of behaviors specified by their role can improve their performance. Another reason consists in the advantage of having a less number of agents that compete for the execution of the same task, since having specialized agents reduce their capability of performing every task. This effect is beneficial in multi-robot systems: less communication and reduced physical interference.

Modelling the Role Allocation Problem Gerkey and Matarić (2004) carried out a formal study of multi-robot task-allocation (MRTA) problem (since "*the underlying problem [of task and role allocation] remains the same*") and proposed a domain-independent MRTA taxonomy. In this study, the authors refer to tasks as sub-goals necessary to achieve the completion of the overall goal. Each task is considered as independent from the others, a strong assumption relaxed by later contributions (Korsah et al. (2013)). Figure 1.2 shows the three axes generating the Gerkey and Matarić's MRTA space. The first variable of a MRTA problem is the robot type: the robot can perform at most a single task (ST) at a time or multiple tasks (MT) simultaneously. The second variable is the task type: the task may need a single robot (SR) or multiple robots (MR) to be achieved. The last variable is the allocation type and separates problems dealing with instantaneous assignments (IA) of tasks to robots and problems concerned with time-extended assignments (TA), i.e. each robot is allocated several tasks, which must be executed according to a given schedule. Therefore, task and agent properties and the desired allocation type place each role allocation problem in the MRTA space.

In literature, there exist three computational models for the role allocation problem (Iterative Optimal Assignment Problem (OAP), Extended Generalized Assignment Problem (E-GAP), Role-based Markov Team Decision Problem (RMTDP)); such models describe the problem with different levels of generality. The simplest one is the OAP (Gale (1989)), that, although restrictive, returns solutions in polynomial time for problems of the (ST-SR-IA) class. To explain how the OAP works it is necessary to introduce the concept of *utility*. It is a scalar value that describes an agent's suitability to execute some particular role and it is a key concept to many role allocation procedures. Utility can be the "agent offer" in auction-based methods for role assignment (Dias et al. (2006)), it can be the variable of objective functions to optimize (Abramson et al. (2005), Agüero et al. (2006)), or a way for agents to decide between roles autonomously (Chaimowicz et al. (2002)). To compute an agent's utility is not trivial, since it depends on the agent's capability for the role responsibilities, on the priority of that role, and on the current and possible future needs of the team.

The OAP solves problems with n agents and n roles, and a non-negative utility value for each agent-role pairing. The goal is to find the optimal matching between agents and roles, maximizing the utility of the whole team and ensuring that each agent is assigned to exactly one role. In a multi-agent system, this means that each agent computes its utility for all roles and communicates the resulting values to the others; then an optimization algorithm is run to determine which role is more appropriate for each agent. If the environment periodically changes, also the utility values may change. In such a case, once the values have been updated the optimization algorithm must be called again.

From Multi-Robot Teams to Mixed Teams The technical report by Bauer et al. (2016) shows five different levels of human-robot engagement currently existing in the physical Human Robot Interaction (pHRI) context (see Figure 1.3). The old paradigm of industrial robotics is pictured as first, showing the human unable to enter the robot workspace defined by physical fences, for safety reasons; in the coexistence scenario the robot is out of the cage, but the workspace is not shared; in the third case the collaboration foresees sequential actions, meaning that only one of the interaction partners is present in the shared workspace at any one time; in the fourth setup both the agents work simultaneously within a common work-zone; while the last scenario is intended to be an adaptive collaboration, i.e. the robot can respond in real-time to the worker's movements. The closer the human-robot interaction is, the higher are the requirements in terms of safety. The problem of safety is paramount for a successful collaboration and it is largely discussed in the literature (De Santis et al. (2008); Haddadin et al. (2009)). Moreover, robot awareness of the human's status and intentions

gains importance through the interaction levels and becomes fundamental in the collaborative approach: the strict division between the worker's manual work and the robot's automated work is abolished, and cobots should be able to understand the operators' needs to help them.

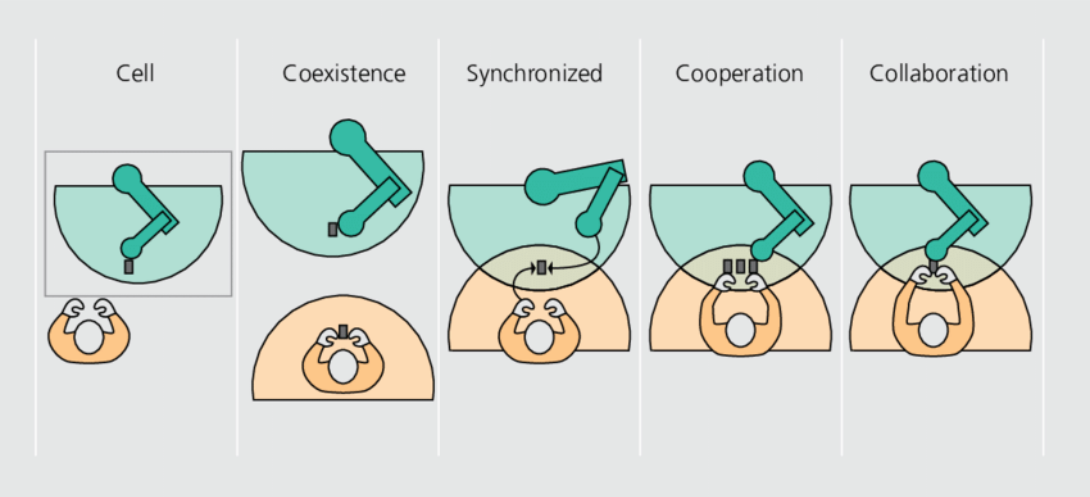


Figure 1.3 Engagement levels in human-robot interaction (Bauer et al. (2016))

Role Allocation Strategies in Human-Robot Teams The role of each agent of the team can be embedded at the control-level or could be planned beforehand. In the latter case, the collaboration plan is established by analyzing and describing the collaborative task as well as the team agents' skills. The typical approach consists in designing functions able to describe the quality of agent-task pairing and through an optimization algorithm find the optimal solution, as for the multi-robot systems. To generalize from a multi-robot scenario to a mixed team of agents (both robots and humans), it is necessary to define an utility function also for the humans: it has to be consistent, i.e. comparable with the ones representing the suitability of robots for roles. The utility function for human agents can take into account postures the human could assume performing each task, loads he/she could handle, task repetitiveness, safety constraints according to the level of human engagement during the pHRI.

Pearce et al. (2018) used the Strain Index (introduced by Steven Moore and Garg (1995)) that considers the intensity of exertion, the duration of exertion, efforts per minute, hand/wrist posture, speed of work, and duration per day. The optimizer provides a set of schedules that enable to consider trade-offs between minimizing time and physical strain on the human worker.

Bänziger et al. (2018) presented a fitness function, which, as the key indexes for an optimal allocation solution, envisions: the task progress (the ratio of the number of completed actions

to the total number of actions in a workplace), the waiting time (the difference between the times the worker and the robot need to finish their tasks), and the traveled distance (the sum of the traveled distance of the worker and the robot), which are calculated in the simulation. In this approach the human–robot task allocation is optimized using genetic algorithms.

Other methods have been developed to dynamically schedule and allocate tasks to workers by minimizing the overall execution time (Casalino et al. (2019); Fusaro et al. (2021)). Nevertheless, these methods cannot account for the variety of different assembly sequences that complex tasks might impose. To overcome this limitation a possible approach is that of enlarging the usage of AND/OR Graph (AOG)s to embed the role allocation problem (Johannsmeier and Haddadin (2016), Darvish et al. (2018)). The AOG data structure was introduced by De Mello and Sanderson (1990) for describing a well-structured task and its decomposition into a sequence of actions, and it is an efficient representation of a state transition graph, with few nodes, that allows an easy search for the feasible assembly plans (see section 2.1). The idea is that of assigning a cost to each transition, encoding the effort of performing that action by each agent in the team, then, by means of a search algorithm the optimal assembly plan and role distribution can be found (see section 2.2). In literature, AOGs are widely used to treat task decomposition problems, while the agent characteristics and the factors that should be considered in solving the task allocation problem in industrial HRC are less investigated.

Lamon et al. (2019) proposed three measurable indexes to better describe the agents' nature, their kinematic and dynamic characteristics, leading to a capability-aware role allocation. The task complexity index describes task properties, defining the agent suitableness to that specific task. The other two indexes are agent dexterity, that measures the agent's capability of planning and moving in the environment, and agent effort, that evaluates ergonomics and safety constraints; such indexes allow selecting the most suitable agent among the ones that can execute the task. The three indexes are combined by means of a weighted sum, obtaining a proper heuristic for the search algorithm. In Figure 1.4 the flow diagram for the roles assignment is presented. First, the task properties are analyzed by decomposing it in many sub-tasks (actions); then the cost of each agent in performing each single action is evaluated through the three designed indexes; finally the roles can be defined according to a cost minimization algorithm.

However, a limitation for such AOG approaches is that they provide the role allocation beforehand, i.e. the agents are allocated tasks before starting the collaboration; then the result of the allocation remains fixed for the whole teamwork. A great contribution to the

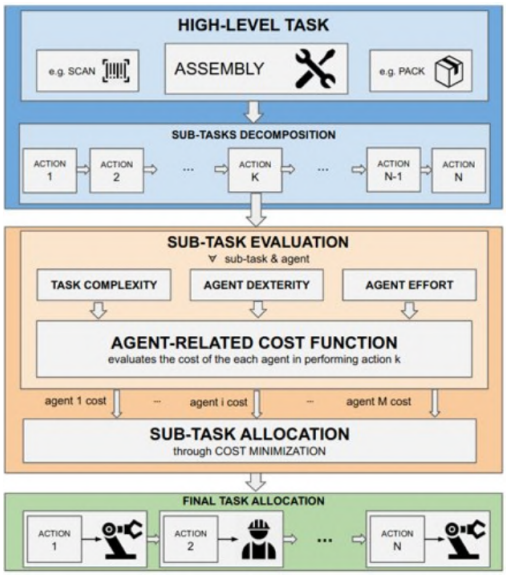


Figure 1.4 Capability-aware Role Allocation Lamon et al. (2019)

researchers' community would be making the AOG-based allocation dynamic, aware of changes in agents' state and in the environment during the collaboration.

On the other hand, it is possible to embed the teamed performance optimization at the control-level. At such level the robot motion control strategies, guiding the real-time robot behavior, are implemented. The recent advances of technology in the hardware design of torque-controlled robots have enabled the development of control algorithms, such as impedance and admittance control (Albu-Schäffer et al. (2007)), that allow cobots to perform a vast number of tasks. However, to optimize the teamed performance, collaborative control strategies tend to overfit the specific task, and sometimes it is not simple to re-use the same algorithm for different tasks. For instance, Mörtl et al. (2012) coped with cooperative load transport tasks, by implementing an algorithm that optimizes the performance in terms of human effort: the same strategy would be not suitable to optimize a different task. The collaborative solution implemented by Peternel et al. (2018) has the same limitation. The physical behavior of the robot is adapted online to the human motor fatigue, measured through EMG sensors, placed at the human shoulder joint. However, according to the executed task (material sawing or surface polishing), the control policy for optimizing the performance changes. Some manipulators involved in HRC processes are presented in Figure 1.5.

Figure 1.5 (left) Lightweight robotic arm Kuka LWR 4+, (right) Lightweight robotic arm Franka Emika Panda equipped with its standard gripper.

1.2.2 Ergonomics in Human-Robot Collaboration

Work-related musculoskeletal disorders (WRMD) are the single largest category of work-related diseases in many industrial countries (Punnett and Wegman (2004)). Repetitive high loads, mechanical compression, vibrations, improper working postures affect the workers every day, negatively influencing their health and performances and, in extreme cases, forcing them to be absent from work. Moreover, a certain portion of workers is represented by advanced age people, which are even more sensitive to heavy working conditions and predisposed to suffer from physical diseases. This scenario encourages the development of solutions to decrease the operators' cognitive and physical load, among which HRC recently proved to be very promising. Indeed, proper control, planning, and perception algorithms aim to make cobots able to understand the human agent's status in terms of physical and mental effort and to adapt their behavior to mitigate it.

In order to have a HRC application apt to temper the workers' burden, the robot should employ human ergonomics data: in literature, numerous approaches can be found for the assessment of workers ergonomics that can be used for such a purpose. The majority of them focuses on the physical load evaluation. One of the most widely used methods is Rapid Entire Body Assessment (REBA) (Hignett and McAtamney (2000)), which considers kinematic information and is based on the observation of the human body configurations. Each joint angle is associated with a score corresponding to a risk of injury/harmful effects: the higher the score, the higher the risk. As the same scores given to, e.g. the upper arm and the trunk, are not equivalent, the method provides comparative tables to assign a final score that ranges from 1 (no risk of WRMDs) to 12 (very high risk, it is recommended to change the posture).

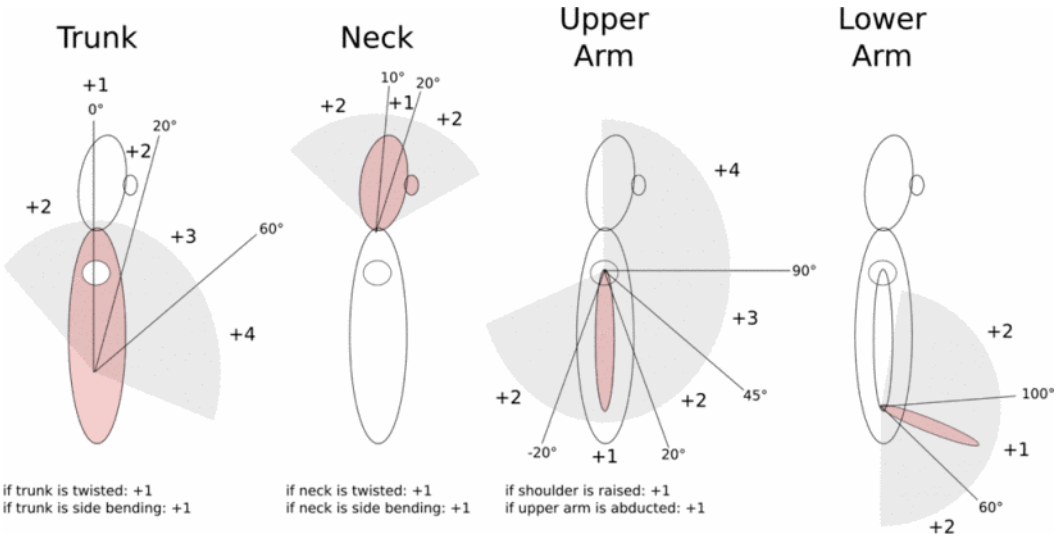


Figure 1.6 Result of REBA calculation for the trunk, neck, upper arms and lower arms (Busch et al. (2018))

In Figure 1.6, an extract of REBA tables is presented: in particular, the result of REBA calculation for the trunk, neck, upper arms, and lower arms discriminates between safe and dangerous ranges of joint angles according to the assigned score.

Originally this method was introduced for pen and paper assessment of industrial workflows, directly from the observation of workers' activities, but the evaluation of the joints position and the computation of the corresponding REBA score can be automatized (Busch et al. (2017); Van de Perre et al. (2018)). Busch et al. (2017) obtained relevant data for the planning phase of the ergonomic human-robot interaction by minimizing a cost function, including also the worker safety level (computed with the extracted REBA scores). A similar approach is used by van der Spaa et al. (2020) for finding the best plan for ergonomic hand-poses in a co-manipulation task. Busch et al. (2018), in a later study, introduced a differentiable version of REBA (dREBA), to directly optimize original REBA scores.

Ovako Working posture Analysing System (OWAS) (Karhu et al. (1981)) and Rapid Upper Limb Assessment (RULA) (McAtamney and Corlett (1993)) are other standard assessment methods which estimate an absolute level of risk depending on kinematic data, similarly to REBA. Faber et al. (2017) used both measurements for establishing the ergonomic cost of assembly steps in manufacturing operations, in order to meet the optimal assembly sequence.

An alternative approach still exploiting kinematic data is proposed by Bestick et al. (2015). They presented a collaborative handover framework able to consider the subject-specific kinematics of the involved subject. The traditional methods are not used and motion

capture data allow the estimation of a personalized kinematic model: such custom interaction enhances the human collaborator ergonomics.

Nevertheless, the knowledge about kinematics might be insufficient to regulate fruitfully HRC requiring the interaction with objects and with the environment, since data regarding forces are needed as well. There already exist methods including static forces: National Institute for Occupational Safety and Health (NIOSH) equation (Waters et al. (1994)) or Washington Industrial Safety and Health Act (WISHA) indices. The latter provide the assessments of risk level during a lifting task by considering human kinematics, but also weight of the lifted object, number of lifts per minute and duration of the whole task. The resulting ergonomics indicators are task-specific, so they do not accurately cover all kinds of activities that may be addressed by cobots.

In principle, this limitation could be coped with by using a tool for the estimation of human dynamics, which can be extended to any generic activity. With this aim, many software packages for musculoskeletal modeling have been developed, that allow for complex motion simulations and their biomechanical analysis. Commonly used packages are, for instance, the AnyBody Modeling System (Damsgaard et al. (2006)) and the open-source software system OpenSim (Delp et al. (2007)). Nevertheless, the provided dynamic musculoskeletal models are so much detailed, that it might be ambitious to exploit them in designing ergonomic HRC frameworks since it would require heavy computations for processing too accurate data, even nonfunctional. A solution may be the design of simpler dynamical models. For instance, Peternel et al. (2019) employed a human arm model for estimating the muscle force given the arm configuration and the endpoint force, during the performance of drilling and polishing tasks. In this study, the estimated muscle force is then mapped into a muscle fatigue level, which drives the selection of a fatigue management protocol for the robot controllers.

In Peternel et al. (2018) framework, the measurements of EMG sensors, placed on the muscles of the shoulder joint, allow the manipulator to predict human motor intentions, enabling a dynamic adaptation of the robot effort during the task performance.

Alternatively, Kim et al. (2017a) used a force plate for measuring the displacements of the Centre of Pressure and the Reaction Ground Force in the presence of external forces, to estimate and monitor the human joint torque overloading (see section 3.2).

Even though the majority of studies for an ergonomic HRC target the reduction of human physical load, the mental and psychological factors should not be disregarded. Indeed, due to the increasing complexity of the production settings, the worker productivity and the efficiency of the overall HRC can be negatively influenced by an extra cognitive load. For example, the framework proposed by Rajavenkatanarayanan et al. (2020) used heart rate and

skin conductivity measures to evaluate such cognitive burden, enabling robots to understand more about their human teammates.

As it can be deduced, the worker's ergonomics and comfort can be encoded by different quantities according to the collaborative task. Moreover, the choice of planning and control strategies determines the assessment method and the role of ergonomics indicators in HRC architectures. The challenge in industrial scenarios lies in designing a framework able to monitor many quantities to better characterize the collaboration (Maurice et al. (2017)) without burdening the operator with an excessive amount of sensors. The usage of too many sensors is not advisable for industrial applications, since they might obstacle the worker in his/her activities and, at the same time, the operator's movements could make the measurements noisy and unreliable. Especially EMG signals are very prone to noise, as such non-invasive devices are applied on the worker's skin.

1.3 Contribution

This thesis aims to dynamically integrate ergonomic constraints at the team-level of a HRC framework by exploiting the AOG theory. In particular, AOGs model the task decomposition and can be adapted to tackle the role allocation problem. The arc weights can be defined as a function of measured quantities describing the human agent's status and, by monitoring such quantities, they can be updated during the collaboration. In this way, the optimal role allocation might change dynamically. The variables that will be considered as arc weights depend intrinsically on the specific collaborative task.

To overcome the limitation of the state-of-the-art methods, we propose an online role allocation strategy, that can assign actions among the agents of the team according to the physical human-worker status. The main contribution of the method is twofold:

- The introduction of online role allocation within the AOG structure, that, at each step of the assembly task, provides the next action and the allocated worker, as a result of an optimization algorithm AO*.
- The integration of a human joint-level status indicator, named *kinematic wear*, to guide an optimal role allocation. It embodies the subject "ergonomic history", accounting for the usage of each joint during the whole execution of an assembly task. However, due to the adaptability of the designed framework, different ergonomic indicators can be employed.

The performances of the method are investigated with simulations that aim to evaluate its time complexity and with a proof-of-concept assembly. The results show the potential of the strategy in preventing risky actions during human-robot cooperative industrial tasks.

1.4 Framework

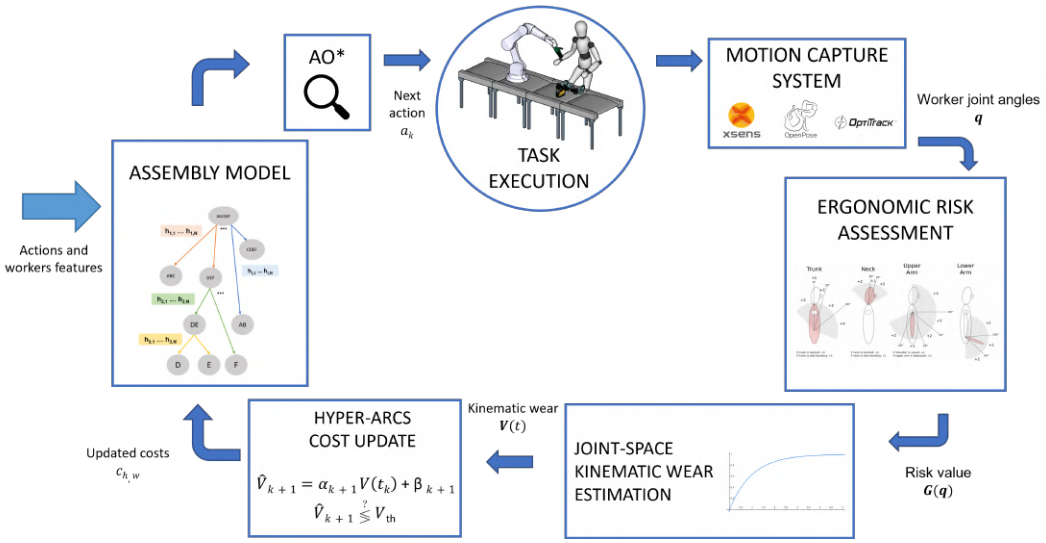


Figure 1.7 Framework scheme.

In this section the HRC framework we designed is presented, with the aim of introducing its main blocks (shown in Figure 1.7) and giving an idea to the readers about the thesis outline.

- The *assembly model* block represents the model we chose to describe the HRC task, taking into account actions and workers features. In section 2.1 and section 2.2 the usage of AND/OR Graph for representing well-structured tasks and for solving the targeted allocation problems is discussed;
- the second block represents the optimal search algorithm we have designed: the AO* algorithm explores the graph finding the optimal assembly sequence and the most suitable agent for the next action, as shown in section 2.3;
- once the human is asked to execute an action, during their performance a motion capture system acquires kinematic data, in particular the upper-body joint angles $q(t)$. These recorded configurations are evaluated with an ergonomic risk score $G(t)$, according to RULA ergonomics assessment rules, one of the most common methods existing in literature;

- the risk values recorded along the action execution are used for updating the *kinematic wear* index of each upper-body joint, that embeds the history of each joint usage without losing information about the most hazardous configuration it has assumed. The design of such index is addressed in section 3.3;
- according to the *kinematic wear* values, the AOG hyper-arcs cost is updated for the next allocation. Such updating procedure envisions the prediction of the entity of the ergonomic risk the human would be exposed in performing each one of the possible future actions. The risk prediction model is defined in section 3.1.

It is important to stress that the just presented framework envisions the *kinematic wear* as risk indicator, since we used it to test the functioning of such dynamic human-robot role allocation method. However, the framework can be paired with other metrics describing the human risk.

Some principles of the discussed method are presented in Merlo et al. (2021).

Chapter 2

AND/OR Graphs for Role and Task Planning

A commonly used approach for representing well-structured industrial tasks, such as assemblies, exploits AND/OR Graph (AOG)s. AOGs are data structures able to model all the possible assembly sequences of an assembly task in a compact representation. Recently, researchers extended the formulation of AOGs to embed also the role allocation problem in human-robot assembly tasks. With such a method, not only the assembly sequence but also the task-agent pairing can be optimized.

2.1 AOG for Industrial Task Planning

This structure is used by De Mello and Sanderson (1990) for describing compactly all the possible assembly sequences for a certain product, which is composed of many pieces. The advantage of using such graphs lies in having fewer nodes and a simpler search for the optimal assembly plan guided by the weights associated with each arc in the graph.

Assembly plans are defined as ordered sequences of operations that change at each step the configuration of parts, starting with all parts disconnected from each other and ending with all parts properly joined to form the desired assembly. A configuration is defined by the subsets of parts that have determined relative position among themselves. The starting configuration is described by as many subsets as the number of single pieces involved in the assembly. Each assembly operation consists in joining two or more pieces together, according to the fixed relative positions among themselves, by increasing the elements in a single subset (a sub-assembly) and decreasing the total number of subsets. A sequence of assembly operations corresponds to an assembly plan.

Introducing some notations, given an assembly P made of M pieces ($P = \{p_1, p_2, \ldots, p_M\}$), a configuration Θ of P is a set of subsets of P such that:

- all the pieces in Θ subsets are assembled with physically feasible and stable connections (the determined ones);
- each p_i belongs to one subset of Θ;
- none p_i belongs to two or more subsets of Θ.

The assembly plan can be seen as a sequence of many Θ, starting from $\Theta_i = \{\{p_1\}, \{p_2\}, \ldots, \{p_M\}\}$ and ending with $\Theta_f = \{\{p_1, p_2, \ldots, p_M\}\}$. If all the possible configurations Θ of P are interpreted as nodes and all the assembly operations that describe the transition between two configurations as edges of a *Directed Graph* it is possible to find a path from Θ_i to Θ_f using path search algorithms. Since some assembly operations could be feasible but could not lead to a solution, it is more advantageous to perform the research backward, i.e. from Θ_f to Θ_i. In this way, the search may be viewed as a decomposable production system in which the problem of disassembling one assembly is decomposed into distinct sub-problems each one consisting in disassembling one sub-assembly. This idea leads to the formalization of AND/OR Graph (AOG)s and the search is redefined as a tree search over it.

The nodes of AOG correspond to subsets of P, i.e., each sub-assembly characterized by fixed physically feasible connections among its pieces. The arcs, called *hyper-arcs* are pairs in which:

- the first element is a node (father);
- the second element is a set of nodes (children) describing the sub-assemblies that can be obtained with a disassembly action applied to the first element.

Introducing some notation, an AOG is described by a set of nodes $N = \{n_1, n_2, \ldots, n_{|N|}\}$ where each n_i corresponds to a subset of p_i firmly connected and a set of hyper-arcs $H = \{h_1, h_2, \ldots, h_{|H|}\}$. Each node $n \in N$ represents a state of the decomposed task, while hyper-arcs define the transitions between states. Each $h \in H$ describes a many-to-one transition, meaning that it connects a set of child nodes with a father node. In the case of assembly tasks, considering that most of the assembly operations join two sub-assemblies, hyper-arcs are modeled as two-to-one connectors. The child nodes connected by the same h are in a logical AND, while different hyper-arcs with the same parent node are in logical OR. To obtain the same assembly configuration $\bar{\Theta}$, many pairs of sub-assemblies to join may exist: the two sub-assemblies in each couple are in AND, since both are necessary to reach $\bar{\Theta}$; while among

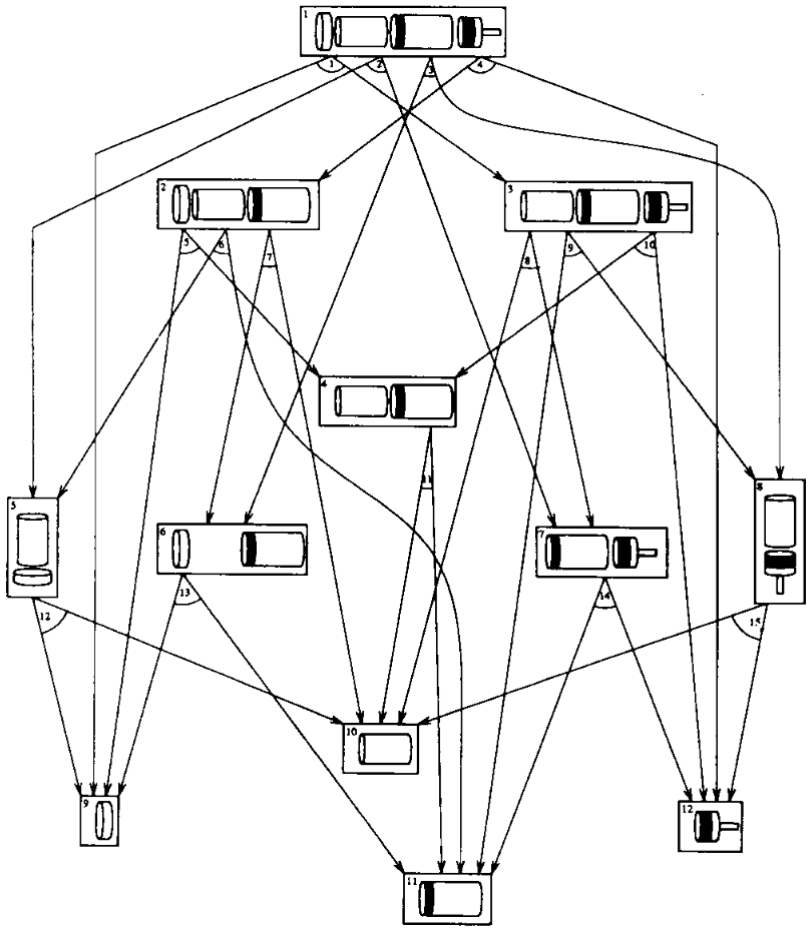


Figure 2.1 AND/OR Graph for a simple product De Mello and Sanderson (1990).

such couples there is an OR relation, because just one of the assembly operation that they model can be executed to have $\bar{\Theta}$.

The only node without a father is named root. The nodes without children are identified as leaf nodes and they are as many as the assembly pieces. To construct an AOG it is necessary to know all the possible feasible assembly actions since they determine the number of nodes.

However, to show the power of AOG representation and to give an idea of how many nodes would be handled according to the number of assembly pieces, De Mello and Sanderson (1990) present two cases:

1. all the assembly parts are connected with the other ones: strongly connection case;

2. the p_i piece is connected only with p_{i-1} and with p_{i+1}: weakly connection case. This means that having M pieces leads to $M-1$ connections.

M	Directed Graph Nodes	AOG Nodes
10	512	55
12	2048	78
15	16384	120

Table 2.1 Number of nodes in the Directed Graph and in the AOG in case of weakly connections among assembly parts.

The second case is more typical of real industrial assemblies in the case of a high number of pieces. In Table 2.1 are listed the number of nodes in case of 10, 12, and 15 pieces for the two assembly representations (*Directed Graph* and AOG, respectively). The more the number of interconnections increases, the more the number of nodes rises, as well as the convenience in choosing the AOG approach.

A graphical representation of an AOG example is given in Figure 2.1. The root node at the top of the picture has no fathers and it is represented as a set with all four assembly parts. Four hyper-arcs are leaving this node: each one links the father with two nodes representing the two sub-assemblies that can be obtained by performing such a disassembly operation. These four hyper-arcs are in logical OR since just one of these actions can be performed. Similarly, each of the other nodes is connected through a hyper-arc with two child nodes describing one possible result of its decomposition. It is important to highlight that there are no nodes that appear more than once in the graph: the fact that the same sub-assembly can be obtained by many disassembly operations is described by that node reached by many hyper-arcs. For example, the sub-assembly represented by node 5 can result from two different operations encoded by hyper-arcs 2 and 6. There are many paths from the root (node 1) to the leaf nodes (9, 10, 11, and 12). Knowing which are the possible sub-assemblies allows us to evaluate each operation effort and assign to each of the hyper-arcs a cost encoding such effort. The hyper-arcs cost guides the search through the optimal solution, returning a tree (like the one in Figure 2.2) which describes the best order of actions for completing the assembly (see section 2.2 and section 2.3).

Some researchers treated the AOG structure considering nodes as sets of processes to be solved before moving to another node, and transitions as sets of actions to be completed before reaching the connected node (Darvish et al. (2018)). This novel schema might manage to model various collaborative tasks and not only assembly.

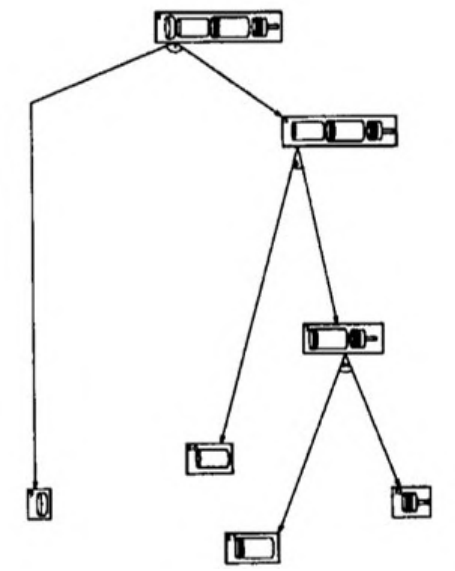


Figure 2.2 Disassembly plan using AOG De Mello and Sanderson (1990).

2.2 AOG for Role Allocation

Recently, this data structure has been employed for handling both the planning and the role allocation problems. By combining them, a unique optimization process can return the best assembly sequence and the optimal task distribution.

The framework proposed by Johannsmeier and Haddadin (2016) envisions the AOG structure for representing human-robot collaborative assembly problems. The formalization of the AOG is a bit different from the one proposed by De Mello and Sanderson (1990) that is also the one used in this thesis, but it provides an idea of how to approach the allocation problem. In particular, they design the assembly actions not with hyper-arcs but with other nodes, named AND nodes (the red ones in Figure 2.3), while they call OR nodes the sub-assemblies (the blue ones). They state that each assembly action can be performed by all the workers involved in the cooperation and that according to some quantities it is possible to establish how much an action costs to each worker: for the human, the cost is determined by cognitive and physical workload, while for the robot the metrics are the execution time and power consumption. In literature, the agent characteristics and the factors that should be considered in solving the task allocation problem in industrial HRC are investigated by Lamon et al. (2019). According to the Johannsmeier and Haddadin (2016) AOG definition, each AND node has as many costs as the participating workers: through an A* search on the AOG it is possible to find the optimal path and role allocation, by visiting the AND nodes many times, each time considering a different cost.

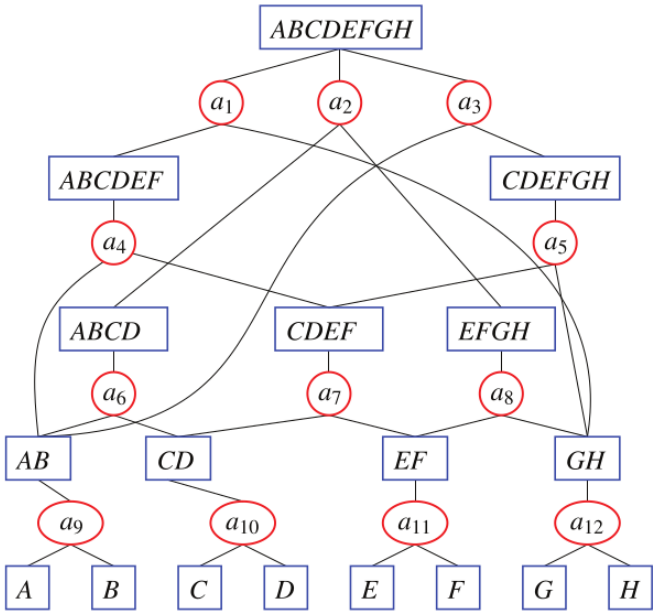


Figure 2.3 AOG according to Johannsmeier and Haddadin (2016).

In this thesis, inspired by the just mentioned method, to embed the role allocation problem in our AOG formulation, two additional sets are defined: the set of workers involved in the collaboration $W = \{w_1, w_2, \ldots, w_{|W|}\}$ and the set of assembly actions $A = \{a_1, a_2, \ldots, a_{|A|}\}$ that workers have to perform. From now on, actions depict both proper assemblies (e.g. screwing two pieces together) and 'relaxed' assemblies (e.g. move an object on top of a table could be considered as an assembly between such an object and the table). As already stated, the desired assembly sequence among all the possible ones that the AOG describes can be computed through the assignment of a cost to each $h \in H$ $(c_{h_1}, c_{h_2}, \ldots, c_{h_{|H|}})$. Since we describe a cooperation scenario, the same action has a different cost for each worker. Unlike the formalization by Johannsmeier and Haddadin (2016), instead of assigning many costs to the same action, the same hyper-arc (that describe an assembly operation) is duplicated for $|W|$ times and a cost, that represents the suitability of w_i to that action, is assigned to each of them. Such values can encode the complexity of performing each assembly action, and, by exploiting an optimal-based search algorithm, the path with the minimum total cost can be found (see section 2.3). In Figure 2.4 is presented a general AOG of an assembly made by six pieces A, B, C, D, and E. Each hyper-arc has a different color, since it represents a different assembly action. The rectangle with the same color of the hyper-arc (e.g. the red one at

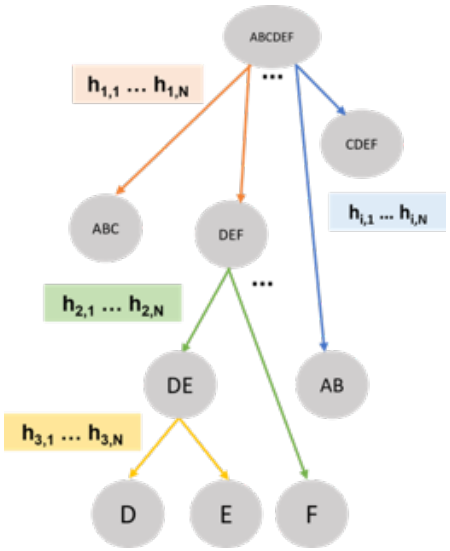


Figure 2.4 General AOG showing the duplication of hyper-arcs for introducing in such structure the role allocation problem. Each of these additional arcs has a different cost according to the suitability of each agent in performing the corresponding action.

top - left) means that the action can be performed by N workers, therefore that hyper-arc is duplicated for N times.

Moreover, Johannsmeier and Haddadin (2016) set AND nodes costs once fixing them for the whole duration of the assembly task. Therefore, it is not designed for a dynamic environment, which can influence the cost associated with workers and actions, and the allocation established at the beginning of the cooperation would not change. Our framework, instead, envisions monitoring the cooperation and updating the costs to ensure the role assignment optimality at each instant.

2.3 AO* for the Optimal Search

To retrieve the desired assembly sequence and the optimal action allocation, a custom AO* search is implemented, since, unlike the standard AO*, the goal of the algorithm is to inspect all the leaf nodes, as they represent the atomic pieces which are all used in the assembly. Such custom algorithm is inspired by the A* functioning.

A* is a best-first search algorithm: given the root node and a goal node it returns the optimal path between them, i.e. the one with the smallest cost (Doran and Michie (1966)).

Algorithm 1 AO* search

Require: root (node), goals (nodes)

```
 1:
 2: SearchState start, goal
 3: start.setState(root)
 4: goal.setState(goals)
 5: Put start in OPEN list
 6:
 7: while !(OPEN.empty()) do
 8:     select the list member current with the lowest cost
 9:
10:     if current state is the goal one then
11:         break
12:
13:     for each node (j) inside the state of current do
14:
15:         for each hyper-arc (i) whose the j node is father do
16:             SearchState successor
17:             successor = current
18:             substitute the j node in the state with the two children of j
19:             successor.setHyperArc(h(i))
20:             successor.setCost(h(i).c)
21:             successor.setFather(current)
22:
23:             if successor is already in OPEN with a greater cost then
24:                 substitute that element with successor
25:
26:             if successor is already in CLOSE with a greater cost then
27:                 remove that element from CLOSE and add successor to OPEN
28:
29:             if successor is not in OPEN nor in CLOSE then
30:                 add successor to OPEN
31:
32:     remove current from OPEN and add it to CLOSE
33:
34: opt_path.includeArc(current.h)
35: opt_path.includeNodes(children of current.h)
36: SearchState tmp = current.father
37:
38: while !(tmp == start) do
39:     opt_path.includeArc(tmp.h)
40:     opt_path.includeNodes(children of tmp.h)
41:     tmp=tmp.father
42:
43: return opt_path
```

At each iteration of its main loop, A* chooses the path that minimizes:

$$f(n) = g(n) + h(n)$$

where n is a successor of the current node on the path, $g(n)$ is the path cost from the start node to n, and $h(n)$ is a heuristic function that estimates how much it costs to reach the goal starting from n. The heuristic function is problem-specific. To be sure A* will return the optimal solution, the heuristic function must be admissible, meaning that it does not overestimate the actual cost to reach the goal.

To let the A* work with our AOG the algorithm 1 is designed. The idea is that of creating a new tree where each node, defined as *SearchState* object, envisions:

- a set of nodes (named state), describing the assembly configuration;
- a hyper-arc (named h), that corresponds to the last action performed to obtain such configuration;
- a cost (named c), encoding how much reaching that configuration costs;
- a link (named father) with its *SearchState* father.

The search acts in a top-down fashion, from the *SearchState* (in the pseudocode named as *start*) which has the root node only in its set of nodes, which represents the state where all the pieces are assembled, to the *SearchState* (in the pseudocode named as *goal*) which has the leaf nodes in its set of nodes. When a node is visited, its expansion consists in creating new *SearchState* objects, describing the assembly configuration that can be obtained starting from the current one. To keep track of those nodes that have been visited but not expanded, a list, named OPEN, is used; while another list is used for the nodes that have been visited **and** expanded, named CLOSE (see Algorithm 1). Once the *SearchState* goal is reached, the optimal path is obtained going backwards to the start one, "father by father", including arcs and nodes in path variable. The desired path is the one that minimizes the sum of the costs of the traveled hyper-arcs. The heuristic we used is $h(n) = 0$, but it is important to highlight that the costs assigned to hyper-arcs embed some information about the further steps to complete the assembly. For the description of the costs meaning and design see section 3.1. To verify the correct functioning of this procedure, a brute force algorithm was implemented, able to find all the possible paths given an AOG, computing the overall cost of each of these paths. The path with the lowest cost computed with the brute force approach corresponds to the one returned by our AO*.

While the optimality of the solution, with fixed costs, is ensured, if costs change during the task execution, the search algorithm should be called after the completion of each action. In such a case, it explores a reduced graph, from the root node to the last reached state. Therefore, each time the AO* search is called, the state of the *SearchState* goal will contain the set of nodes representing the last reached assembly configuration.

The performances of AO* in terms of computational time have been tested and the results are discussed in section 4.2.

Chapter 3

Integrating Ergonomics in Human-Robot Role Allocation

Up to now the task decomposition and allocation model has been introduced, as well as the used optimization strategy to retrieve the best allocation results, conforming to the hyper-arcs cost. Presently, the focus becomes how the ergonomics constraints can be introduced in such a framework. This is achieved by designing the hyper-arcs costs such that they embed the ergonomics information. In this chapter the costs design and the updating procedure are introduced in section 3.1. Then, section 3.2 reports two examples of risk indicators that could be employed in our framework, proving its adaptability to many metrics. Finally, in section 3.3 we present the risk index we formulated and used to test the overall framework performances.

3.1 Risk Prediction for hyper-arcs cost updating

To include the ergonomics data in a HRC framework means to manage to adjust the system behavior to be responsive to human exigencies. The strategy we adopted in our framework consists in embedding ergonomics in the hyper-arcs costs, that encode the risk associated to the execution of each action starting from the current conditions of the human status. In order to implement such a strategy we need:

- a system monitoring some human variables (body postures, muscle activity, self-touching...), according to a chosen risk indicator function, during the whole duration of the task execution. In this work we refer to such indicator as $V(t)$;

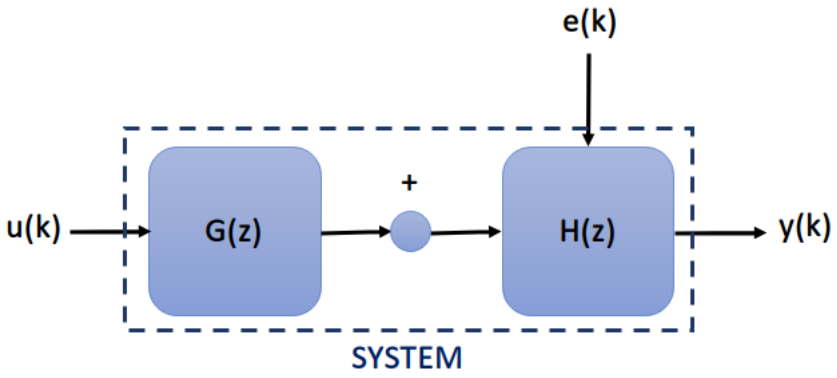


Figure 3.1 An arbitrary LTI system in discrete time, with an exogenous input u, a disturbance e, and an output y. The two input signals are filtered with transfer functions $G(z)$ and $H(z)$, respectively.

- a model that processes the monitored variables through the appropriate ergonomic rules, returning a risk value $V(\bar{t})$ at instant $\bar{t}$;
- a prediction model able to retrieve the risk level related to each action if it would be executed next $\hat{V}(t_{k+1})$, given the current human worker conditions $V(t_k)$. Here t_k is the current instant (at the end of action a_k) and t_{k+1} the instant in which action a_{k+1} is completed;
- a procedure that translates the predicted risk into hyper-arcs costs.

While the first two items are specific of the formulation of the ergonomic risk function, mapping the human monitored quantities into the corresponding risk level $V(t)$, the last two can be fixed and exploited with different $V(t)$.

Prediction model Given an arbitrary linear and time-invariant Single-Input Single-Output (SISO) system in discrete time with an input $u(k)$ and an output $y(k)$, where k is the sampling instant index, the difference equation which describes the input/output relation is:

$$y(k) = \alpha_1 y(k-1) + \alpha_2 y(k-2) + \cdots + \alpha_n y(k-n) + b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m) \tag{3.1}$$

where it is clear that the value of y at the current instant depends on the previous values of y and on the current and previous values of u.

By assuming $a_i = -\alpha_i$ and interpreting z^{-1} as the one step delay operator ($z^{-i}y(k) = y(k-i)$) (Lennart (1999)), we can rewrite Equation 3.1 as:

$$[1 + a_1 z^{-1} + \cdots + a_n z^{-n}]y(k) = [b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}]u(k). \tag{3.2}$$

Then, recognizing $[1 + a_1 z^{-1} + \cdots + a_n z^{-n}]$ and $[b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}]$ as two polynomials in z^{-1}, $A(z)$ and $B(z)$ respectively, we obtain:

$$y(k) = \frac{B(z)}{A(z)}u(k) = G(z)u(k). \tag{3.3}$$

This is named *hybrid notation*, since both time and frequency are present in the same equation. This notation is valid only if we consider the forced output, while the free response is assumed to be vanished or, at least, negligible. By assuming to have a disturbance $e(t)$ as an additional input to the system, filtered by a transfer function $H(z) = \frac{C(z)}{D(z)}$, Equation 3.3 becomes:

$$y(k) = G(z)u(k) + H(z)e(k) \tag{3.4}$$

and a graphic representation is given in Figure 3.1. $e(k)$ is assumed to be a white sequence: it means that $e(k)$ is uncorrelated with $e(k-r)$, $\forall r > 0$. If the system dynamics is known, that is the coefficients of polynomials $A(z), B(z), C(z)$, and $D(z)$ are known, given $I(k-1) = \{y(0), \ldots, y(k-1), u(0), \ldots, u(k-1), u(k)\}$ it is possible to predict the output at the next instant k, by using Equation 3.4 manipulated like:

$$y(k) = z\left[1 - \frac{C(z)}{D(z)}\right]y(k-1) + \frac{C(z)}{D(z)}\frac{B(z)}{A(z)}u(k) + e(k). \tag{3.5}$$

Since $e(k)$ is a white sequence, the expected value of $y(k)$ is:

$$\hat{y}(k|I(k-1)) = z\left[1 - \frac{C(z)}{D(z)}\right]y(k-1) + \frac{C(z)}{D(z)}\frac{B(z)}{A(z)}u(k). \tag{3.6}$$

If $D(z) = 1$ and $C(z) = A(z)$, equation 3.5 becomes

$$y(k) = z[1 - A(z)]y(k-1) + B(z)u(k) + e(k). \tag{3.7}$$

In literature this is known as the ARX model (Lennart (1999)), where AR and X stand for Auto Regression and Exogenous input, respectively. In this case, the output prediction

formula 3.6 simplifies to

$$\hat{y}(k|I(k-1)) = z[1-A(z)]\,y(k-1) + B(z)u(k). \tag{3.8}$$

In the case we don't know how the system evolves (black box case), we can use the prediction formula also to estimate the model parameters, by exploiting the system history, trying to minimize the error between the actual system output and the predicted one. The ARX model allows to find the parameters with a one-shot solution (non iteratively).

In our case, by exploiting the information carried by the risk indicator $V(t)$ about the human status at the current instant t_k and the a-priori knowledge about the risk associated to each action in the task, it is possible to predict the human conditions $\hat{V}(t_{k+1})$ at the end of each possible future action a_{k+1}. Therefore, rewriting the Equation 3.8 as:

$$\hat{y}(k+1|I(k)) = z[1-A(z)]\,y(k) + B(z)u(k+1) \tag{3.9}$$

the prediction equation we used is Equation 3.9, with $\hat{y}(k+1) = \hat{V}(t_{k+1})$, $y(k) = V(t_k)$, and $u(k+1)$ representing the monitored quantities describing the human status at instant t_{k+1}. Even though the input and output variables may change, such a model can be used to predict the risk future values, both when the system dynamics is known ($A(z)$ and $B(z)$ known) and when it has to be identified ($A(z)$ and $B(z)$ to be estimated).

Hyper-arcs cost update To make the role allocation algorithm able to choose which action is suitable for the human worker given their current status, the ergonomic risk predicted values are used to update the cost of each AOG hyper-arc. Having such costs, that change according to the previous performance and that estimate which would be the human conditions in the future step, the AO* solution can prevent the human subject from executing dangerous actions.

To compute the cost for the hyper-arc h_j, modeling the corresponding future human action a_{k+1}, the formula is:

$$c_{a_{k+1,w_h}} = c_{h_j,w_h} = \sum_{i=1}^{m} \gamma_i, \tag{3.10}$$

where m is the number of monitored joints and

$$\gamma_i = \begin{cases} \hat{V}_i(t_{k+1}) & \text{if} \quad \hat{V}_i(t_{k+1}) < V_{th} \\ \hat{V}_i(t_{k+1}) + \gamma_{high} & \text{if} \quad \hat{V}_i(t_{k+1}) \geq V_{th} \end{cases} \qquad \forall \quad \hat{V}_i(t_{k+1}) \tag{3.11}$$

where V_{th} represents a threshold value, chosen as a certain percentage of V_{max}, the maximum value that $V(t)$ can assume. This means that the j-th hyper-arc cost equals the sum of the predicted risk values at each joint if the action modeled by such j-th hyper-arc would be performed as a_{k+1}. In the case one (or more) $\hat{V}_i(t_{k+1})$ is greater than V_{th}, another addend γ_{high} is involved in the sum, that increases the hyper-arc cost so much that AO* will exclude each path containing the j-th hyper-arc.

Instead of this simple sum, it is possible to introduce a weighted sum with a different weight for each joint, to ensure that some joints prevail in the cost computation.

As discussed in subsection 1.2.1, when two or more agents aim to cooperate in a team, the functions describing their suitability to each task must be comparable to allow the allocation algorithms to deal with consistent data. In our case, the worker's aptitude for performing a specific action is encoded by the hyper-arcs costs. Therefore, we have to assign comparable costs to hyper-arcs modeling human and cobot actions. To accomplish that, as cobot actions cost, a constant $c_{a_{k+1,w_r}} = c_{h_j,w_r} = c_{const}$ of the same order of magnitude of $c_{a_{k+1,w_h}}$ is selected. The choice of its value determines the percentage of robot activity during the collaboration: the lower c_{const} the higher such a percentage and vice versa.

3.2 Ergonomic Risk Indexes

The measured and monitored quantities to keep track of the human risk $V(t)$ are decided by task features and by the required human engagement level. The presented framework is not coupled with any specific metrics, but it remains a valid approach if we manage to:

1. find functions able to describe the risk the human incurs when executing each action in the task,

2. define an index able to describe the current human status for each instant of the task execution,

3. choose proper parameters for the risk prediction model.

As discussed in the state-of-the-art (subsection 1.2.2), the description of the human physical status can be kinematic (by considering only body configurations and postures) or dynamic (evaluating also the effect of the interaction forces); while other kinds of tasks require the measurement of the cognitive load to guarantee a stress level mitigation.

A dynamic risk indicator An example of dynamic risk indicator is based on joint torques variations assessment, that could be used to estimate the human physical status during the execution of heavy manipulation tasks. Kim et al. (2017b) present a new method for the real-time estimation of joint torque variations based on the deviations of the Centre of Pressure (CoP) and Ground Reaction Force (GRF) in the presence of interaction forces.

By computing the difference $\Delta\tau$ between the human joint torque in loaded $\tau_{\mathbf{wt}}$ and natural $\tau_{\mathbf{wo}}$ (without interaction forces except for the GRF) conditions, under the assumption of quasi-static movement (due to the natural motion when handling heavy loads) the overloading joint torque is obtained:

$$\mathbf{S}^\top \Delta\tau = \mathbf{S}^\top (\tau_{\mathbf{wt}} - \tau_{\mathbf{wo}}) = -\left[\sum_{i=1}^{n_f} \left(\mathbf{J}_{C_{P_i}}(\mathbf{q})^\top \mathbf{f}_{wt,i} - \mathbf{J}_{\hat{C}_{P_i}}(\mathbf{q})^\top \mathbf{f}_{wo,i}\right) + \sum_{j=1}^{n_h} \mathbf{J}_{a_{h_j}}(\mathbf{q})^\top \mathbf{f}_{h,j}\right] \tag{3.12}$$

where:

- $\mathbf{q} = [\mathbf{x}_0^\top \quad \theta_0^\top \quad \mathbf{q}_h^\top] \in \mathbb{R}^{6+n}$ represent the generalized coordinate of the system with $\mathbf{x}_0$ and θ_0 are the position and the orientation of the base frame $\{0\}$, at the pelvis level, with respect to the inertial frame $\{w\}$ (the human model is floating base), while $\mathbf{q}_h = [q_1 \dots q_n]^\top \in \mathbb{R}^n$ is the vector with the angular position of n revolute joints modelling the human body joints;
- $\mathbf{S} = [\mathbf{0}_{n\times 6} \quad \mathbf{I}_{n\times n}] \in \mathbb{R}^{n\times(6+n)}$ is the actuation matrix;
- $\mathbf{f}_{wt,i}$, $\mathbf{f}_{wo,i}$ are the vertical GRF with and without the effect of $\mathbf{f}_{h,i}$, respectively (see Figure 3.2). $n_f = \{0, \dots, f \leq 2\}$ are the number of ground contact points, while n_h are the contact points of the external forces;
- $\mathbf{J}_{C_{P_i}}$, $\mathbf{J}_{\hat{C}_{P_i}}$, and $\mathbf{J}_{a_{h_j}}$ are the Jacobian matrices that transform the Cartesian forces ($\mathbf{f}_{wt,i}$, $\mathbf{f}_{wo,i}$, and $\mathbf{f}_{h,i}$ respectively) into generalized forces at the joints level.

To evaluate the cumulative effect of the overloading torque induced on the joints over time by payloads, Lorenzini et al. (2019) introduce a fatigue model based on the joint torque capacity and the overloading effect on the given joints, also considering an individual's subjective perception of fatigue. The fatigue behavior is modelled as an RC circuit with zero initial charge state. The overloading joint fatigue of the i-th joint V_i at time instant t is defined as:

$$V_i(t) = V_{i,max}\left(1 - e^{-\int K_i \frac{\Delta\tau_i(t)}{V_{i,max}} dt}\right) \tag{3.13}$$

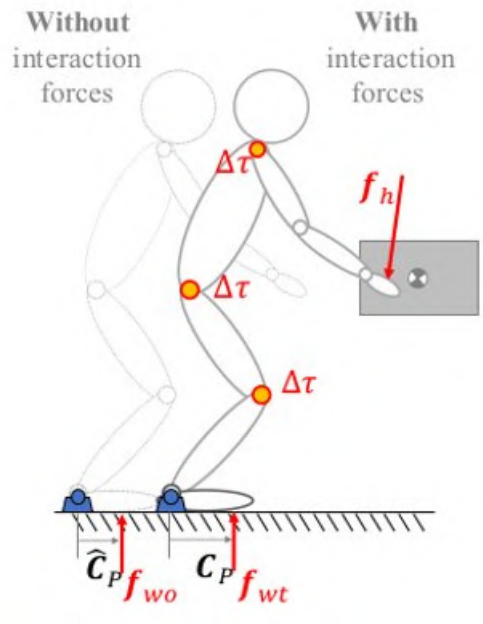


Figure 3.2 COP and vertical GRF variations without and with interaction forces. At the joint level $\Delta\tau$ describes the overloading torque (Kim et al. (2017b))

where $V_{i,max}$ is the maximum joint overloading set according to biomechanical data and K_i is the fatigue ratio and takes into account the subject's physical capacity and feelings. Besides the fatigue model is presented another model to describe the recovery of force generation capacity during the rest periods:

$$V_i(t) = V_{i,max} - (V_{i,max} - V_i(0))e^{-R_i(t)} \tag{3.14}$$

with $V_i(0)$ is the initial overloading fatigue and R_i the recovery ratio for the i-th joint. Then, by assuming $V_{i,th}$ as an overloading threshold, the relation between the two models follows:

$$V_i(t) = \begin{cases} \text{Fatigue model} & \text{if } \Delta\tau_i > V_{i,th} \\ \text{Recovery model} & \text{if } \Delta\tau_i \leq V_{i,th} \end{cases} \tag{3.15}$$

The aim of the overloading joint fatigue/recovery model is that of giving an estimation of the risk associated to task in real-time, and suggesting how to reduce such a risk. In Lorenzini et al. (2019) work the overloading joint fatigue model is used to monitor the joints state and

to trigger the robot assistance once the overloading overcomes the predefined $V_{i,th}$: the robot reacts leading the human to assume a body configuration that mitigates the risk.

Lamon et al. (2019), instead, exploit a simplified version of this fatigue model to obtain "*a metric that is able to estimate some features of the dynamic behaviour of the agent in executing each action*". By estimating the fatigue level at each joint, it is possible to define the agent effort associated with each action in the task. Such a metric is valid for both human and robot agents: the joint torque results from muscle activity and motors, respectively.

Since a way for computing the overloading joint torque online is given (Kim et al. (2017b)), as well as the definition of a dynamic index able to track the overloading joint history by considering the cumulative effects of handling payloads over time (Lorenzini et al. (2019)), and since such information can be embedded in a numeric value describing the agent effort level associated to the execution of each action (Lamon et al. (2019)), the overloading joint fatigue seems suitable to be used as ergonomic risk indicator also in the framework this thesis introduced.

A cognitive risk indicator On the other hand, if the collaborative task requires a demanding mental performance, other factors are necessary to well describe the extra cognitive load risk. The framework presented by Lagomarsino et al. (2021) for the evaluation of excessive cognitive load in assembly workers analyses:

- the gaze direction and head orientation to estimate the level of attention;
- the body language activities, as self-touching, to evaluate the stress-level;
- the cognitive effort envisioned to understand the instructions and to handle the right tools and components for the task completion.

In Table 3.1 the mental effort factors and stress level factors that influence the cognitive load of a human worker are listed and the corresponding formulas for computing each contribution are reported. Combining such indexes together by means of a weighted sum, a final score is obtained that can reflect the human cognitive processing.

[1] Lagomarsino et al. (2021) define the movements s_k^j of j-th joint (j=$[1,2,\dots,N]$ where N is the number of joints) in a time window τ as the sum of the 3D position displacements $d_{k-i,k-i-1}^j$ within two subsequent frames (where k refers to a system pipeline loop). In an initial calibration phase, the mean motion $\mu_1,\dots,\mu_N$ of upper body joints and their standard deviation $\sigma_1,\dots,\sigma_N$ is computed. During task execution, the deviation Δ_k^j of each joint from the baseline μ_j is periodically computed and a parameter a_k^j is associated by comparison with the stored σ_j. A unique descriptor of activity level a_k is determined as the mean of all the upper body joints' activity.

Table 3.1 Definition of *mental effort* and *stress level* factors by Lagomarsino et al. (2021).

Mental effort factors	**Expression**
Concentration Loss: time that the subject does not explicitly dedicate to the task accomplishment.	$1 - \sum_{w=1}^{M} \frac{[\text{attention time}]_w}{\text{time elapsed}}$
Learning Delay: ability to rapidly learning a novel rule from instructions and automaticity in completing assemblies.	$\frac{\text{dwell time on assembly}}{\text{time elapsed}} = \frac{[\text{attention time}]_1}{\text{time elapsed}}$
Concentration Demand: estimation of the incidence of attention failures.	$\sum_{d=1}^{D} \frac{[\text{instant of attention loss}]_d}{\text{time elapsed}}$
Instruction Cost: estimation of the general quality of instructions.	$\sum_{c=1}^{C} \frac{[\text{instant of not required switch}]_c}{\text{time elapsed}}$
Task Difficulty: estimation of the required cognitive effort to perform tasks.	$\sum_{b=1}^{B} \frac{[\text{instant of instruction check back}]_b}{\text{time elapsed}}$
Collaboration Cost: attention that the subject gives to the assistant during the collaborative assembly task.	$\frac{\text{assistant fixation time}}{\text{time elapsed}} = \frac{[\text{attention time}]_3}{\text{time elapsed}}$
Mistrust for assistant: level of trust of the subject toward the assistant.	$\sum_{a=1}^{A} \frac{[\text{instant of assistant check}]_b}{\text{time elapsed}}$
Stress level factors	**Expression**
Self-touching: behavioural indicator of stress and anxiety.	$\sum_{s=1}^{S} \frac{[\text{instant of self-touching}]_s + 60 - t}{60}$
Hyperactivity[1]: high activity periods with respect to baseline movements in terms of joint's displacement over time.	$s_k^j = \sum_{l=0}^{\tau-1} d_{k-l,k-l-1}^j$ if $\Delta_k^j = s_k^j - \mu_j > \sigma_j$ then $a_k^j = \frac{\Delta_k^j}{\sigma_j} - 1$ $a_k = \min\left(\frac{1}{N}\sum_{j=0}^{N} a_k^j, 1.0\right)$

Note: M is the number of workstations, while D, C, B, A, and S are the total occurrences of the corresponding event while working on the task.

By using such scores it is possible both to characterize each assembly action in terms of required mental activity and to assess the cognitive human status during the task execution. Therefore, also this risk indicator can be compatible with our framework.

3.3 Kinematic Wear

In this work, since assembly tasks of a large number of lightweight pieces are addressed, a kinematic index can capture more precisely potentially damaging situations: the manipulated objects are not so much heavy to consider the interaction forces as a main contribute in the ergonomic risk computation. Therefore, we designed a kinematic risk indicator to show potentialities and limitations of such a framework.

The idea behind the *kinematic wear* design consists in having an index able not only to describe the current risk by measuring the current body configuration but also to convey information about the past postures hazard, unlike the traditional kinematic indexes. Therefore, to model the *kinematic wear* index at the joint level we chose an RC circuit behavior:

$$V_i(t) = 1 - (1 - V_i(t_0))\, e^{-\int_0^t \frac{G_i(\mathbf{q}(\tau))}{C} d\tau} \tag{3.16}$$

where $V_i(t) \in [0,1)$, $V_i(t_0)$, C_i, and $G_i(\mathbf{q}(t))$ are, respectively, the *kinematic wear* level at the current instant, the *kinematic wear* level in the initial conditions, the endurance capacity, and the current risk score of the *i-th* upper body joint evaluated according to its configuration $\mathbf{q} = [q_x \;\; q_y \;\; q_z]$.

According to the cooperation protocol, we impose that while the robot executes an action, the following action cannot be allocated to the human: during this rest period the *kinematic wear* level of each joint decreases according to the recovery function (RC circuit discharge):

$$V_i(t) = V_i(t_0)\, e^{-\frac{r}{C}t} \tag{3.17}$$

where r is the recovery rate. Such model has been used already in literature to describe human muscle usage (Lorenzini et al. (2019); Ma et al. (2010)) and also thermal motor usage (Urata et al. (2008)). In a similar way, with Equation 3.16 we would like to provide a joint-level kinematic usage descriptor. Further details about the design of model parameters follow.

Ergonomics Assessment Function G(q) The ergonomics assessment method we chose for evaluating the ergonomics of body configurations is the Rapid Upper Limb Assessment (RULA) (McAtamney and Corlett (1993)). As already mentioned in subsection 1.2.2, this method assigns to each joint (shoulder, elbow, wrist, trunk and neck) a score that represents the associated postural risk: a high score means a high risk. In particular, for each joint, the Range of Motion (RoM) in the sagittal plane (movements mostly occur on this plane) is divided in regions and to each region corresponds a risk level. Then, to each q_s (with this notation we refer to the angle generated by the joint rotation around the axis normal to the sagittal plane) is associated the risk of the region it belongs. With this score assignment criterion, a jump discontinuity happens each time a joint changes region of RoM.

Figure 3.3 RULA scores for shoulder postures in sagittal plane

The shoulder RoM split into five areas, each of them with its risk score (see Figure 3.3), is an instance of such "jumps":

$$G_{shoulder}(q_s) = \begin{cases} 1 & \text{if } -20° \leq q_s < 20° \\ 2 & \text{if } q_s > 20° \\ 2 & \text{if } -45° \leq q_s < -20° \\ 3 & \text{if } -90° \leq q_s < -45° \\ 4 & \text{if } q_s \leq -90° \end{cases} \tag{3.18}$$

where a positive angle describes the arm extension, while the negative angle the arm flexion. To smooth the risk score assignment, $G_i(q_s(t))$ is modeled as a weighted sum of sigmoid functions, to approximate the RULA scores, ensuring continuity. In this way, at the "border" between two regions, the risk does not change instantly but gradually. Then we compute $G_i(q_s)$ as:

$$G_i(q_s) = 1 + \sum_{j=1}^{n_{t,i}} \frac{K_{i,j}}{1 + e^{m(q_s + \bar{q}_{s_{i,j}})}} \tag{3.19}$$

where the unitary constant term is to ensure that $G_{i,min} = 1$ is the minimum value of such function, $n_{t,i}$ is the number of transitions between adjacent areas with different risk score, K_i is the sigmoid gain, $\bar{q}_{s_{i,j}}$ is the angle corresponding to the j-th transition, while m is the slope coefficient: the more $|m|$ increases, the more the resulting function will match the staircase RULA function. In the shoulder case the risk score function, hence, results to be:

$$G_{shoulder}(q_s) = 1 + \frac{1}{1 + e^{-m(q_s - 20°)}} + \frac{1}{1 + e^{m(q_s + 20°)}} + \frac{1}{1 + e^{m(q_s + 45°)}} + \frac{1}{1 + e^{m(q_s + 90°)}} \tag{3.20}$$

As previously specified, the imposed RULA constraints evaluate the antero-posterior movements (shown in Figure 3.3 and Figure 3.4), but the method does not provide any specific risk distribution to distinguish ergonomic joint angles from unsafe ones. Since the human ergonomics assessment cannot be efficient if it refers to a subset of movements, to

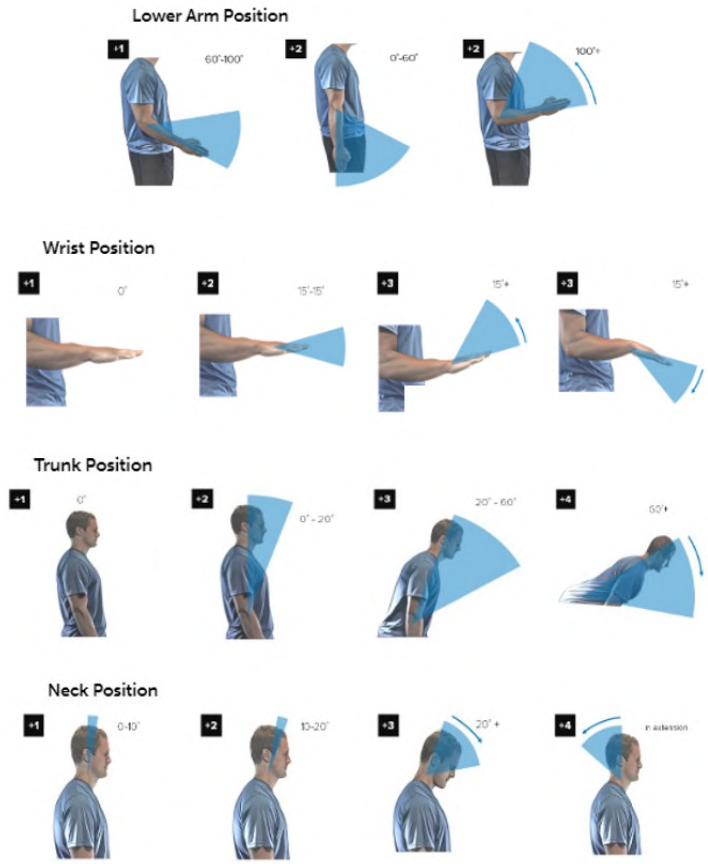


Figure 3.4 RULA scores for postures in sagittal plane of elbow (top), wrist (second), trunk (third), neck (bottom)

have a risk evaluation of postures in 3D, we had to set these additional constraints, which can grade the risk attached to: arm adduction-abduction, external-internal rotation of the arm, wrist side bending and wrist twist, trunk side bending and trunk twist, neck side tilt and neck twist. The criteria we used for setting these constraints consist in adding to the retrieved $G_i(q_s)$ an additional point when the i-th joint overcomes the 75% of its RoM, two additional points when it approaches the RoM limit. Thus, for sake of clarity:

$$G_i(\mathbf{q}) = \begin{cases} G_i(q_s)+1 & \text{if } |q_a| > \frac{3}{4}|q_{a,max}| \\ G_i(q_s)+2 & \text{if } |q_a| \approx |q_{a,max}| \end{cases} \tag{3.21}$$

where q_a refers to the joint angles describing the "additional" movements and $q_{a,max}$ is the joint limit.

Endurance Capacity and Recovery Rate Given that a subject, in a static configuration, can exert a low force (e.g., holding a lightweight object) for $T_{max} = 240\ s$ without physical discomfort (McAtamney and Corlett (1993)), we set the capacity C to allow the human joint to reach the asymptotic value of $V = 1$ after T_{max}, starting from 0 initial condition.

To retrieve the value of C, Equation 3.16 is inverted with a value of $V_i(t) = 0.993 = V_{max}$ (corresponding to five time constants), with an average level of RULA (risk value $G_{avg} = 3$). The score $G_{avg} = 3$ represents a medium joint risk since the values of $G_i(\mathbf{q}(t))$ are bounded by 1 to 7. In this way, the capacity value is the same for all the joints.

$$C = -G_{avg}\,\frac{T_{max}}{\ln(1 - V_{max})} = -3\,\frac{240}{\ln(0.007)}. \tag{3.22}$$

In practice, according to Equation 3.22, the *i-th* joint motion during the task execution generates a change of the RULA score $G_i(\mathbf{q}(t))$, which, in turn, increases the corresponding *kinematic wear* $V_i(t)$, with a slope proportional to the risk level of the new posture: the riskier the posture the steeper the *kinematic wear* increase (see Figure 3.5).

For what concerning the recovery rate r, it is obtained by inverting Equation 3.17 considering to match the recovery time (discharge) with the wear time (charge), i.e. the joint can fully recover in a period of 240 s. Therefore, we have:

$$r = -\frac{C}{T_{max}}\,\ln\left(\frac{1 - V_{max}}{V_{max}}\right), \tag{3.23}$$

It is important to highlight that both C and r values have been decided according to some assumptions based on data found in literature. However, they could be object of an optimization process.

Prediction Model Parameters The model described in section 3.1 can also apply to the *kinematic wear* prediction. A one-step predictor is used to predict the future risk level at the end of each action, in order to decide the next allocation. The prediction $\hat{V}_i(t_{k+1})$ of the risk associated to the execution of the next action a_{k+1}, where t_k is the instant when a_k was completed, can be computed as:

$$\hat{V}_i(t_{k+1}) = \alpha_{k+1,i} V_i(t_k) + \beta_{k+1,i} \tag{3.24}$$

where $\alpha_{k+1,i}$ and $\beta_{k+1,i}$ are the linear parameters that regulate the raise of *kinematic wear* over time, for the i-th joint. From Equation 3.24 is clear that:

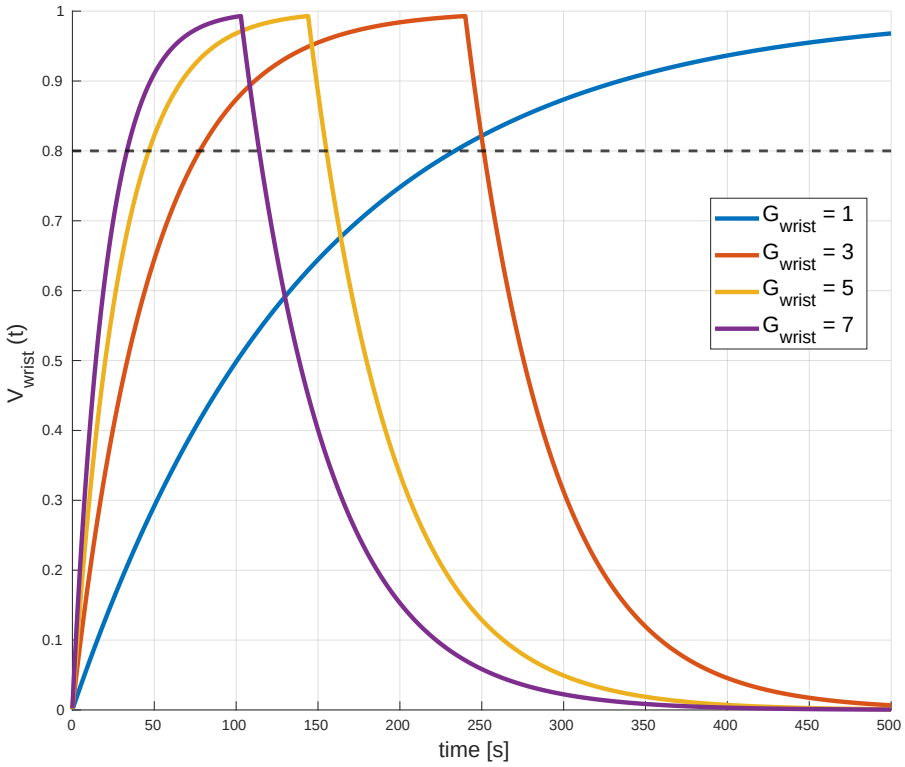


Figure 3.5 Charge and discharge trend of wrist *kinematic wear* $V_{wrist}(t)$ if joint angles remain in the same risk area ($G = 1$, $G = 3$, $G = 5$, and $G = 7$). The higher the risk (greater G), the faster the charge and discharge time.

(i) the predicted risk value is linear with respect to the initial conditions. i.e. the wear accumulated by the previous actions,

(ii) the increase due to the execution of action a_{k+1}, i.e. the model parameters, does not depend on the past actions.

Therefore, the predicted value $\hat{V}_i(t_{k+1})$ is the result of combining information only related to the current human status, embedded by $V(t_k)$, and data about the "future" conveyed by the prediction parameters.

Moreover, $\alpha_{k,i}$ and $\beta_{k,i}$ are representative of the level of the i-th joint involvement in the action a_k and hence should be estimated offline. In Chapter 5, we will show how we estimate these parameters by performing a calibration phase before starting with the effective HRC.

By comparing model (3.16) with (3.24), it is straightforward to understand that the RC circuit-like model provides also a good prediction of the *kinematic wear*, since $\alpha_{k+1,i} = e^{-\int_0^{t_{k+1}} \frac{G_i(\tau)}{C} d\tau}$, and $\beta_{k+1,i} = 1 - e^{-\int_0^{t_{k+1}} \frac{G_i(\tau)}{C} d\tau} = 1 - \alpha_{k+1,i}$. As a result, only one value per joint i per action $k+1$ (i.e. $\alpha_{k+1,i}$) should be estimated and stored to compute $\hat{V}_i(t_{k+1})$ online.

Chapter 4

Software Architecture and Performances

Now that all the framework blocks have been described, the aim of this chapter is that of explaining how the system has been implemented as a software architecture. In particular, section 4.1 shows the main processes and how they communicate among each other, while in section 4.2 the computational complexity in terms of time is discussed in particular for the "bottleneck" component.

4.1 Software Architecture

The architecture has been developed in C++ and Python, on Ubuntu 18.04 and ROS Melodic; the three main processes are defined as ROS nodes, that communicate according to the publisher-subscriber protocol by means of ROS topics. The ROS graph generated by *rqt_graph* representing the overall structure is shown in Figure 4.1.

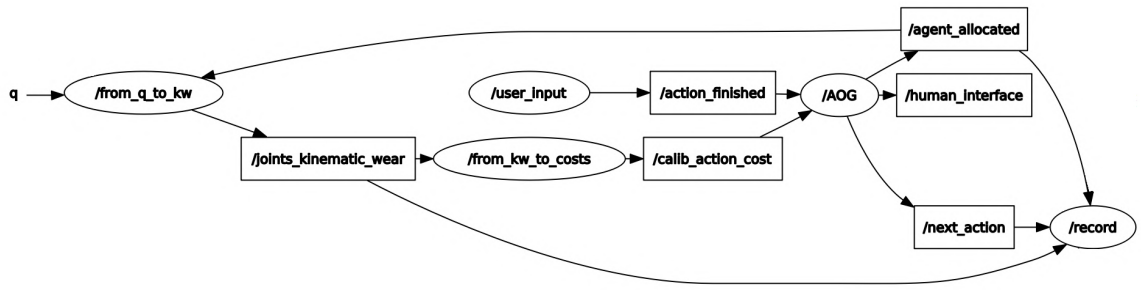


Figure 4.1 Software Architecture of the designed framework. The circles represent ROS nodes, while the squares are ROS topics. The arrows describe the data flow.

From joint angles to Kinematic Wear (from_q_to_kw)
input:

- $\mathbf{q}_i = [q_{i,x} \;\; q_{i,y} \;\; q_{i,z}]$ for each i-th human joint
- w_k, worker allocated for the current action

output:

- $V_i(t)$ for each i-th joint

This node is responsible of converting each upper body joint configuration $\mathbf{q_i}$ into the corresponding risk level $G_i(\mathbf{q})$ and then it updates $V_i(t)$ that keeps track of the entire joint wear history. The second input signal for this node w_k encodes which agent has been allocated for the current action, as the AO* search result. If $w_k =$ human, it means an increase of *kinematic wear*, while if $w_k =$ robot it means that the robot performs the action allowing the human worker to have a rest:

$$V_i(t) = \begin{cases} \text{Fatigue model} & \text{if } w_k = \text{human} \\ \text{Recovery model} & \text{if } w_k = \text{robot} \end{cases} \tag{4.1}$$

This node runs at 20 Hz: this choice is due to the fact that, as shown by Antonsson and Mann (1985), human activity frequencies are between 0 and 20 Hz and the 98% of the FFT amplitude is contained below 10 Hz. Then, according to the Nyquist-Shannon theorem, we can choose 20 Hz as sampling frequency without losing the main joint configuration variations.

From Kinematic Wear to Action costs (from_kw_to_cost)
input:

- $V_i(t)$ for each i-th joint

output:

- c_{a,w_h} for each human action a in the task

This node is subscribed to the *joint_kinematic_wear* topic, reading the *kinematic wear* values with a frequency of 70 Hz, not to lose any message from the publisher. In this way, the process, starting from $V_i(t_k)$ (value of V_i at the end of action a_k), can predict $V_i(t_{k+1})$ (value of V_i at the end of action a_{k+1}) for each possible future action a_{k+1}, by applying the a-priori

knowledge about the joint wear each of them provides. Such a knowledge is collected during a calibration phase, as shown in section 5.2, in which the human worker is asked to perform in a stand-alone modality each action. This phase anticipates the effective HRC.

Then, costs for each human action $c_{a_{k+1,w_h}}$ are computed according to the procedure discussed in section 3.1 and sent to the *AOG* node through a topic, in order to update the corresponding hyper-arcs cost.

AND/OR Graph (AOG)
input:

- c_{a,w_h} for each human action a in the task
- signal encoding the end of action executed by the human

output:

- next action to execute
- worker allocated to that action
- message for the human interface

If action a_k is allocated to the human, when that action performance is finished (at instant t_k), an exogenous signal triggers the hyper-arcs cost updating procedure and the AO* search on a reduced AOG aware that a_k has been just performed (as explained in section 2.3). If instead, a_k is allocated to the robot, the signal that activates the role allocation algorithm corresponds to the robot confirmation of having terminated the action.

The allocation results are published on two different topics and recorded in a log file. While, on the *human_interface* topic a message that informs the human about the AO* solution is sent.

4.2 Computational Complexity Evaluation

According to the asymptotic complexity analysis, if we consider to limit the monitored joints to the upper-body ones, process *from_q_to_kw* increases its computational time linearly with the number of human workers involved in the collaboration N_w. The computational cost, hence, is $O(N_w)$. Similarly, process *from_kw_to_costs* complexity is linearly dependent to both N_w and the number of actions in the task N_a. That is, its cost corresponds to $O(N_w \times N_a)$.

Therefore, since increasing the AOG size causes an exponential growth of the number of possible paths from root to goal state (see section 2.1), to test the time complexity of the whole framework it is sufficient to evaluate the AO* performances. Moreover, since the search is iteratively performed after the execution of each action, this evaluation is important to ensure that a solution can be found within the duration of the current action. We carried out the simulations on a laptop with an Intel Core i7-8565U 1.8 GHz × 8-cores CPU and 8 GB RAM.

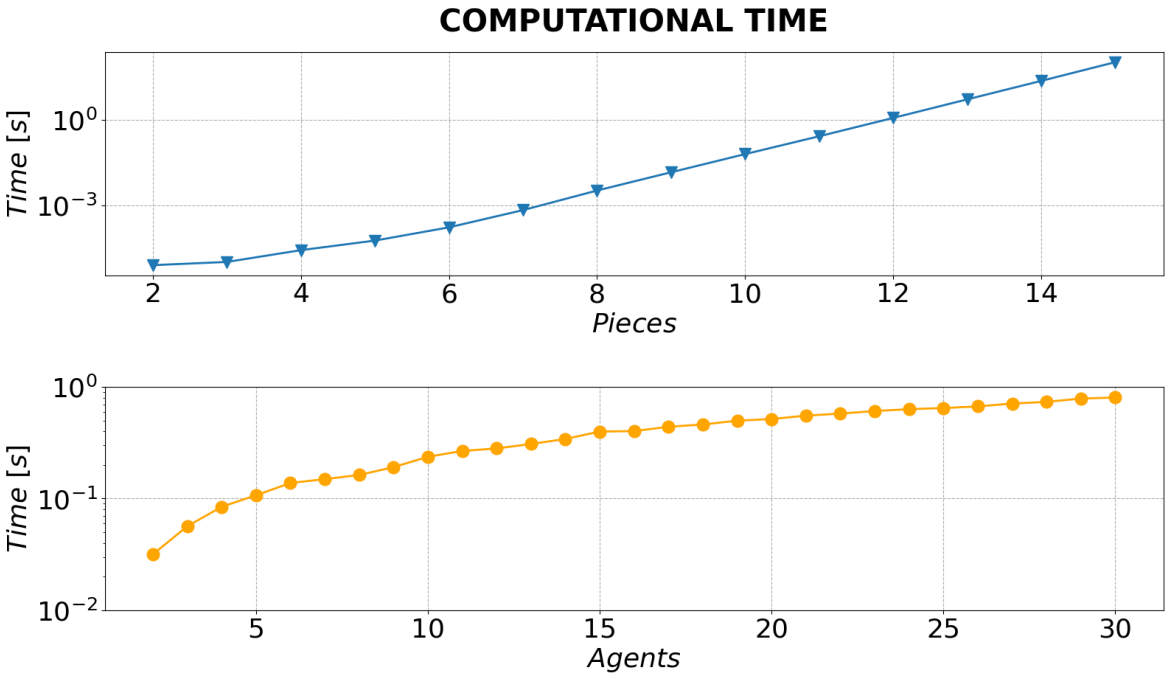


Figure 4.2 AO* computational time (in log scale) obtained by increasing the number of pieces to be assembled (top) and workers (bottom).

The computational complexity of the AO* algorithm was evaluated by running the search on AOGs with an incremental size of:

1. leaf nodes, from 2 to 15, representing the atomic assembly pieces (with a fixed number of agents equal to 2).

2. agents involved in the cooperation, from 2 to 30 (with a fixed number of assembly pieces equal to 10).

These AOGs were generated (via yaml file) by considering the hypothesis of stable and feasible interconnections between adjacent pieces. This means that, given M pieces, there are N-1 interconnections between the M pieces, i.e. the *i-th* interconnection connects piece

p_i and part p_{i+1} (De Mello and Sanderson (1990)). In Figure 4.2 both graphs present an exponential trend, even more promising than the one presented in Darvish et al. (2020) for what concerning the standard representation of AOGs. In particular, the plot in the bottom graph (increasing number of agents) presents a flattened curve. This is because an increment of the number of agents implies a raise of the number of hyper-arcs, while an increment of the number of pieces entails a raise of both nodes and hyper-arcs. An assembly with 15 pieces and 2 agents corresponds, in the worst case, to an AOG with 120 nodes and 1120 hyper-arcs, and the optimal search on such a graph is a considerably large problem. Anyway, the hypothesis that all the interconnections between pieces are feasible is a strong assumption for a general assembly, i.e. not all the adjacent pieces can be connected together. For example, to assemble a table, made of a plate and 4 legs, the legs cannot be assembled together but only with the plate. Moreover, it can be noticed that, after the execution of an action, the search algorithm operates on an AOG with a reduced size, hence the time to generate iteratively the solution is exponentially descending.

Chapter 5

Experiments

To evaluate the performances and investigate the limitations of our framework, we conducted three different experiments. Their common denominator was a proof-of-concept assembly task that we defined purposely, which consisted in the assembly of a corner joint (J) with three aluminum profiles, i.e. two sides (S_1, S_2) and a leg (L). The profiles S_1 and S_2 had the same length, while L was shorter. Each of them interlocked in a predefined hollow of the corner joint J: near each hollow a label with the name of the interlocking profile was pasted (see Figure 5.1).

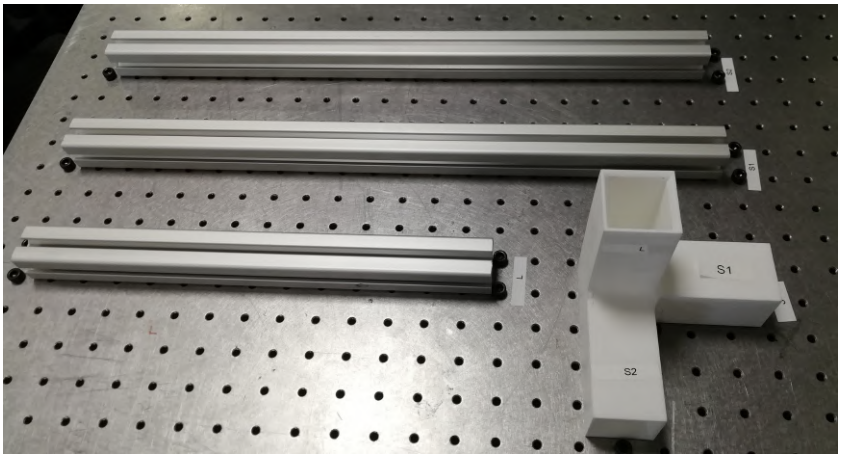


Figure 5.1 The four pieces used for the proof-of-concept assembly task. The three aluminum profiles (L, $S1$, $S2$) and the corner joint (J).

For each experiment, the goal was to complete the simple assembly shown in Figure 5.2, but the experimental setup and the cooperation rules changed to allow a focus on different aspects.

The first two experiments were pilot ones and they were carried out during the development of our framework. They aimed to test:

1. the correct functioning of the **dynamic** role allocation algorithm, without considering the *kinematic wear* values. That is, we wanted to test the software implementation, with regard to the hyper-arcs costs update operation (section 5.1);

2. the *kinematic wear* potentialities in recognizing the riskiest postures and keeping track of the past joint configurations hazard (section 5.2).

These two tests were performed to separately evaluate the functioning of the two main modules of our framework, i.e. the allocation algorithm results employing an AOG that changed size and costs dynamically, and the *kinematic wear* as a tool to describe the joint wear during the actions performance. For this reason, it was considered sufficient to conduct the experiment with a single subject. After performing these two unit testings , by joining the two contributions, we arranged a multi-subject experiment to analyse the functioning of the overall framework and to evaluate its usability in industrial contexts (section 5.3).

Figure 5.2 The complete assembly.

5.1 Pilot Experiment: Dynamic Role Allocation testing

This first pilot experiment was designed to check the correctness of the role allocation method. In particular, we wanted to test the functioning of the hyper-arcs cost updating procedure and the resulting allocation once these costs dynamically changed. To do this, we arbitrarily decided the costs meaning and the updating rules.

Experimental Setup The four assembly pieces (L, $S1$, $S2$, J) were placed on the top of a workbench in a shared workspace between the human subject and a Franka Emika Panda manipulator, which was fixed at the operator's left. The two aluminum profiles S_1 and S_2 were placed, one above the other, in front of the human worker, but on the opposite side of the table, as well as the corner joint J. The profile L, instead, was on the right side of the table. On another table, placed near the robot workspace, a monitor was located, to show to the operator the result of the allocation algorithm. In particular, the message consisted in a sentence: "*n. WORKER - OBJECT*", where n was a progressive number encoding that the action to be executed was the n-th, WORKER could be HUMAN or ROBOT, and OBJECT could be L, $S1$, $S2$, or J. In Figure 5.3, the experimental setup is presented. Such a pilot experiment was conducted with one subject (gender: male, age: 29 years, height: 182 cm).

Costs meaning and updating procedure The atomic actions of the task were: (a_1) pick the corner joint J from its initial position and place it on the workbench closer to the human operator, (a_2) pick and insert S_1 in the upper hollow of J, (a_3) pick and insert S_2 in the hollow of J at the operator's left, (a_4) pick and assemble L in the hollow of J at the operator's right, (a_5) pick the complete assembly and place it on the additional table, further from the human. In this experiment, we fixed the assembly sequence to test the allocation algorithm, namely, the five actions had to be executed in this order: a_1, a_2, a_3, a_4, and a_5. The variability here was implemented by assigning each one of these actions either to the human worker or to the robot, according to the hyper-arcs cost that changed during the performance. The atomic actions are shown in Figure 5.4, Figure 5.5, Figure 5.6, Figure 5.7, and Figure 5.8. For the robot, the atomic actions were defined in the same way. In all our experiments, for the sake of simplicity, the positions of the assembly pieces to be handled by the robot were directly inserted into the robot script.

This first experiment aimed to test the dynamic allocation process, i.e. the correct updating of the hyper-arcs costs, which is the base of a successful role allocation output. Therefore, we arbitrarily assigned the hyper-arcs costs without any reference to a specific risk indicator $V(t)$. However, such costs were determined by roughly evaluating the ergonomics

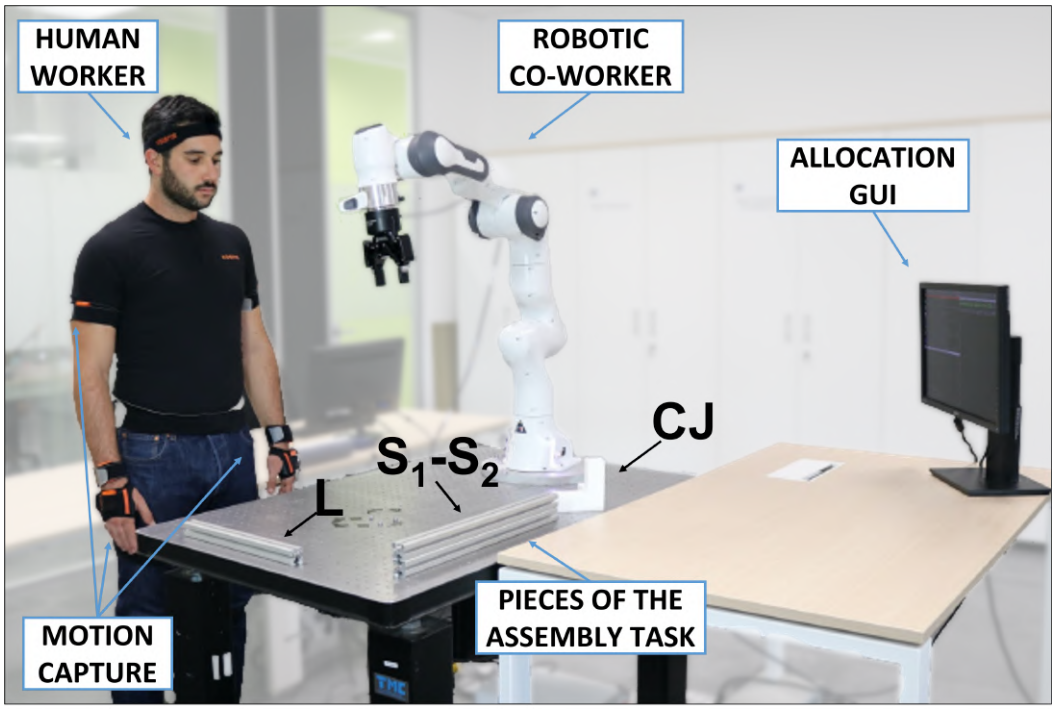


Figure 5.3 Experimental setup. The human worker and the robot co-worker shared the workbench where the assembly pieces were placed. A monitor informs the human worker of allocation results.

of each action performance. We assigned a higher cost to the hyper-arc modelling an action that we retained less ergonomic and vice versa. In Table 5.1, the costs that were assigned at each action performed by the human are listed in column $c^{nominal}_{a,w_h}$, while the cost of robot actions remained fixed at $c_{const} = C_R = 35$ (column c_{a,w_r}).

To handle the costs updating, a variable named *cumulRiskScoreH* was defined. It kept track of the human performance by adding to its initial value the cost of the executed action. Then, at the end of each action a_k, it was possible to know the current accumulated score $cumulRiskScoreH(t_k)$ and also make a prediction on the score obtained in the case the human

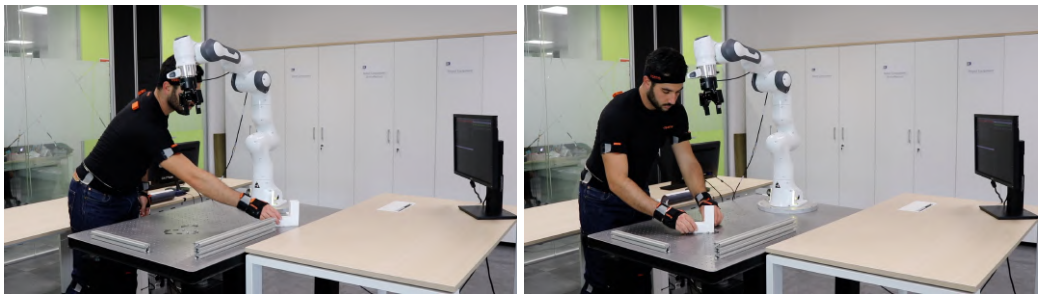


Figure 5.4 Action 1: pick (left) and place (right) the corner joint J

Figure 5.5 Action 2: pick (left) and assemble (right) the aluminum profile S_1

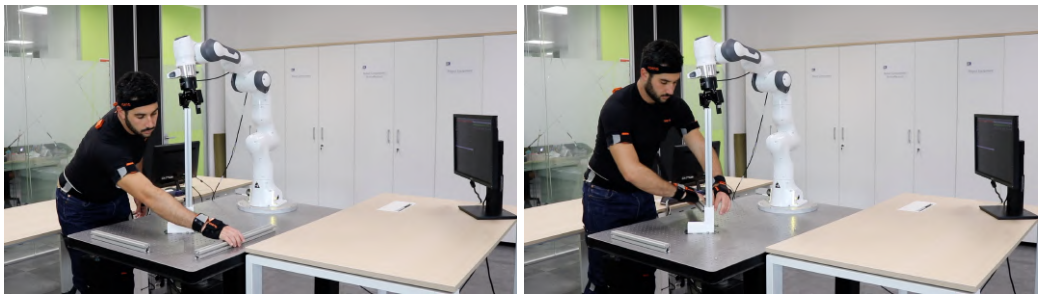


Figure 5.6 Action 3: pick (left) and assemble (right) the aluminum profile S_2

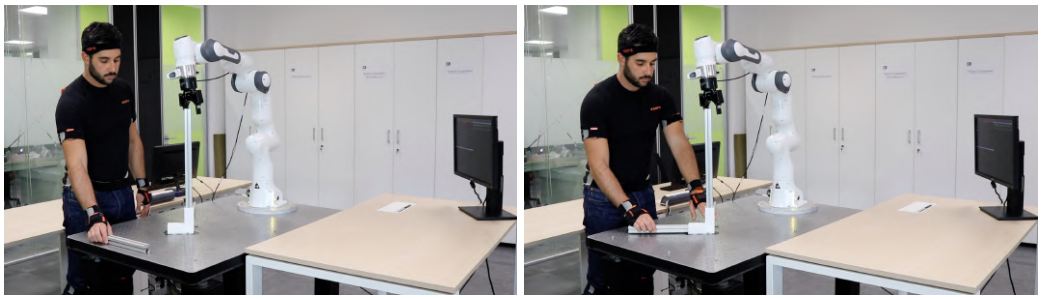


Figure 5.7 Action 4: pick (left) and assemble (right) the aluminum profile L

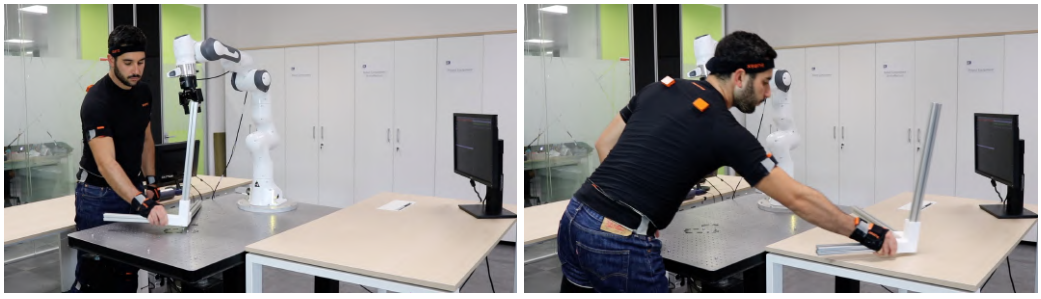


Figure 5.8 Action 5: pick (left) and place (right) the entire assembly

Table 5.1 Action costs for the human worker and robot

action a_k	$c^{nominal}_{a_{k,w_h}}$	$c_{a_{k,w_r}}$
a_1	10	35
a_2	20	35
a_3	15	35
a_4	5	35
a_5	30	35

executed the next action a_{k+1}:

$$cumulRiskScoreH(t_{k+1})^{pred} = cumulRiskScoreH(t_k) + c^{nominal}_{a_{k+1,w_h}}.$$

Such $cumulRiskScoreH(t_{k+1})^{pred}$ became the new cost of the human hyper-arc modeling a_{k+1}:

$$c^{new}_{a_{k+1,w_h}} = cumulRiskScoreH(t_{k+1})^{pred} = cumulRiskScoreH(t_k) + c^{nominal}_{a_{k+1,w_h}}.$$

We expected the allocation to work in this way:

$$\begin{cases} a_{k+1} \text{ allocated to human} & \text{if } c^{new}_{a_{k+1,w_h}} < C_R \\ a_{k+1} \text{ allocated to robot} & \text{if } c^{new}_{a_{k+1,w_h}} \geq C_R \end{cases} \tag{5.1}$$

If action a_{k+1} was assigned to the robot co-worker, then the accumulated cost was reset: $cumulRiskScoreH(t_{k+1}) = 0$.

Experimental Results By considering non zero initial conditions ($cumulRiskScoreH(0) = 20$) the results of the cooperative assembly are shown in Figure 5.9. The first action was allocated to the human, since, due to the non zero initial condition, $c^{new}_{a_{1,w_h}} = 20+10 = 30 < C_R$. The second action, instead, was allocated to the robot, since $cumulRiskScoreH(t_1) = 30$ and adding the nominal cost of action a_2 yielded $c^{new}_{a_{2,w_h}} = 30+20 = 50 > C_R$. The dashed line in red shows the resulting *cumulRiskScoreH* in the case the human would have performed a_2 ($cumulRiskScoreH(t_2)^{pred} = 50$).
Since the robot executed a_2, *cumulRiskScoreH* was reset ($cumulRiskScoreH(t_2) = 0$), then the cost of hyper-arc modeling a_3 was $c^{new}_{a_{3,w_h}} = c^{nominal}_{a_{3,w_h}} = 15 < C_R$ and the algorithm assigned a_3 to the human.

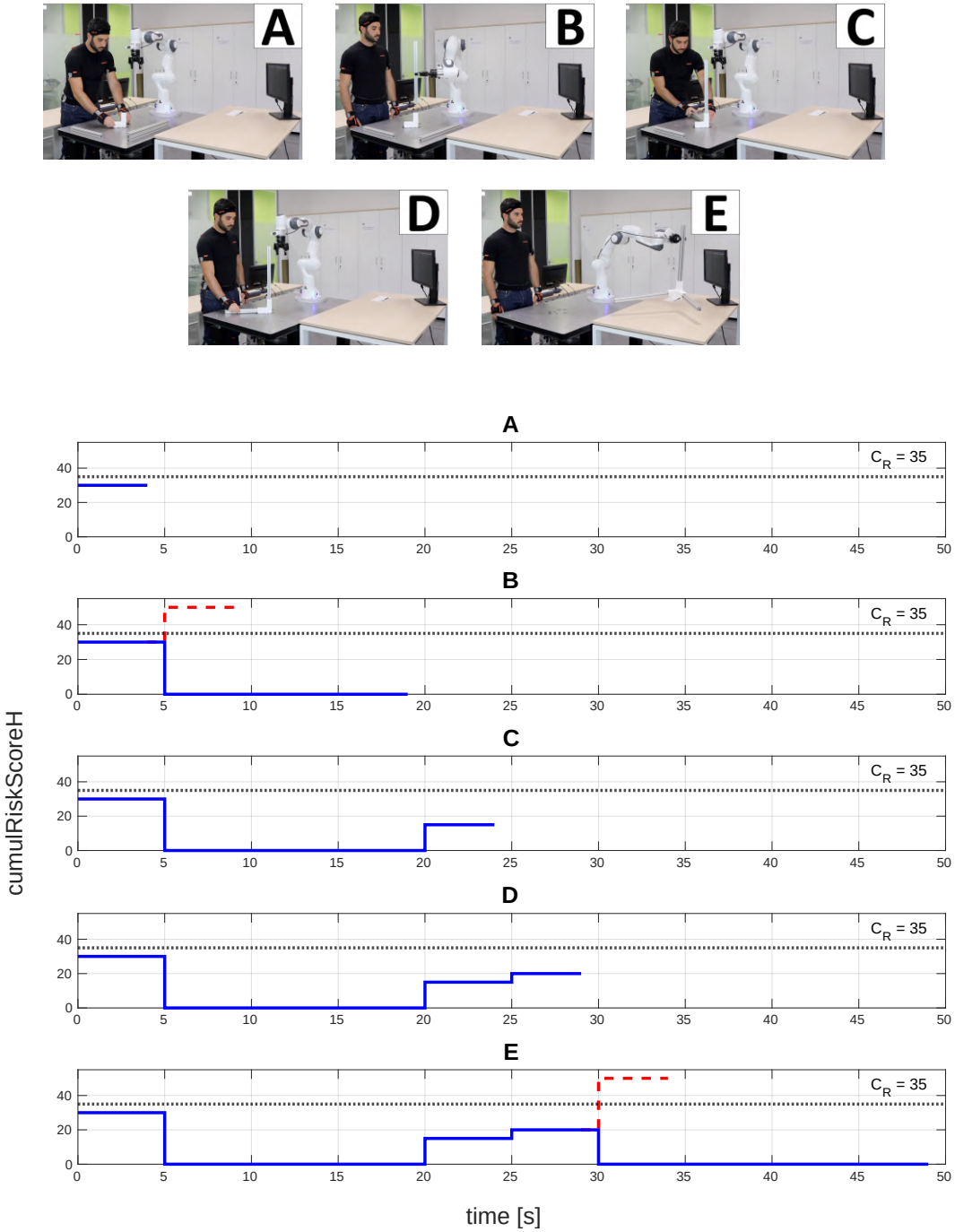


Figure 5.9 (Top) Snapshots of the experiment. (Bottom) *cumulRiskScoreH* value across the five actions (blue), the resulting values in case a damaging action would be assigned to the human (dashed red line), the robot hyper-arcs cost C_R (black) working as a threshold on *cumulRiskScoreH*.

After a_3 we had $cumulRiskScoreH(t_3) = 15$, while $cumulRiskScoreH(t_4)^{pred} = 15+5 = 20$: a_4 was allocated to the human, due to $c^{new}_{a_{4,w_h}} = 20 < C_R$.
The last action, instead, was allocated to the robot: the predicted value $cumulRiskScoreH(t_5)^{pred} = 20+30 = 50$ (dashed red line) changed the cost of hyper-arc modeling a_5 to be $c^{new}_{a_{5,w_h}} = 50 > C_R$. The last allocation reset again the *cumulRiskScoreH* value.

The *cumulRiskScoreH* indicator that we designed for this unit testing, mimed the *kinematic wear* behavior, by increasing when the action was assigned to the human, decreasing otherwise, and being exploited also to predict its future values, the new hyper-arcs costs. Clearly, such a discrete indicator could not module the increasing and decreasing trends according to the monitored joint angles, taking into account how the action were actually performed. Nevertheless, it allowed us to prove the correctness of the software implementation of our framework, concerning the dynamic update of the hyper-arcs cost, a mainstay for a correct allocation result, and of the optimal search AO* that ran on an AOG that dynamically changed size and costs. It should be noted that the subject wore the Xsens suit, an IMU-based suit for the tracking of his joint angles. The reason was to collect some data we used for the *kinematic wear* testing: however, this was not the objective of the just discussed pilot experiment.

5.2 Pilot Experiment: Kinematic Wear testing

In this second pilot experiment, we tested the *kinematic wear* characteristics and its capability to recognize the hazardous body configuration and track the history of joint wear. To do so, a single subject (gender: female, age: 25 years, height: 158 cm) was asked to complete the assembly in represented Figure 5.2 composed by the pieces illustrated in Figure 5.1, as in the previous experiment (section 5.1). In this experiment, however, the initial position of the pieces was set differently. The latter were located in the workspace in a way to force the subject, during the experiment, to assume body configuration with high level of risk according to RULA. For certain actions, a specific body joint was pushed toward the limit of its Range of Motion to the extent possible. The aim was to investigate the resulting trend of the kinematic wear and verify its capability to account for joint usage over time. In Figure 5.10, the experimental setup is shown. The aluminum profiles were placed on the workbench, S_1 and S_2 at the subject's left, while L at her right. Near the workbench there was a shelf 171 cm tall, with J on the highest ledge. As in the previous experiment, when the assembly was completed, it had to be moved from the assembly area. In this case, the final destination was another ledge of the shelf. The subject wore the Xsens suit, to track her joint

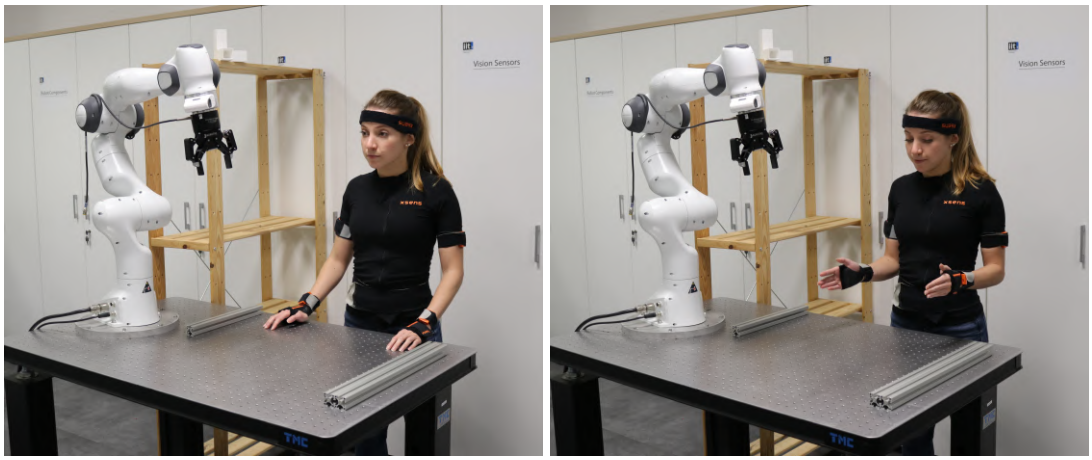


Figure 5.10 Experimental setup. The workstation consisted in a workbench in front of the human operator and in a shelf at the human's right. The corner joint J was on top of the shelf, while the aluminum profiles were on the workbench. The human wore an Xsens suit as motion-capture system. The human operator assumed the minimum risk position (right), i.e. RULA score was about 1 for all the monitored joints.

angles. In this experiment, we monitored shoulder, elbow, and wrist of the dominant arm (the right one: our subject was right-handed), trunk, and neck (as discussed in section 3.3).

In this experiment, the human was asked to perform each assembly action separately from the others, at most with the right arm (the monitored one) and to start and finish, to the extent possible, each execution in the minimum risk posture (shown in Figure 5.10 on the right), i.e. the body configuration in which all the monitored joints had minimum RULA score. It should be noted that it was not always possible for the subjects to reach this minimum score (i.e. for all joints, in all cases). This was most likely due to the Xsens system sentivity to sensor movements during the acquisition, which may affect the data, and to the fact that the subject was not always matching the instructed posture, i.e. the joint angles that corresponded to the minimum risk posture according to RULA. However, the expected results were not affected by this initial condition. The reason behind such experimental protocol consisted in proving that, given any action that required to move all the joints, the *kinematic wear* could give a description of how much and how (i.e. crossed portions of RoM) the single joint was used in that action performance.

Moreover, to have a *kinematic wear*-based description of each action in the task was mandatory to allow the allocation algorithm, which used the *kinematic wear* as the risk indicator, to predict the risk associated to each possible future action, saving the human from hazards (as shown in subsection 5.2.2).

5.2.1 Kinematic Wear-based Action Description

Before starting with the description of each single action in terms of *kinematic wear*, it was necessary to introduce how the monitored quantities were mapped into body segments movements.

From human body generalized coordinates to body segments motions: the adopted conventions To measure the human worker joint angles we used an IMU-based suit (Xsens suit xsens.com), with 17 IMU sensors, that is advantageous for the user readiness and the unlimited field of view of the system; otherwise other motion capture systems could be employed, such as infrared or RGB-D camera systems running skeleton tracking algorithms. In particular, the latter looks promising since it does not require the worker to wear any sensorized suit or marker, but it is prone to occlusions. The Xsens suit captured the human kinematic configurations, in terms of joints relative angles and links positions. The joint values were then processed by the ergonomic risk assessment method, returning the *kinematic wear* index $V_i(t)$ for each joint i.

The recorded angles value was interpreted according to the position and orientation of the corresponding joint frame. As we can see in Figure 5.11, each framework has the origin at the joint level and it is rotated so that:

- x-axis is normal to the screen, pointing outwards,
- y-axis lies on the screen plane, pointing rightwards,
- z-axis lies on the screen plane, pointing upwards.

The red squares highlights the frames we considered in this experiment. From now on q_x, q_y, and q_z refer to rotations around x, y, and z-axis, respectively.

Shoulder For shoulder joint, we monitored the x, y, and z components of $q14$, in Figure 5.11. An increasing value of $q_{sh,x}(t)$ meant the worker was bringing his right arm close to the body (adduction), a decreasing trend of $q_{sh,x}(t)$ described the opposite movement (abduction). An increasing value of $q_{sh,y}(t)$ meant the worker was moving his right arm posteriorly (extension), a decreasing trend of $q_{sh,y}(t)$ described the opposite movement (flexion). An increasing value of $q_{sh,z}(t)$ meant the worker was performing an internal rotation of his right arm, a decreasing trend of $q_{sh,z}(t)$ described the opposite movement (an external rotation towards midline).

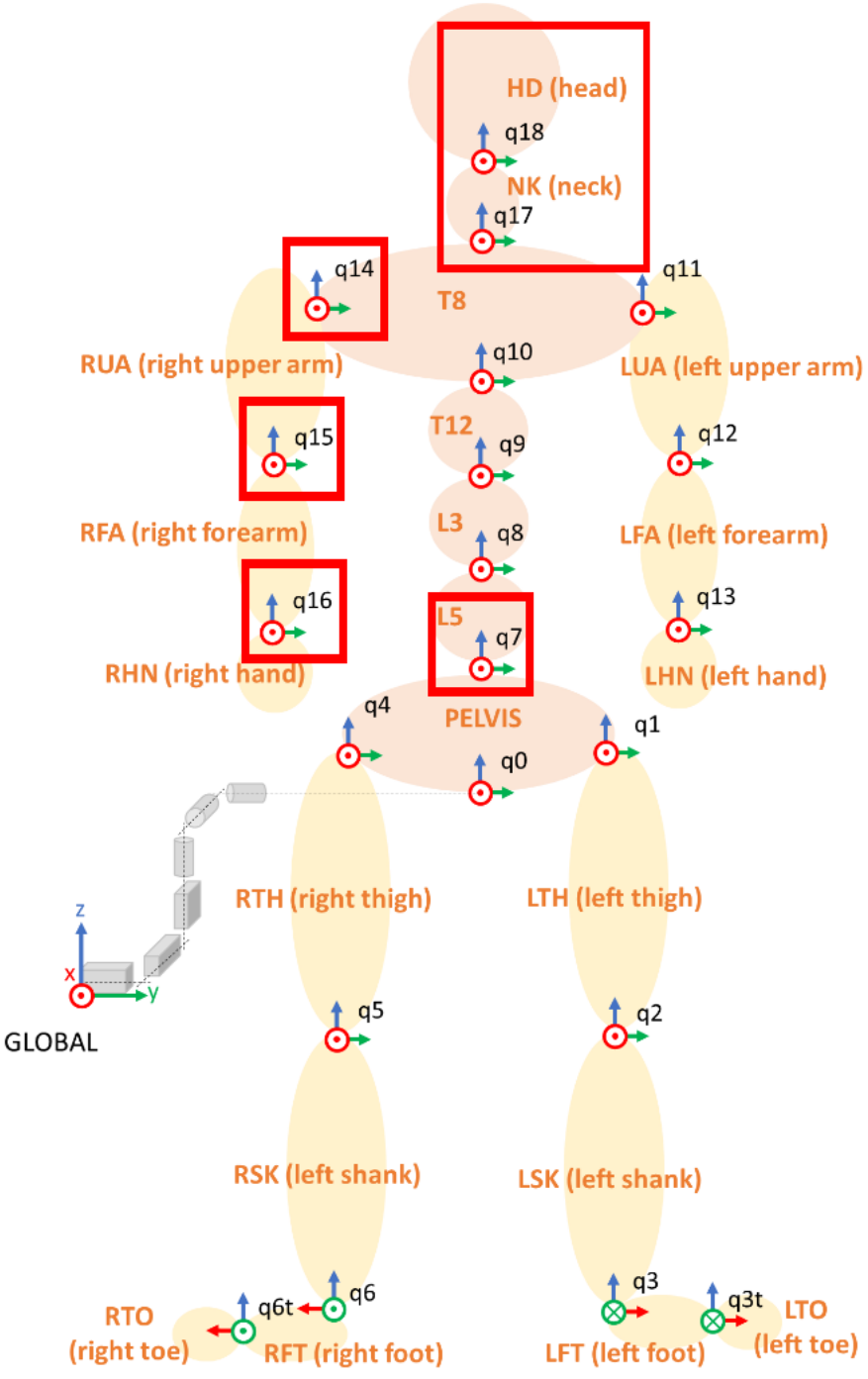


Figure 5.11 X-sens human model. The red squares highlights the considered frames.

Elbow For elbow joint we monitored only the y component of $q15$, in Figure 5.11. A decreasing value of $q_{el,y}(t)$ meant the worker was bringing his right lower-arm up close to the right upper-arm, towards an angle equaling zero between the two segments (flexion), an increasing trend of $q_{el,y}(t)$ described the opposite movement (extension).

Wrist For wrist joint we monitored the x, y, and z components of $q16$, in Figure 5.11. An increasing value of $q_{wr,x}(t)$ meant the worker was bringing his right hand down (flexion), a decreasing trend of $q_{wr,x}(t)$ described the opposite movement (extension).
An increasing value of $q_{wr,y}(t)$ meant the worker was bending the wrist to the little finger (ulnar deviation), a decreasing trend of $q_{wr,y}(t)$ described the opposite movement (radial deviation).
An increasing value of $q_{wr,z}(t)$ meant the worker was bringing his right hand palm pointed towards the floor (pronation), a decreasing trend of $q_{wr,z}(t)$ described the opposite movement (supination).

Trunk For monitoring the trunk movements, the pelvis joint $q7$ (see Figure 5.11) was monitored in its components $q_{tr,x}(t)$, $q_{tr,y}(t)$, and $q_{tr,z}(t)$. An increasing value of $q_{tr,x}(t)$ meant the worker was bending his trunk rightwards, a decreasing trend of $q_{tr,x}(t)$ described the opposite movement (left bending).
An increasing value of $q_{tr,y}(t)$ meant the worker was performing a simultaneous forward pelvic tilt and flexion of the spine (flexion), a decreasing trend of $q_{tr,y}(t)$ described the opposite movement (extension).
An increasing value of $q_{tr,z}(t)$ meant the worker was twisting his trunk leftwards, a decreasing trend of $q_{tr,z}(t)$ described the opposite movement (right twist).

Neck Since the quantities we measured for a specific joint were defined with respect to the "joint father", in this case for monitoring and describing movements that RULA associated to the neck, we obtained the angles that we looked for ($q_{ne,x}(t)$, $q_{ne,y}(t)$, and $q_{ne,z}(t)$) by summing the components of the head with respect to the neck ($q18$) and of the neck with respect to the trunk ($q17$): in this way, we had the q_{ne} describing the movement of the head with respect to the trunk. An increasing value of $q_{ne,x}(t)$ meant the worker was tilting his head rightwards, a decreasing trend of $q_{ne,x}(t)$ described the opposite movement (left tilt).
An increasing value of $q_{ne,y}(t)$ described the worker head flexion, a decreasing trend of $q_{ne,y}(t)$ described the opposite movement (extension).

$q_{i,axis}[rad]$	+	-
$q_{sh,x}$	adduction	abduction
$q_{sh,y}$	extension	flexion
$q_{sh,z}$	internal rotation	external rotation
$q_{el,y}$	extension	flexion
$q_{wr,x}$	flexion	extension
$q_{wr,y}$	ulnar deviation	radial deviation
$q_{wr,z}$	pronation	supination
$q_{tr,x}$	right bending	left bending
$q_{tr,y}$	flexion	extension
$q_{tr,z}$	left twist	right twist
$q_{ne,x}$	right tilt	left tilt
$q_{ne,y}$	flexion	extension
$q_{ne,z}$	left twist	right twist

Table 5.2 From *i*-th joint configuration to the corresponding segment movements. The two columns "+" and "-" indicate the movement described by a positive and negative rotation around the corresponding axis.

An increasing value of $q_{ne,z}(t)$ meant the worker was twisting his head leftwards, a decreasing trend of $q_{ne,z}(t)$ described the opposite movement (right twist).

With in mind the correspondences listed in Table 5.2, we were able to interpret the $q_{i,x}$, $q_{i,y}$, and $q_{i,z}$ evolution during each action performance in terms of human body segment movements.

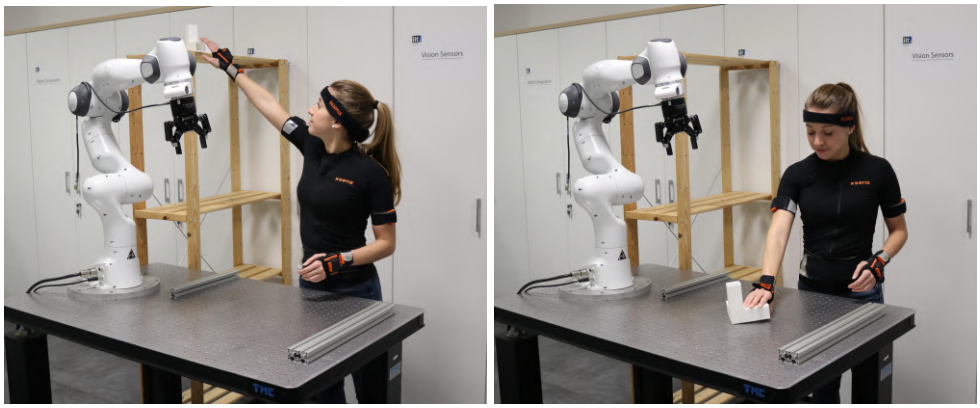


Figure 5.12 Action 1: pick and place *J* on the workbench

Action 1: pick and place *J* on the workbench The human worker was asked to reach the corner joint at the highest ledge of the shelf and to place it in the assembly area.

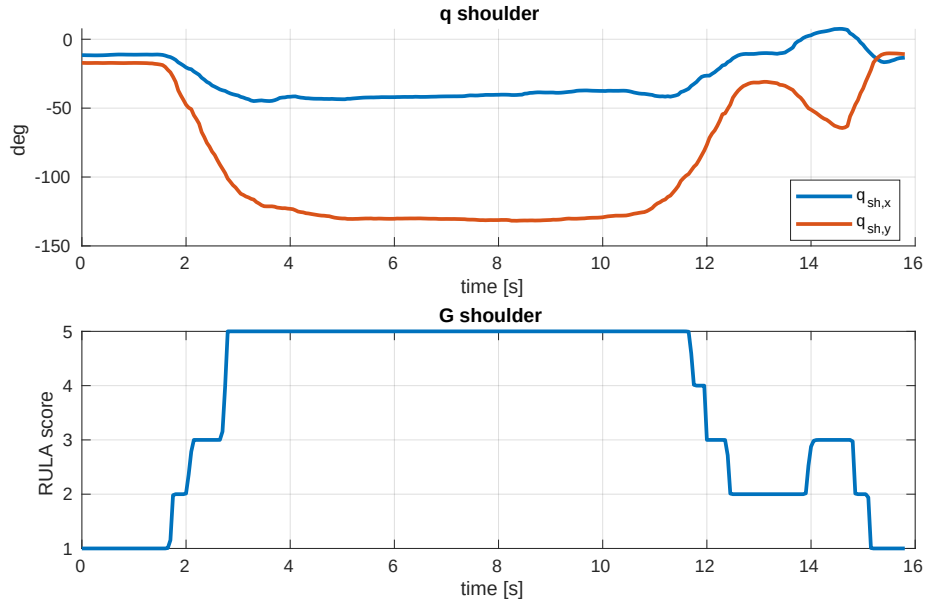


Figure 5.13 Action 1: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant.

In Figure 5.13, the top graph shows $q_{sh,x}$ (in blue) and $q_{sh,y}$ (in red) trends. The decreasing trend of $q_{sh,y}$ means that the human was flexing the right arm to reach the piece on top of the shelf; the plateau corresponds to the grasping phase; then the movement continued in placing the object in the new configuration, closer to the operator, so the arm was not too much far from the body anymore ($q_{sh,y}$ closer to zero). The $q_{sh,x}$ curve shows that for reaching and grasping the object the arm was abducted, returning to the initial posture during the "place phase". The bottom graph, instead, shows the shoulder risk level during the performance of this action, that was maximum during the arm flexion and abduction once J was reached.

In Figure 5.14, the elbow extension movement was described by the increasing trend of $q_{el,y}$ (red line), while the blue line corresponds to the angle the shoulder described rotating around the z-axis, $q_{sh,z}$. These two components were put together due to the risk assigned to the lower arm position according to the RULA tables we used for the risk evaluation (see section 3.3). Such a risk is shown in the bottom graph and the highest score is associated to the elbow extension (rad 0) performed for reaching the object J.

In Figure 5.15, the top graph presents $q_{wr,x}$ (in blue), $q_{wr,y}$ (in red) and $q_{wr,z}$ (in yellow) trends. The $q_{wr,x}$ values show that once the object J was reached, to grasp it the wrist performed a flexion. The $q_{wr,y}$ curve shows an almost constant ulnar deviation that increased in the object placement. The $q_{wr,z}$ curve starts from 0 rad, corresponding to have the hand palm pointing towards body midline (neutral position). Then the pick motion required the

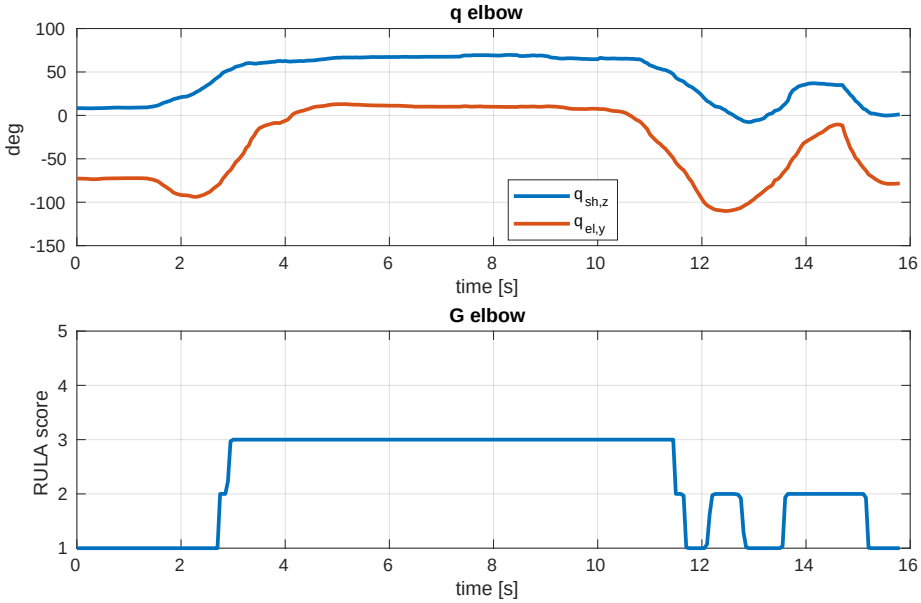


Figure 5.14 Action 1: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant.

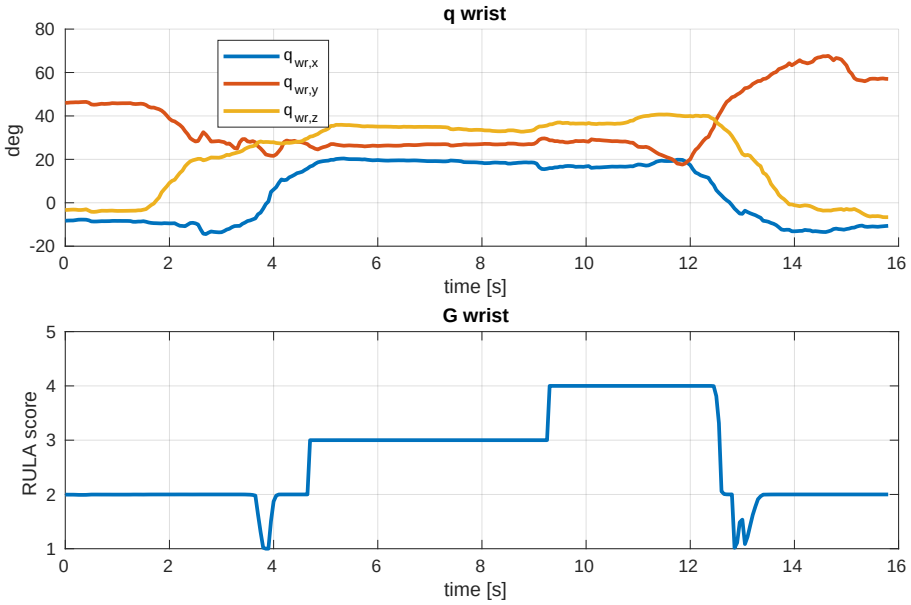


Figure 5.15 Action 1: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant.

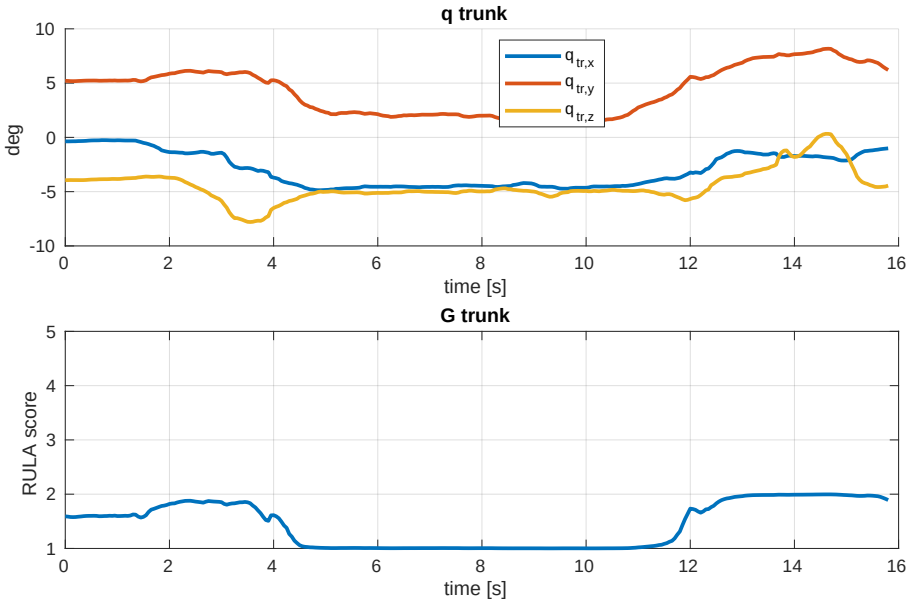


Figure 5.16 Action 1: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant.

hand pronation; the opposite rotation was necessary to restore the initial configuration. In the bottom graph, the risk score shows a higher risk in the grasping phase.

Figure 5.16 describes the trunk angles, that remained almost constant for the whole duration of action a_1, since to reach the object in its position it was not required to bend the trunk. Therefore, the risk associated to the trunk of this action was small, as shown in the bottom graph.

Figure 5.17 shows in the top graph the neck angles, in particular $q_{ne,z}$ (in yellow) described a right twist, and $q_{ne,x}$ (in blue) a right tilt, due to the object J on top of the shelf at the operator's right. While the value of $q_{ne,y}$ described an almost constant neck flexion. The corresponding ergonomic risk is shown in the bottom graph.

Figure 5.18 illustrates the *kinematic wear* trend for each of the monitored joints. According to Equation 3.16, the increase of V_i is proportional to the risk accumulated during the performance of a_1, represented by the term $\frac{1}{C}\int_0^{t_1} G_i(\tau)d\tau$ (shown in the bottom graph). As we expected, since to reach the object the human needed to stretch the right arm and to rise it almost approaching the RoM limit, the shoulder was the joint most at risk. On the other hand, the trunk was the less stressed joint, since any trunk bending was not necessary to accomplish the a_1 goal.

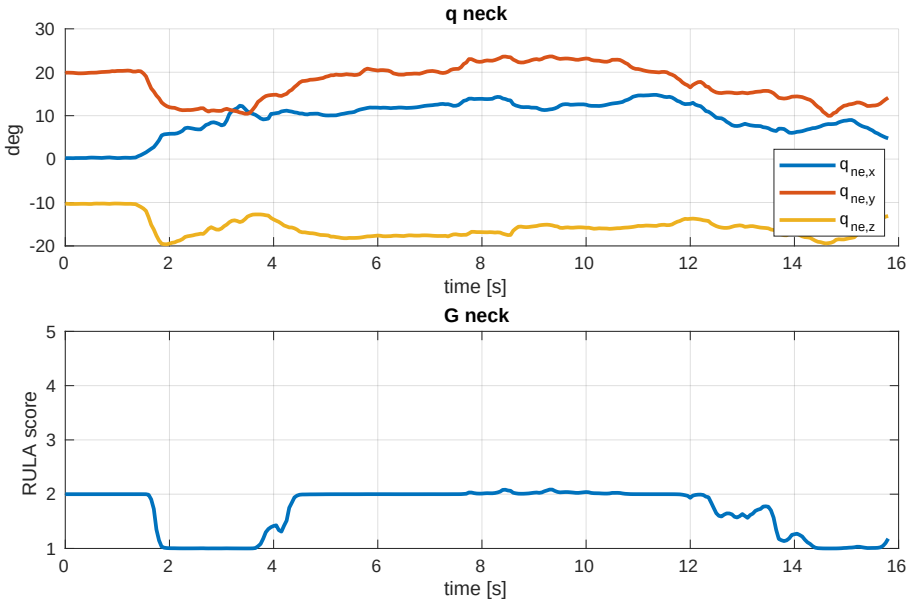


Figure 5.17 Action 1: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant.

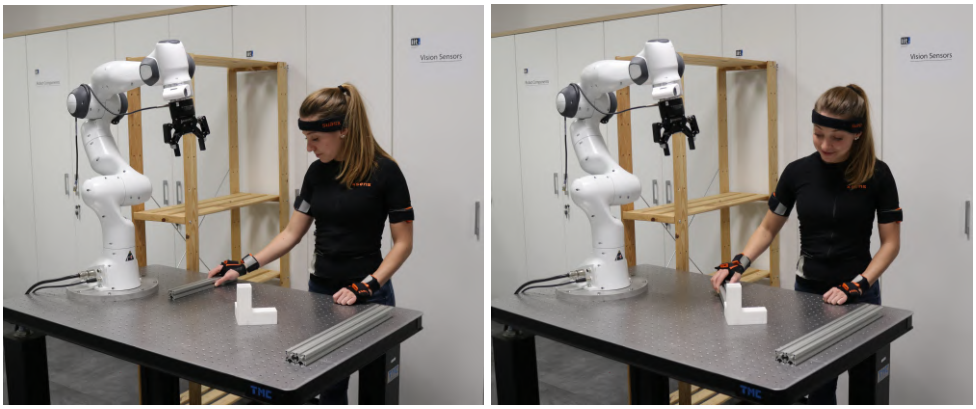


Figure 5.19 Action 2: pick and assemble the aluminum profile L

Action 2: pick and assemble the aluminum profile L The subject was asked to grasp the profile L at her right and to assemble it in the assembly area. Since the object was really close to the human right hand, as well as the corresponding hole where L interlocks, this action required a low effort, little movements for all the joints, except for the neck: the human had to perform the neck flexion and a rightward twist, in order to allow the gaze pointing the object (Figure 5.24). Figure 5.25 shows, hence, the neck as the most at risk joint for the execution of a_2.

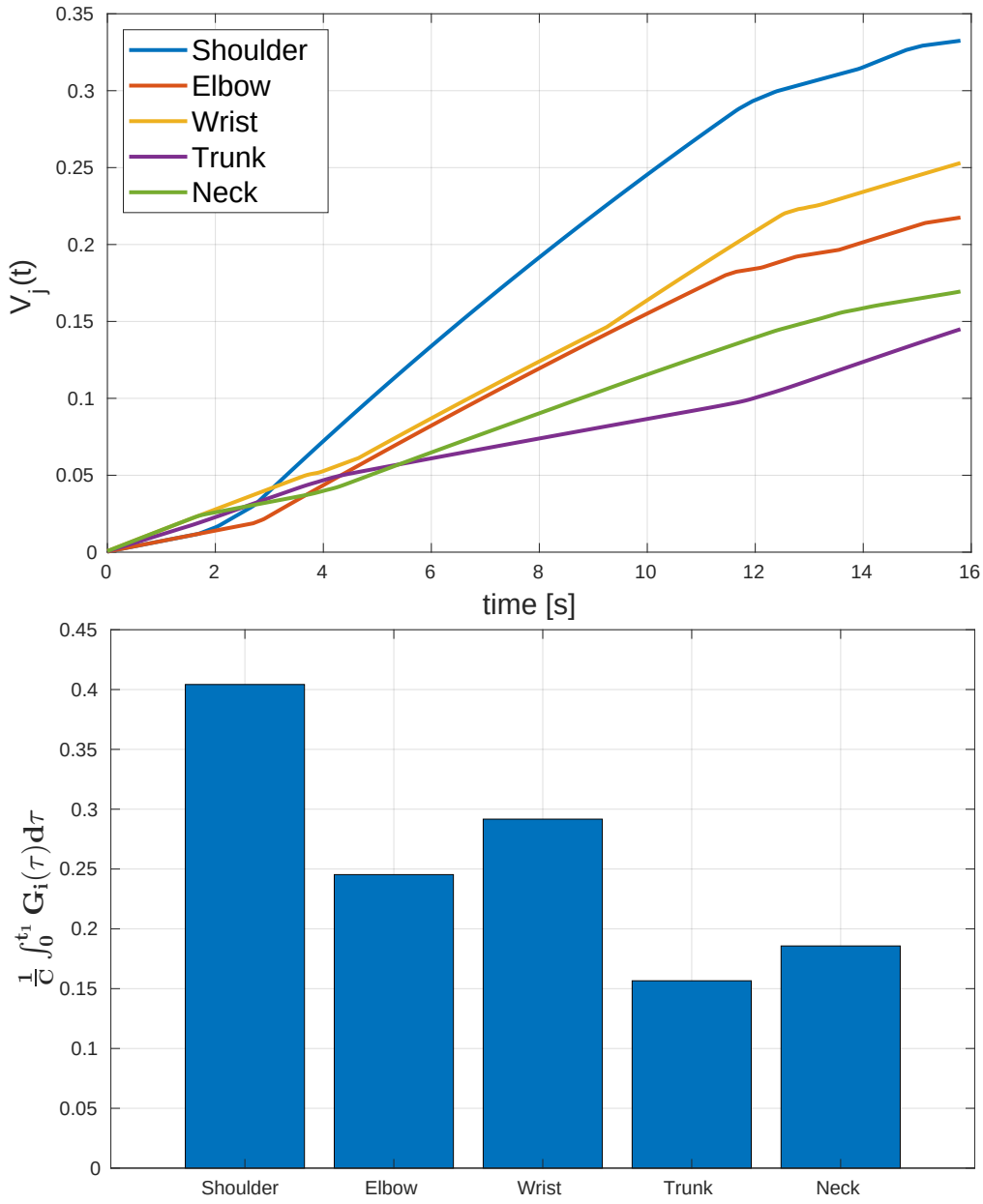


Figure 5.18 The *kinematic wear* trend for all the monitored joints during the performance of a_1, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_1 (bottom).

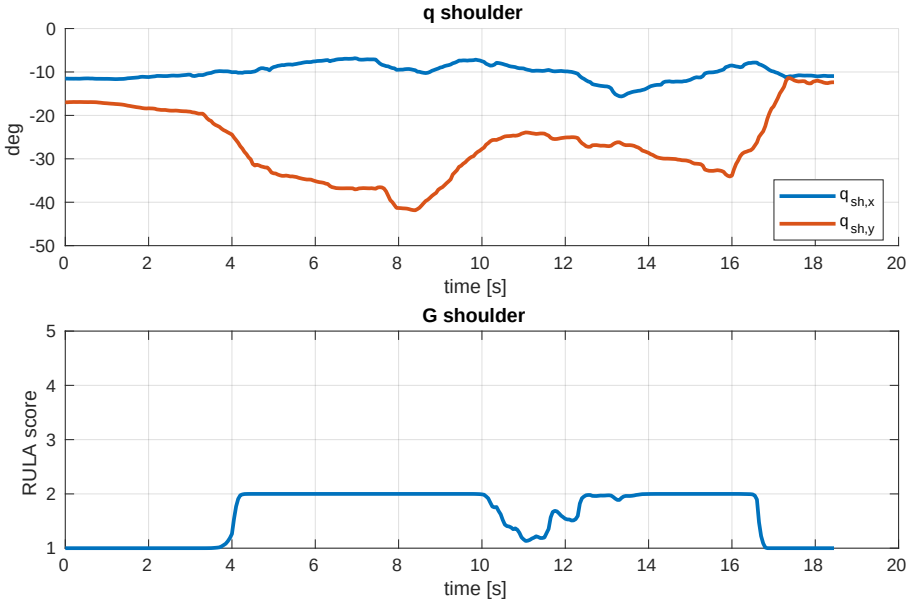


Figure 5.20 Action 2: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant.

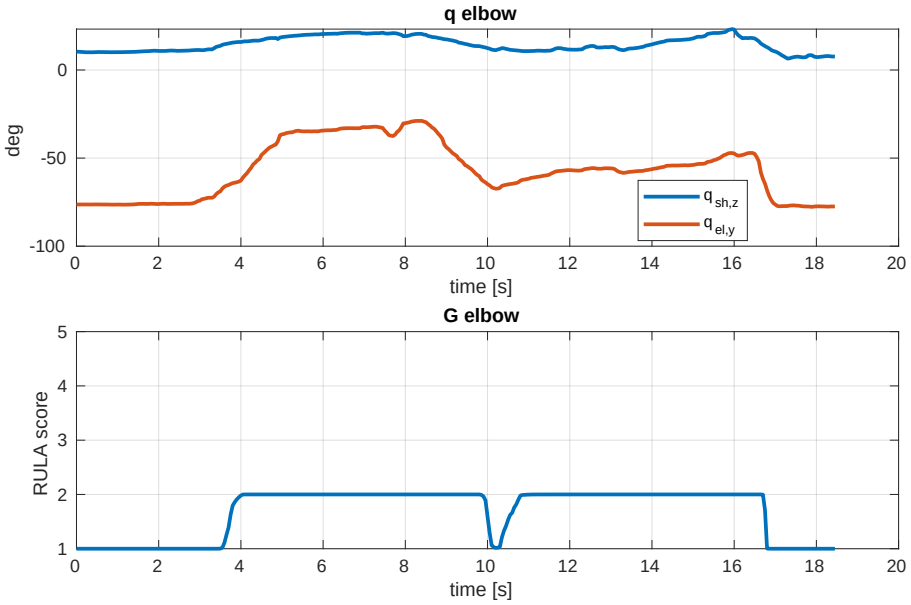


Figure 5.21 Action 2: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant.

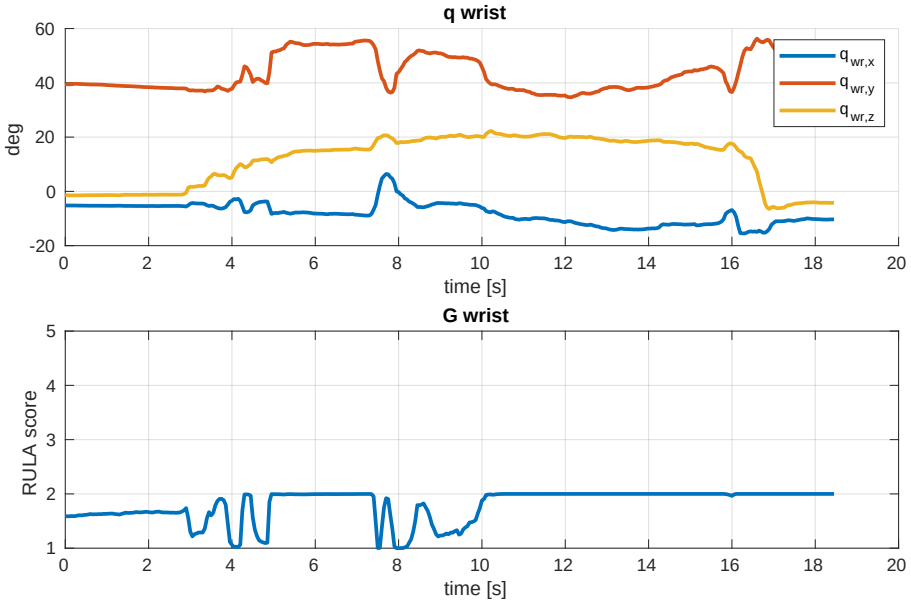


Figure 5.22 Action 2: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant.

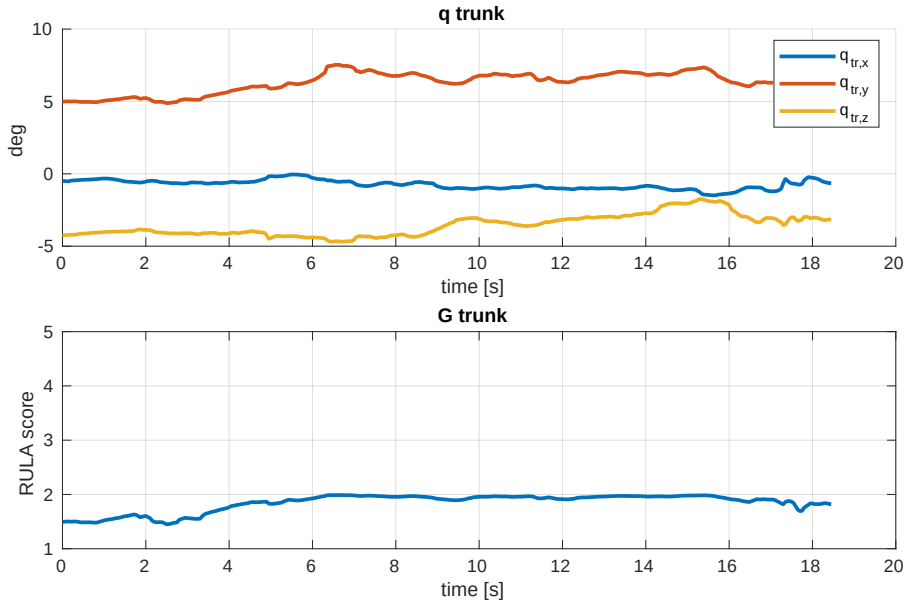


Figure 5.23 Action 2: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant.

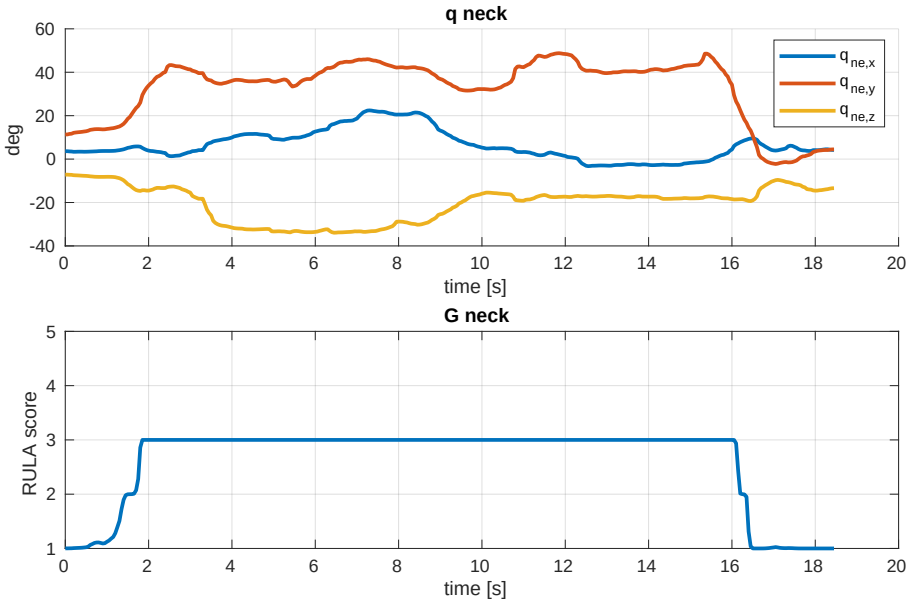


Figure 5.24 Action 2: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant.

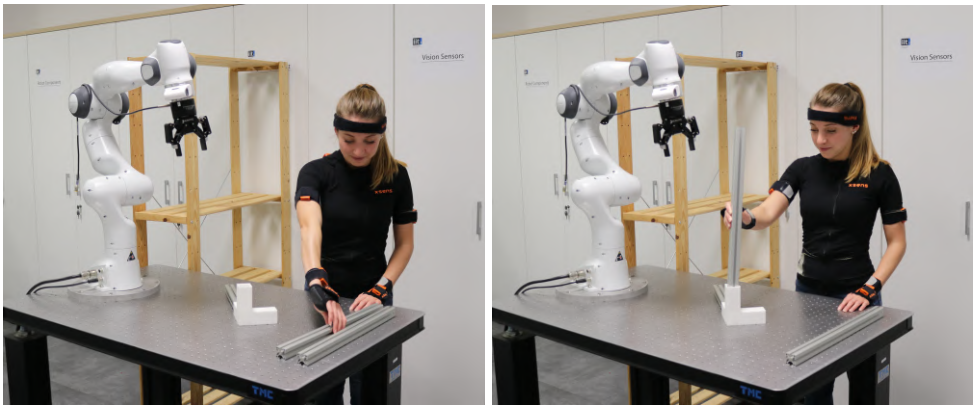


Figure 5.26 Action 3: pick and assemble the aluminum profile S_1

Action 3: pick and assemble the aluminum profile S_1 This action consisted in grasp and assemble the aluminum profile S_1 that was at the human's left. The corresponding hollow for S_1 was the one of J pointing upwards. As already mentioned, the human was asked to perform each action with the right arm, since it was the one under control. Therefore, to reach object S_1 in its position, shoulder was raised (Figure 5.20), the lower arm extended (Figure 5.21), the trunk performed a flexion and a left twist (Figure 5.23). This action was equivalently risky for every joint (as shown in Figure 5.32).

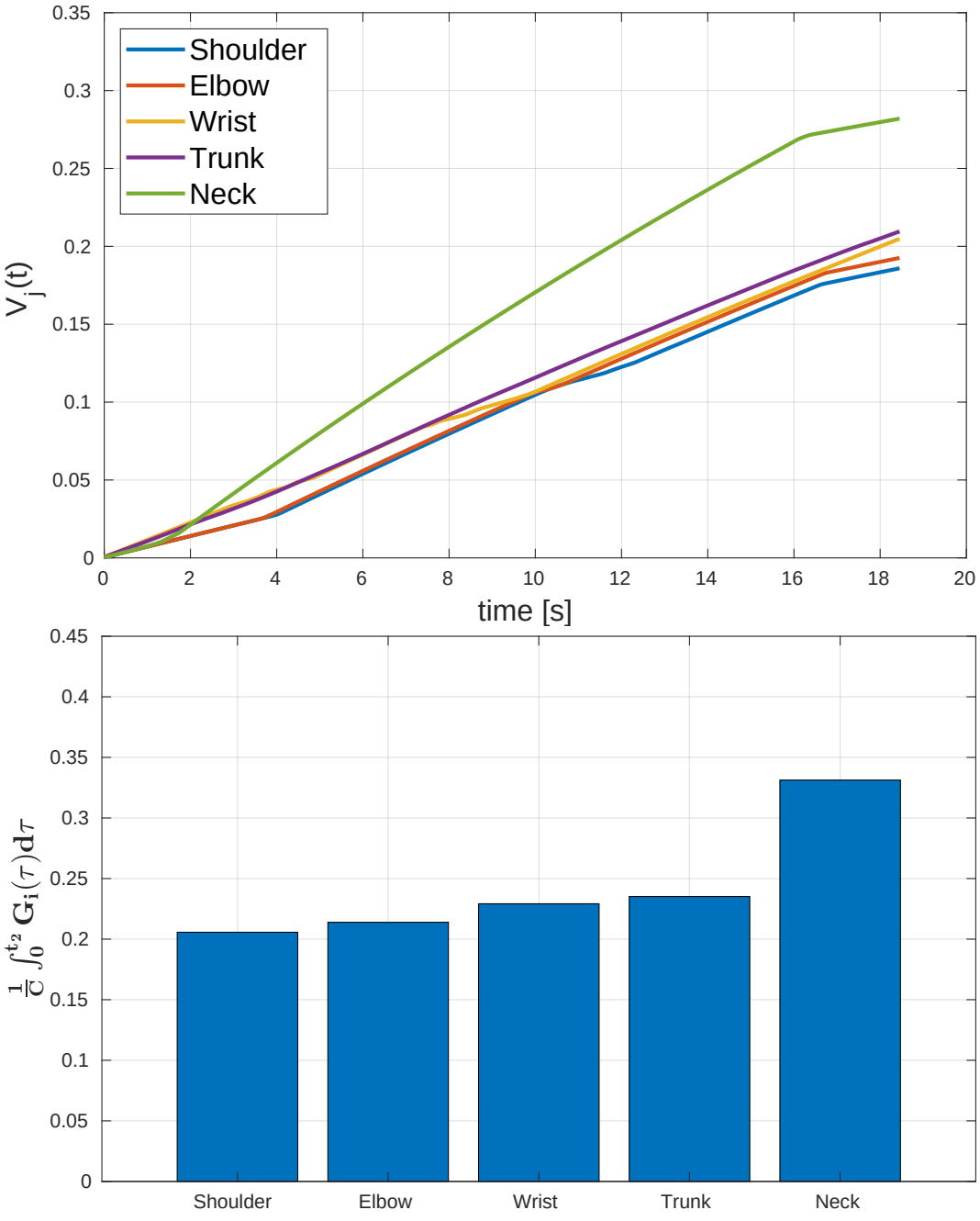


Figure 5.25 The *kinematic wear* trend for all the monitored joints during the performance of a_2, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_2 (bottom).

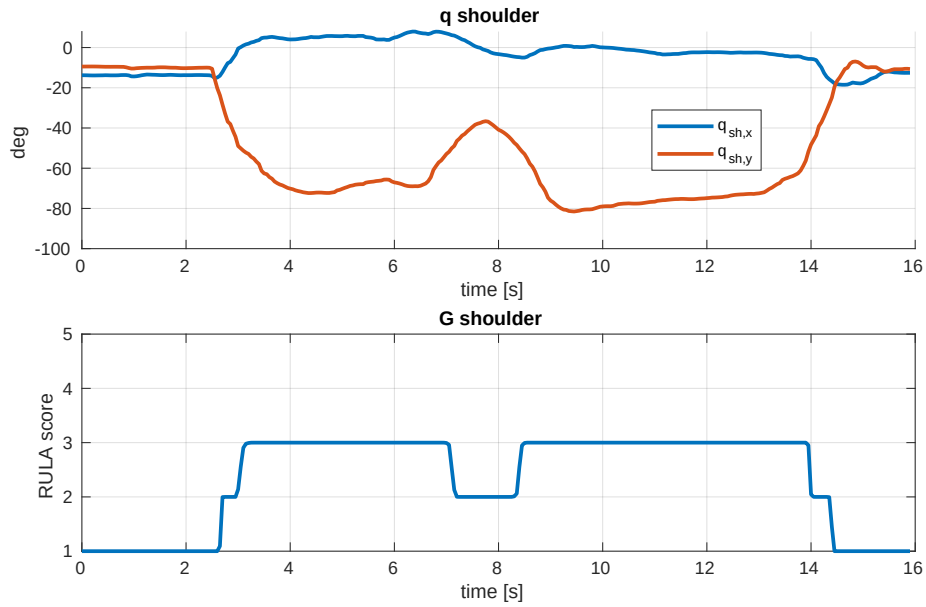


Figure 5.27 Action 3: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant.

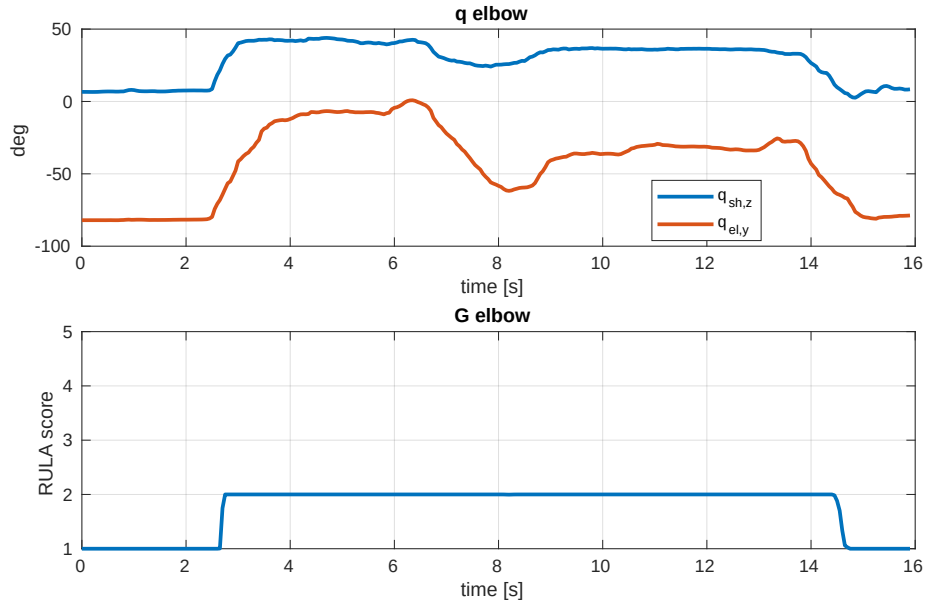


Figure 5.28 Action 3: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant.

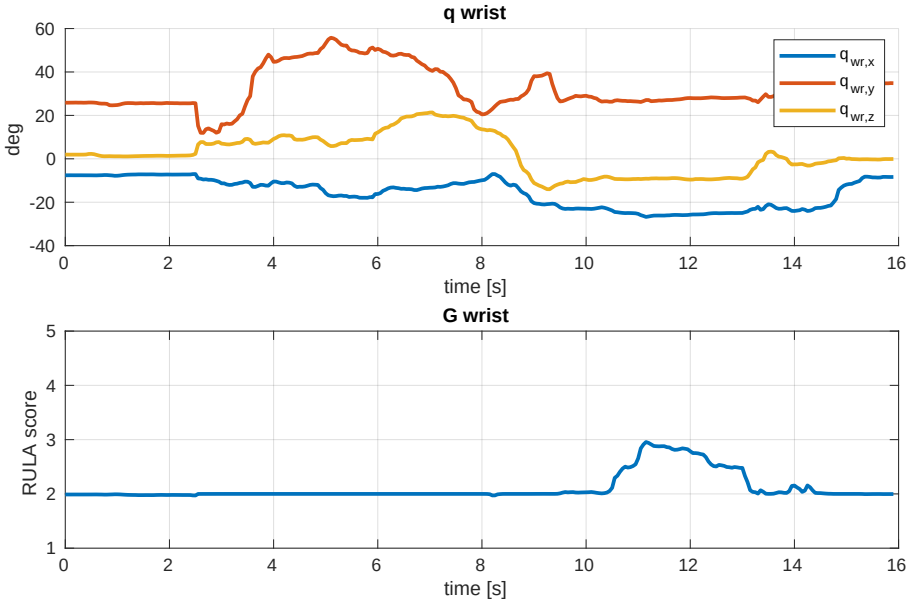


Figure 5.29 Action 3: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant.

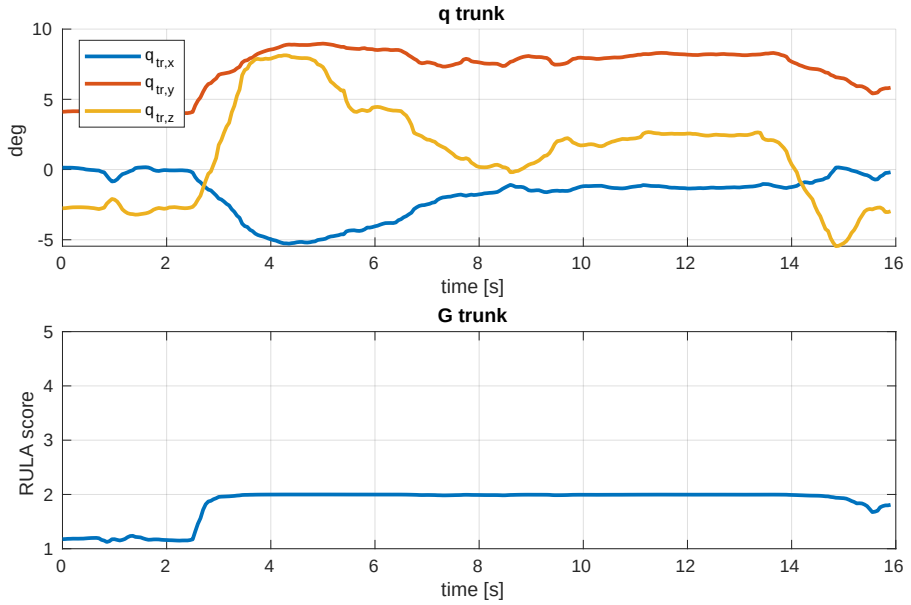


Figure 5.30 Action 3: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant.

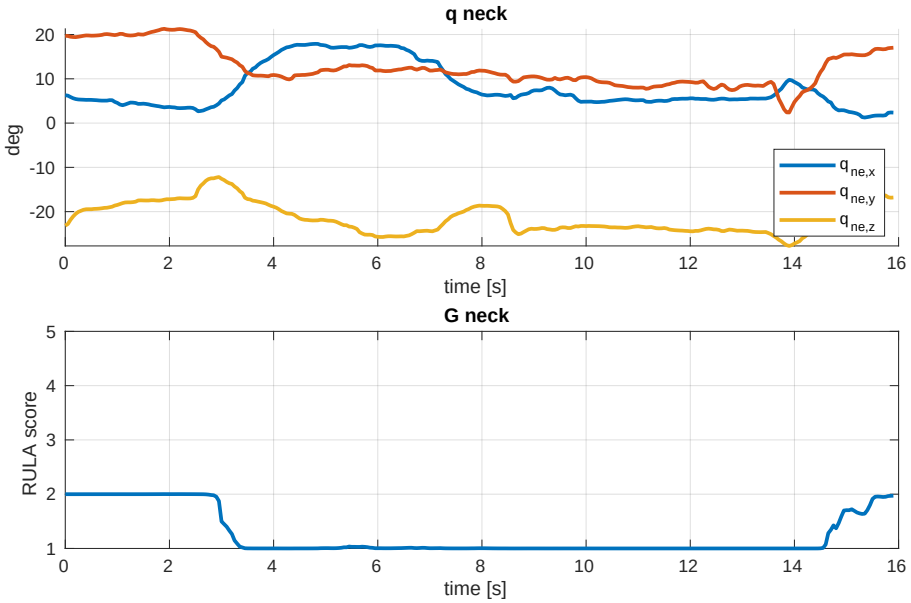


Figure 5.31 Action 3: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant.

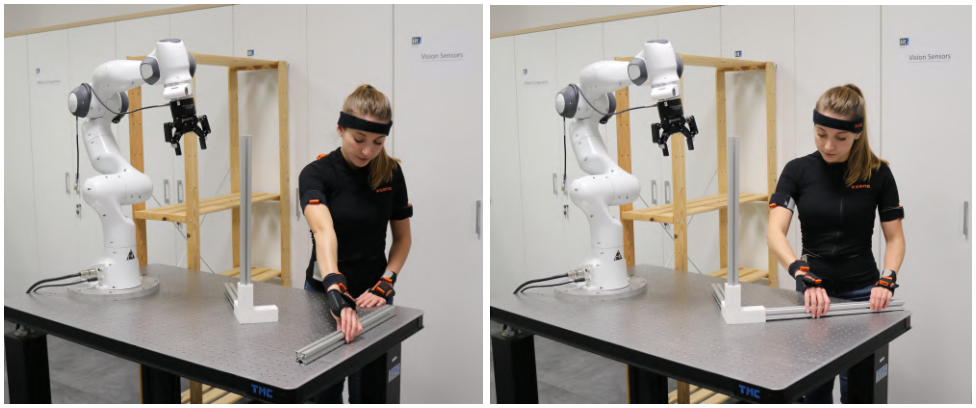


Figure 5.33 Action 4: pick and assemble the aluminum profile S_2

Action 4: pick and assemble the aluminum profile S_2 Action a_4 consisted in grasping and assembling the profile S_2 with the correct J hollow (the left one). Similarly to a_3, since the grasping phase was performed with the right hand and the object S_2 was at the operator's left, the shoulder was raised (Figure 5.34), the lower arm extended (Figure 5.35), the trunk bent ((Figure 5.37)). Then all the joints were at a comparable level of risk at the end of the execution (Figure 5.39). The difference with respect to a_3 was in the assembling phase: to interlock S_2 in the left hollow J required to perform a limited arm flexion, than the one necessary to insert S_1 in the upper hole. This is the reason why the shoulder column is taller for a_3 than for a_4.

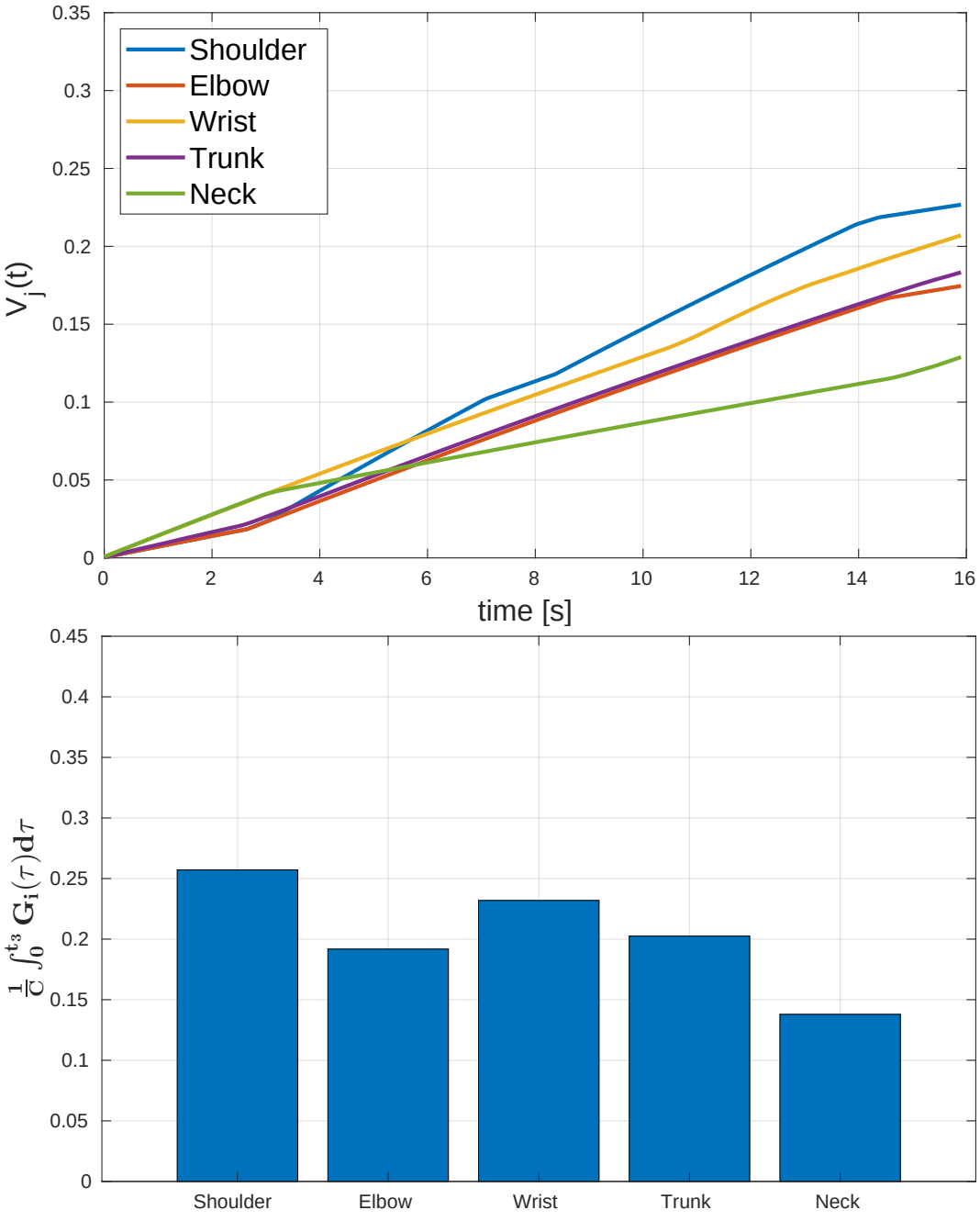


Figure 5.32 The *kinematic wear* trend for all the monitored joints during the performance of a_3, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_3 (bottom).

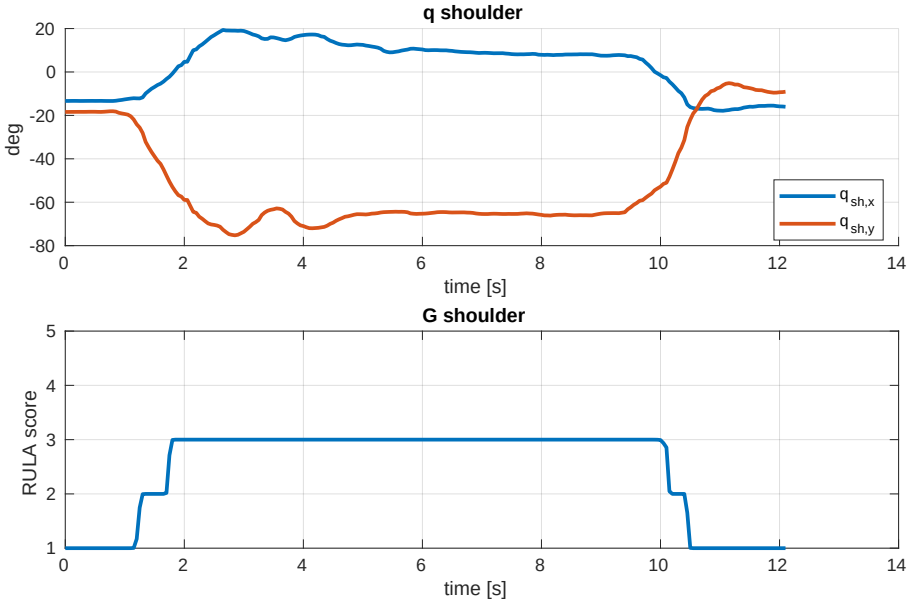


Figure 5.34 Action 4: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant.

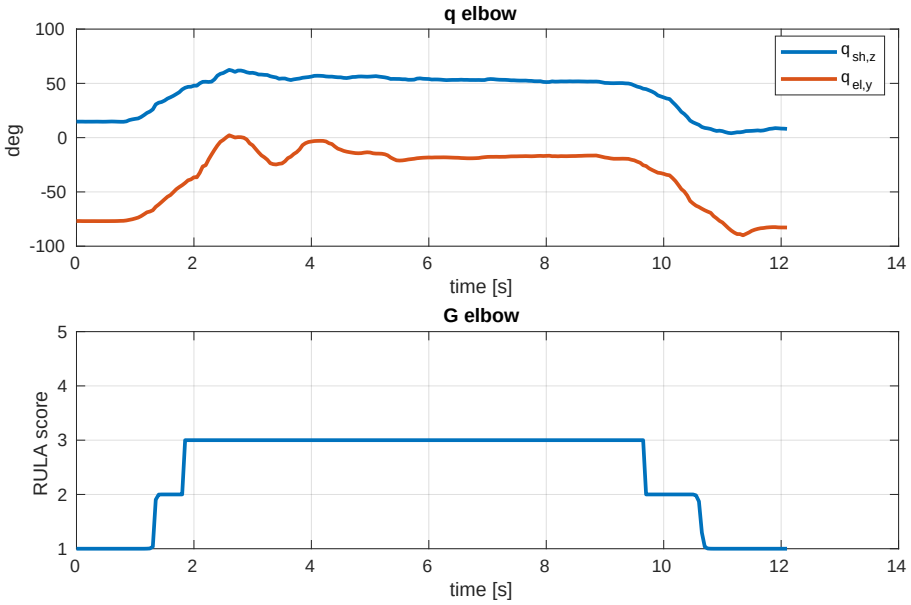


Figure 5.35 Action 4: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant.

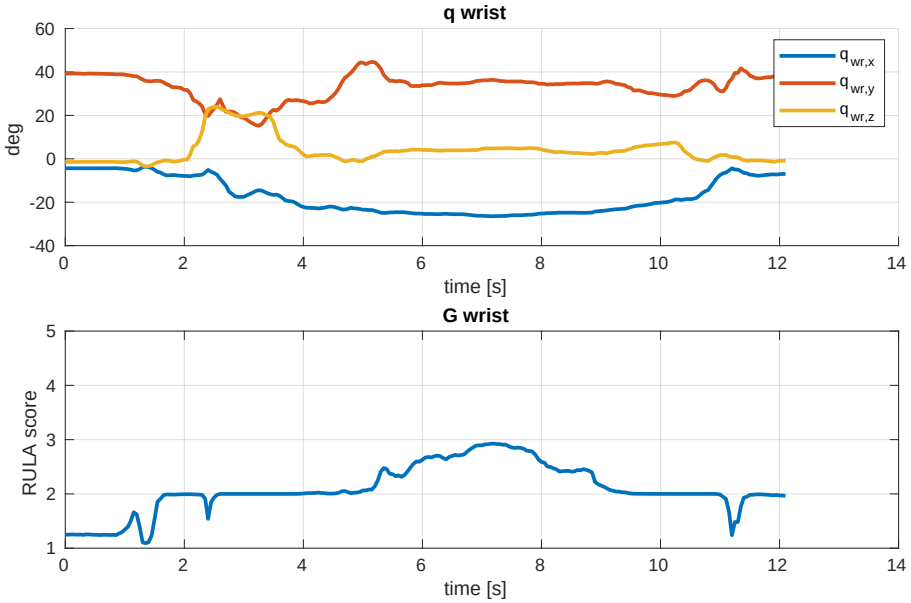


Figure 5.36 Action 4: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant.

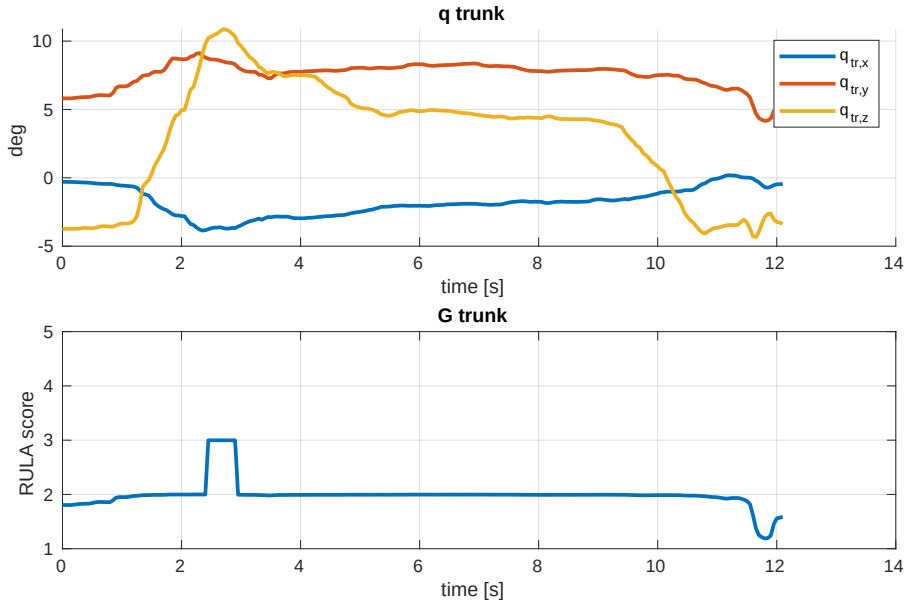


Figure 5.37 Action 4: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant.

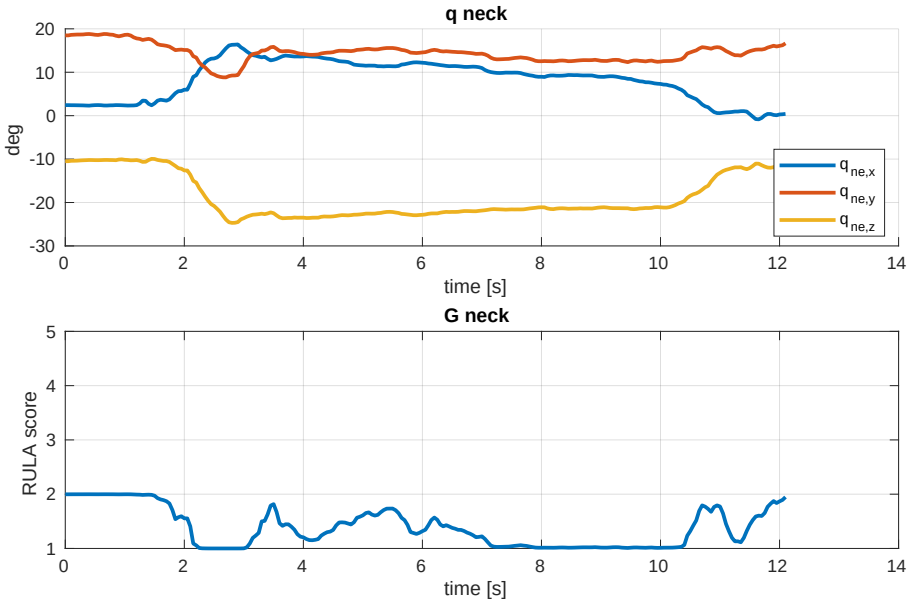


Figure 5.38 Action 4: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant.

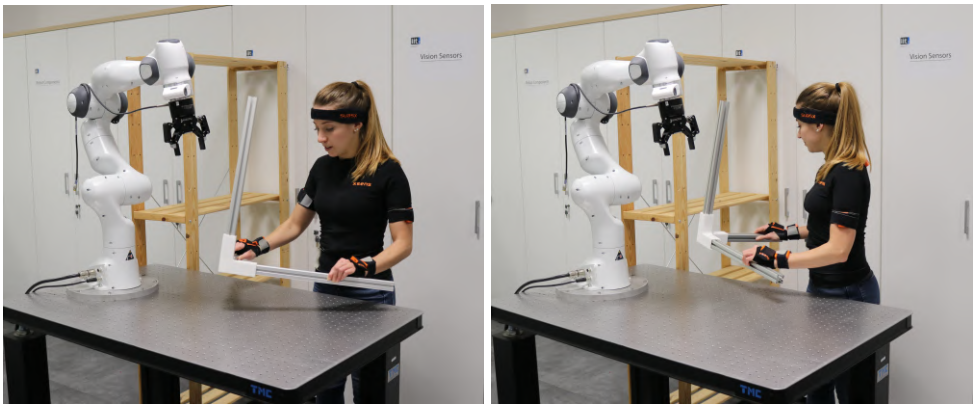


Figure 5.40 Action 5: pick and place the entire assembly on the shelf

Action 5: pick and place the entire assembly on the shelf This action consisted in moving the completed assembly away from the assembly area and in positioning it on the shelf. The performance of this action resulted in having the trunk more at risk than the other joints, as shown in Figure 5.46. This is due to the fact that the shelf was at the operator's right, then, once the entire assembly was grasped, the human needed to rotate the trunk rightwards, increasing the RULA score associated to this joint.

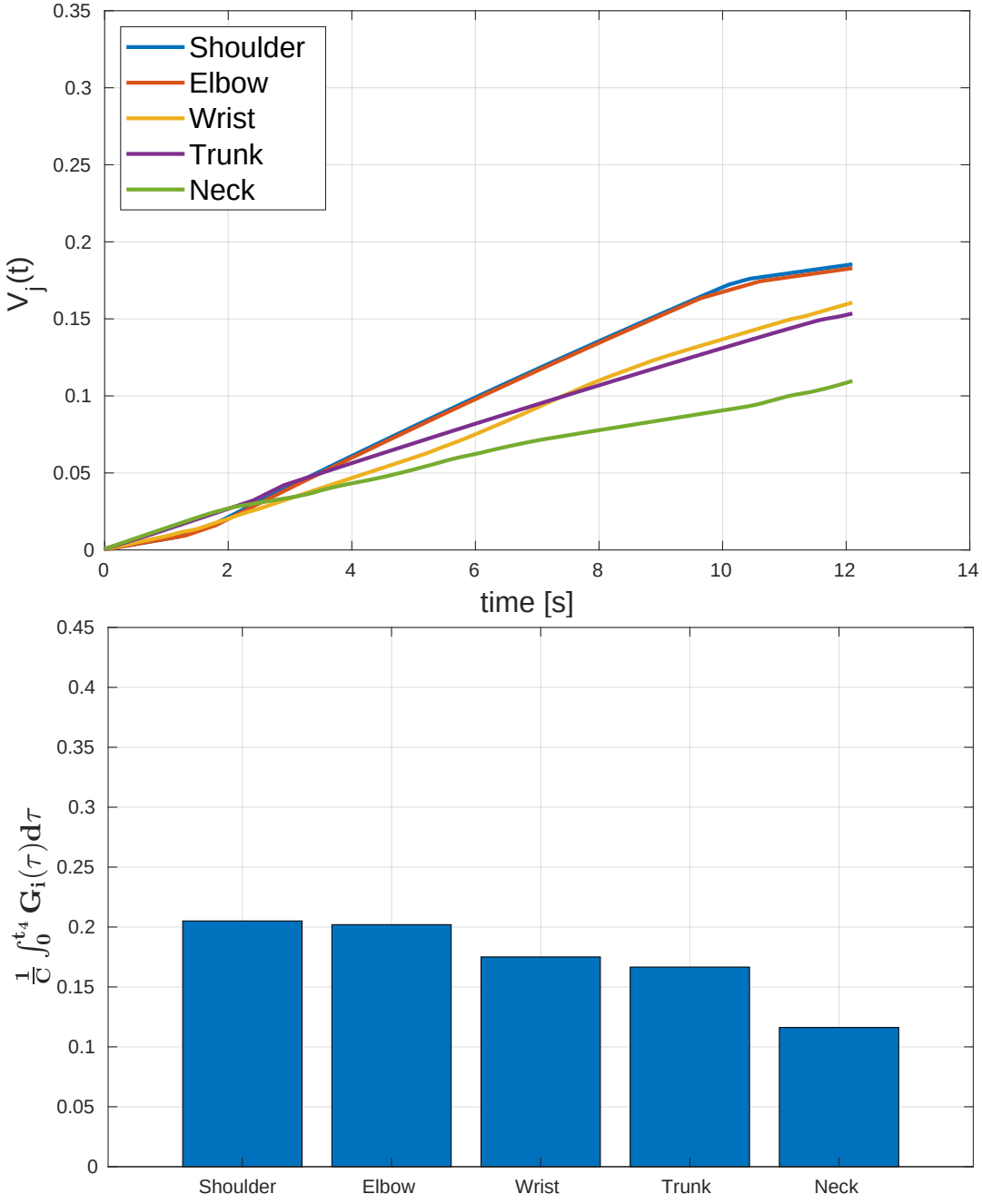


Figure 5.39 The *kinematic wear* trend for all the monitored joints during the performance of a_4, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_4 (bottom).

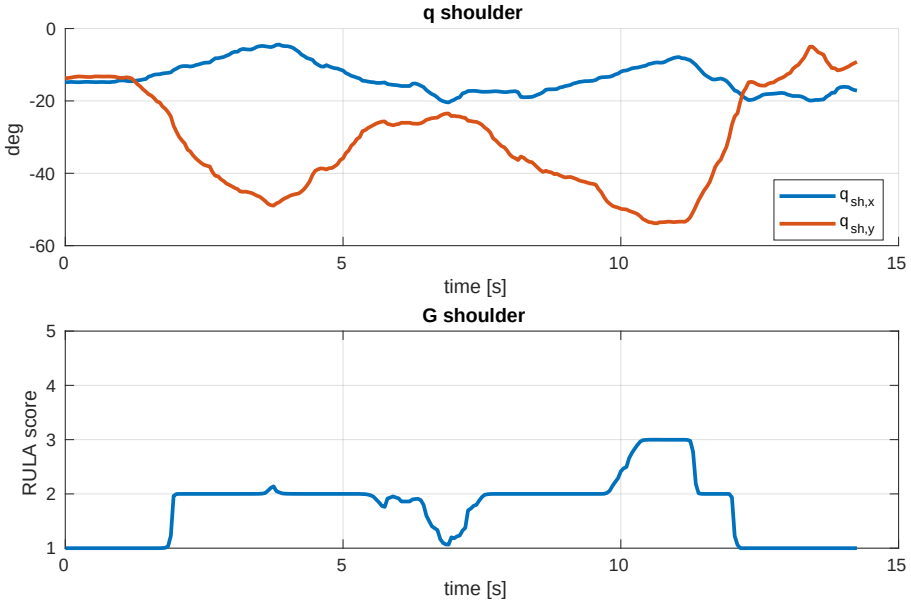


Figure 5.41 Action 5: effects at shoulder level. (top) joint angles, (bottom) resulting RULA score at each instant.

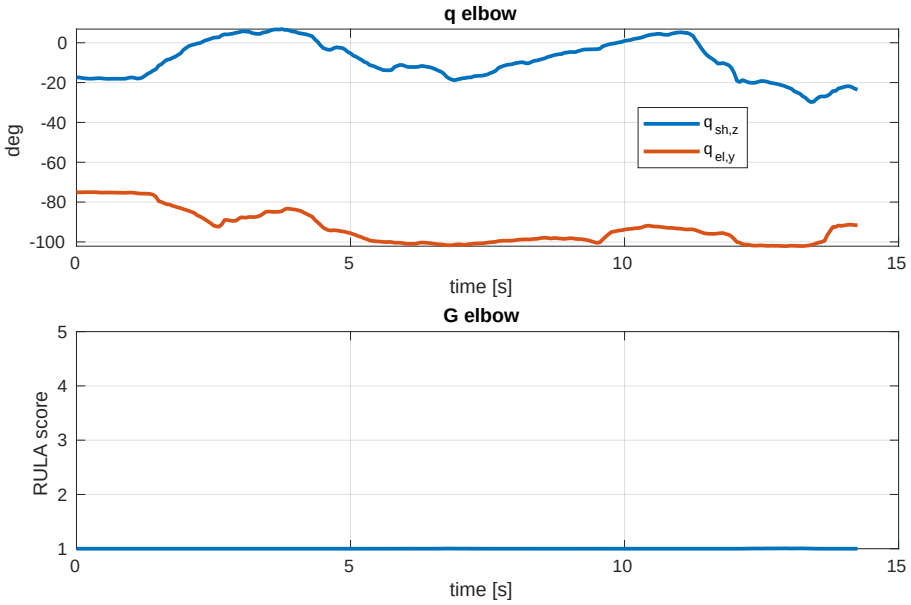


Figure 5.42 Action 5: effects at elbow level. (top) joint angles, (bottom) resulting RULA score at each instant.

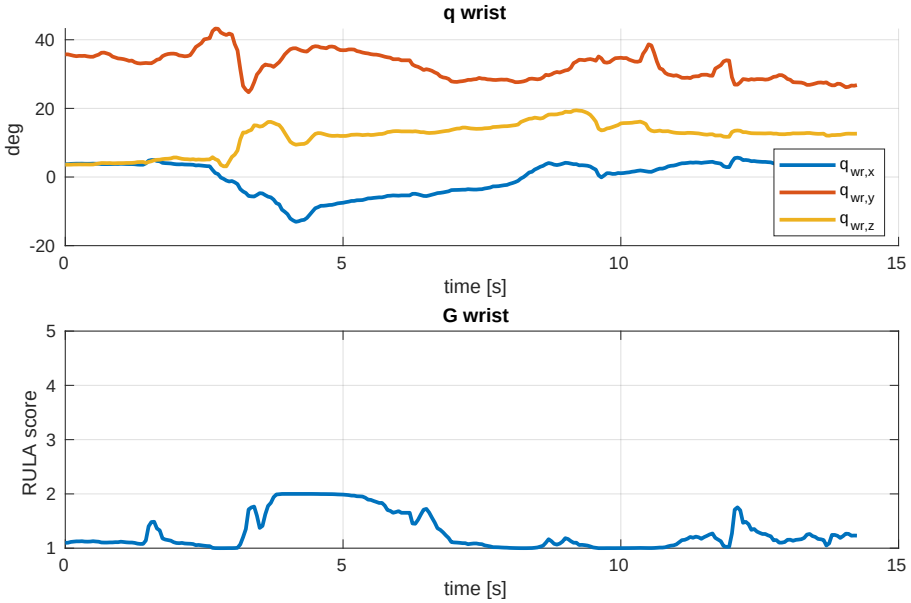


Figure 5.43 Action 5: effects at wrist level. (top) joint angles, (bottom) resulting RULA score at each instant.

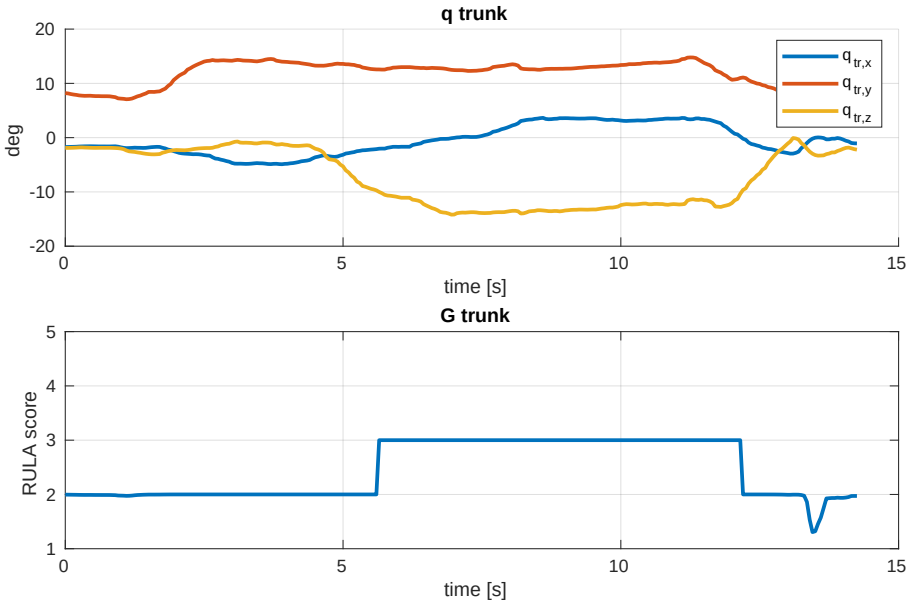


Figure 5.44 Action 5: effects at trunk level. (top) joint angles, (bottom) resulting RULA score at each instant.

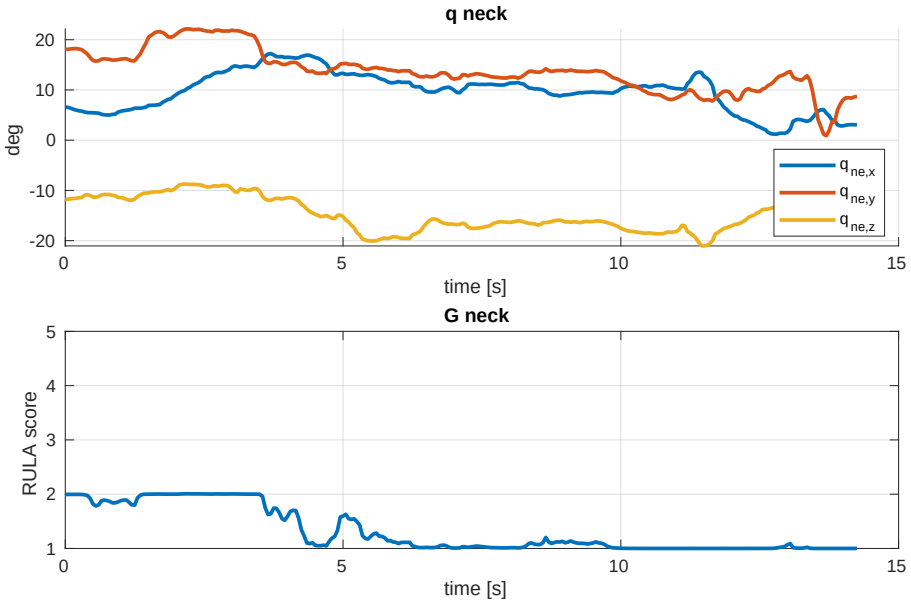


Figure 5.45 Action 5: effects at neck level. (top) joint angles, (bottom) resulting RULA score at each instant.

5.2.2 Overview of the results

Regarding the formalization of the *kinematic wear* (Equation 3.16), the experimental results suggest that:

- it increased steeply if the current joint angles belonged to a portion of RoM of higher risk;
- at the end of the performance of action a_k, $V_i(t_k)$ depended also on the time spent in each portion of RoM that joint i crossed.

Moreover, we can appreciate that such a risk indicator can describe every action, allowing a distinction among them according to which joint it is stressed at the most: according to the considered subject performance, a_1 resulted to be more risky for the shoulder, a_2 for the neck, and a_5 for the trunk, and we can understand it by just looking at the *kinematic wear* behavior, without checking joints configurations.

In the experiment described in section 5.1, the human status was updated just considering if an action was performed or not, without accounting for the way the action was executed. Instead, in this experiment, the analysis of each single action in the task in terms of the chosen ergonomic risk indicator was necessary to allow a prediction of the future risk, starting from the current effective human status. In particular, this analysis corresponds to

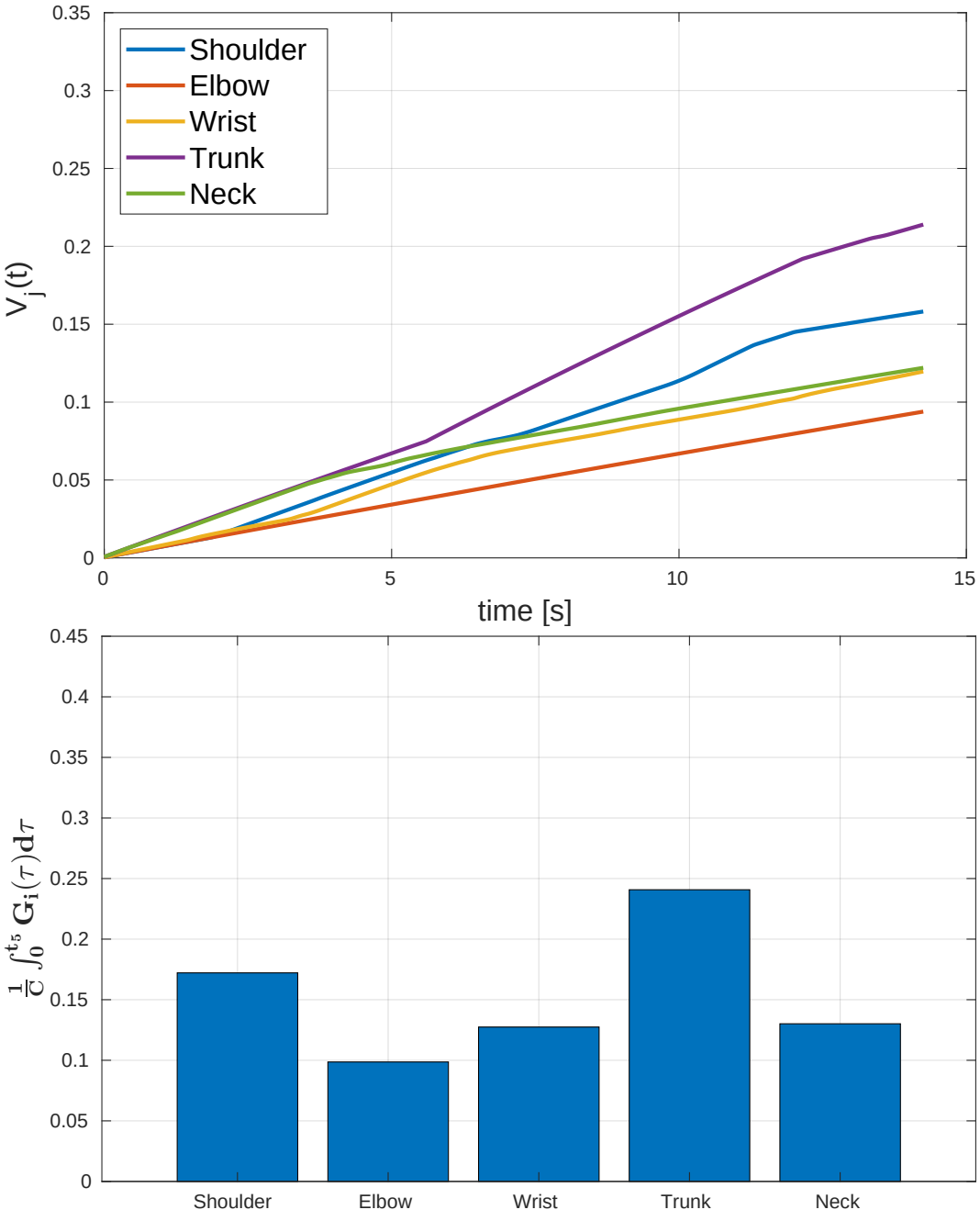


Figure 5.46 The *kinematic wear* trend for all the monitored joints during the performance of a_5, starting from zero initial conditions (top). The increase of each V_i depends on the accumulated risk during a_5 (bottom).

the calibration phase, a subject-specific step that has to be performed before the effective HRC, since it allows to find the prediction model parameters (see section 3.3). Once such parameters were computed, we simulated a cooperative task envisioning four repetitions of the assembly sequence. The sequence was fixed: a_1, a_2, a_3, a_4, and a_5. The variability was in assigning each of these action either to the human worker or to the robot. The hyper-arcs costs were assigned as described in section 3.1, while the cost of robot hyper-arcs was such as to assign the action to the robot when at least one $\hat{V}_i(t_{k+1}) > V_{th}$, where the $V_{th} = 0.8$. In Figure 5.47, the *kinematic wear* trend for each joint during the whole task is shown. In particular, it can be appreciated that:

- the charge phase was different for each joint, according to the level of involvement in the current action;
- the discharge phase, that occurred when the action was allocated to the robot, allowing the human to rest, was equal for all the joints, since it did not depend on the Ergonomics Assessment Function $G_i(\mathbf{q})$.

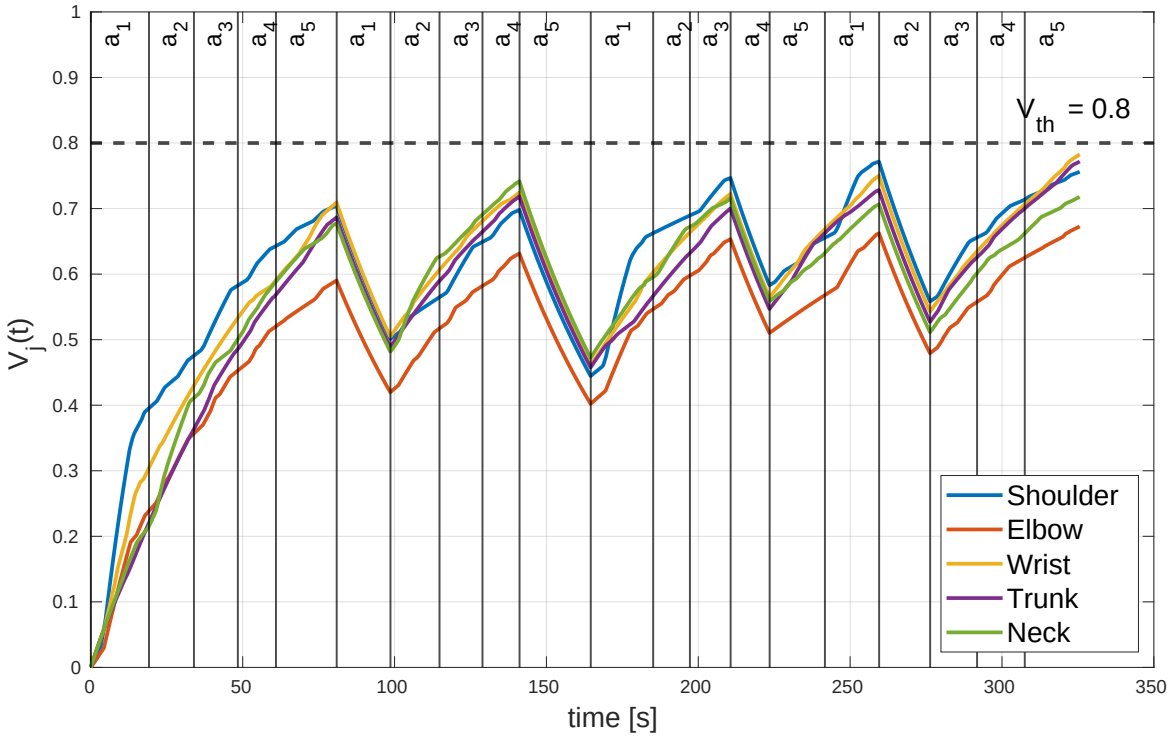


Figure 5.47 The *kinematic wear* trend for each joint during the performance of a cooperative assembly. The increasing trend occurred when the action was allocated to the human operator, the decrease when the action was allocated to the robot.

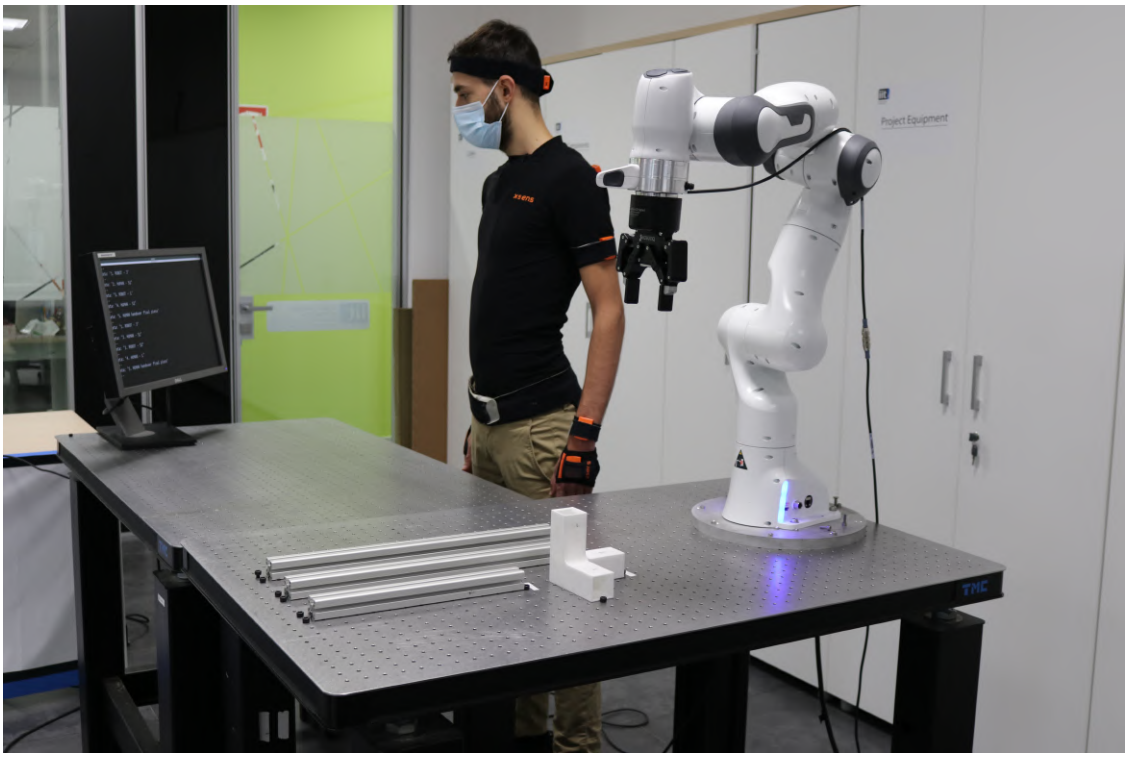


Figure 5.48 Experimental Setup. The workstation had a "L" shape, the assembly pieces were placed at the operator's left as well as the robot co-worker; in front of the human, the assembly area and the monitor to read the allocations were located. The human wore the Xsens to record the joint angles.

5.3 Multi-subject Experiment

Once the correctness of the role allocation process and the properties of the *kinematic wear* as risk indicator were evaluated, we carried out a multi-subject experiment. The goal was to test the complete framework functioning, by mimicking an industrial assembly line. Subjects were asked to work both by themselves and in cooperation with the robot. Then, with the aid of questionnaires (attached in A), we managed to draw conclusions about the presented method efficiency and its level of usefulness in industrial contexts.

Experimental Setup The goal of this experiment was to repeat the assembly of the pieces in Figure 5.1 for ten times, moving it from the assembly area, once each assembly was completed. As Figure 5.48 shows, the workstation had a "L" shape. On the workbench, at the subject's left, the assembly pieces were positioned from the nearest to the further S_2, S_1, L, J. The Franka Emika Panda manipulator was on the same workbench. The table in front of the subject, instead, provided the assembly area, i.e. where the pieces were assembled.

Moreover, the monitor on the right communicated to the worker the solution of the allocation algorithm. The human wore the Xsens suit to monitor their joint angles. The five actions necessary to conclude a repetition were: pick and place J in the assembly area (a_1), pick and insert L in the upper hollow (a_2), pick and insert S_1 in the left hollow (a_3), pick and insert S_2 in the right hollow (a_4), move away the entire assembly (a_5). In particular, the last action envisioned the handover of the entire assembly to another human operator, who was in charge of disassembling the final product and re-positioning the pieces in their original site, to simulate an industrial conveyor roller. Furthermore, to replicate the production line time constraints typical of industrial setups, we introduced the takt time, the manufacturing term to describe the required product assembly duration that was needed to match the demand. In particular, given that subjects were new to HRC tasks, then maybe not confident to interact with a robot, and considering the robot speed limitations that avoid it to move too fast, we set such a time to $T_{takt} = 38s$. The five actions had to be completed within such a time.

Moreover, since the right arm was the monitored one, the subjects were asked to reach and grasp each object with the right hand, then they could handle it with both the hands. Differently from the pilot experiments, the sequence of actions was not fixed: while actions a_1 and a_5 were fixed as the first and the last of the assembly sequence, the order of the other three actions was online decided by the AO*. Each action could be assigned either to the human or to the robot, except for the last one, always assigned to the human. The hyper-arcs costs were assigned as described in section 3.1, while the cost of robot hyper-arcs was set to $C_R = 2.5$, meaning that the allocation to the robot happened when $\hat{V}_i(t_{k+1}) \approx V_{th}$, where the $V_{th} = 0.8$.

Experimental Protocol As mentioned before, the task consisted of ten repetitions of the assembly sequence in two different modalities, one in cooperation with the robot (*HRC* mode), the other without any co-worker (*H only* mode). The order in which the two task were executed changed randomly. We proposed the experiment to twelve healthy subjects (2 girls and 10 boys 28 ± 3 year-old). Half the subjects starts with the *HRC* modality, the other half with the *H only* one.

***HRC* mode** In this condition, the human was asked to complete the task with the robot aid. Therefore, to know which action was the next one to perform and which was the agent allocated, the human had to look at the monitor, where the allocation algorithm results were printed. If the action was allocated to the human, it meant he/she had to pick and place the indicated object (see Figure 5.49). On the other hand, if the action was allocated to the robot,

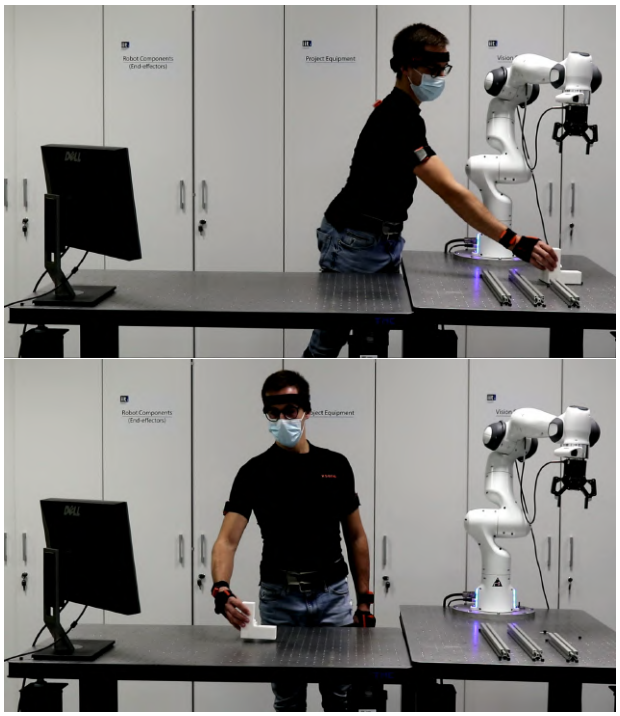


Figure 5.49 Pick and place object *J* action assigned to the human worker.

then, the robot reached, grasped, and handed over the concerned object to the human worker (see Figure 5.50). The positions of the assembly pieces to be handled by the robot and the handover position (chosen to make the handover as much natural as possible) were directly inserted into the robot script. Moreover, we programmed the robot velocity to ensure to complete each repetition within T_{takt}.

***H only* mode** In this modality, the human had to perform the ten repetitions alone, without the robot help. The worker could choose the assembly plan, as long as pieces were picked one at a time. To comply with the takt time constraint, we asked the human to handle a chronometer, to be sure to start the new repetition after the T_{takt} was expired.

After the execution of each task modality, the subject were required to answer to a standard questionnaire, the NASA Task Load Index (NASA-TLX). Therefore, each subject filled two NASA questionnaires, one after *H only* and another after *HRC*. It is a widely used, subjective, multidimensional assessment tool that rates perceived workload in order to assess a task, system, or team's effectiveness or other aspects of performance. In addition, at the end of the two experiences, a custom questionnaire was proposed to evaluate the *HRC* modality. Unlike the NASA TLX, this custom questionnaire is not validated, but it can give useful information

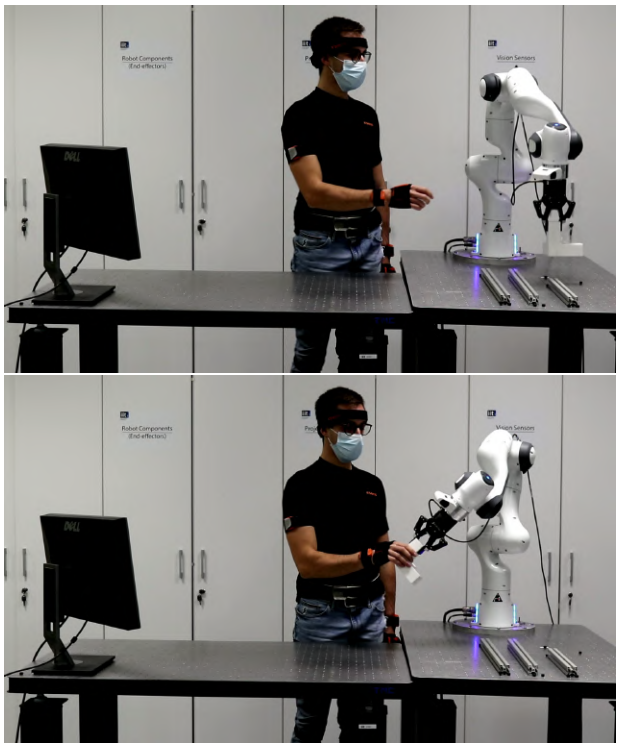


Figure 5.50 Pick and place object *J* action assigned to the cobot.

with regard to some aspects of the presented framework, like the ergonomic role allocation efficiency and the usability in industrial environments. Both the questionnaires are attached in A. Hereafter the experimental protocol follows:

1. wear and calibrate Xsens, according to the subject body measurements
2. parameter calibration procedure: the task atomic actions were described in terms of *kinematic wear* to compute the prediction model parameters
3. RANDOM choice (to avoid biases in the results due to training effects) of which task the subject had to perform first, between the *H only* mode or the *HRC* mode
4. task according to *H only* (or *HRC*) mode
5. questionnaire NASA TLX to compile in relation to the just performed task
6. task according to *HRC* (or *H only*) mode
7. questionnaire NASA TLX to compile in relation to the just performed task
8. custom questionnaire to compile in relation to the task in *HRC* mode

Parameter Calibration Procedure Before starting the first task performance, a calibration procedure was performed. Such a procedure required the characterization of each assembly action in terms of *kinematic wear* (see section 5.2) and it was necessary to compute the prediction model parameters $\alpha_{k,i}$ and $\beta_{k,i}$ for each action a_k and each joint i (see section 3.3). Due to the design of such prediction model parameters, an error in prediction meant that, during the online performance, each action a_k was executed in a different manner with respect to the one recorded in the calibration phase. To reduce this mismatch between the same action performances, we designed a calibration phase in which each single action a_k was repeated several times, and then the obtained $\alpha_{k,i}$ coefficients were averaged. We set to three the number of repetitions; however, a higher number of repetitions could lead to have parameters mirroring the standardized performance of each action. Another approach, that could be used to find such parameters, was to compute them during the online task performance by means of a weighted average. By associating a higher weight to the $\alpha_{k,i}$ returned by the last time action a_k was executed, the system would have learned how the worker was used to execute each action.

Experimental Results The results collected by analysing the subjects' answers to the NASA TLX questionnaire are shown in Figure 5.51. The paired column stand one for the experiment in *H only* mode (the blue one), and one under the *HRC* conditions. The aspects the questionnaire evaluates are: the required Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), the Performance (P) quality, the spent Effort (E), and the felt Frustration (F). Each of these aspects can be rated from 0 to 100 (each notch counts as 5). Then, by means of an additional table (see A), that presents all the pairs of the aspects mentioned above, it is possible to choose, for each pair, which has been more relevant in the task accomplishment. In this way, one can determine how much each item is weighted in the subject's experience in the just performed task. In particular, such weights are expressed in percentages and the overall sum leads to 100%. Therefore, the bar height in Figure 5.51 encodes the mean value of scores given by all the subjects to that aspect, while the percentage was found by averaging the weights given to that aspect. For each feature, such percentages were almost the same in the two conditions, suggesting that the two tasks can be compared by inspecting the differences in each single aspect. Even though the Mental Demand resulted low for both the conditions, it is a bit higher for the *HRC* mode: this may be due to the fact that checking on the monitor which action had to to be executed next and the corresponding allocated worker, was kinf of stressful. However, the development of an intuitive interface was not the primary objective of this work, since our interest was just to communicate to the

worker the allocation solution. Future works could improve such communication channel, that of course may influence a HRC experience.

On the other hand, the Physical Demand is absolutely higher for the *H only* mode. Moreover, the difference between the two conditions resulted to be statistically significant according to the Wilcoxon signed-rank test, a non-parametric statistical hypothesis test (Woolson (2007)). The p-value returned by the comparison for PD in the two experimental conditions is $p = 0.0059$, while the test statistic is $W = 1.5$. Considering that the critical value of W equals 14 (according to the number of subjects and of the chosen critical $p = 0.05$), since $W < 14$, we have a statistically significant evidence about such a difference. Since the final goal of our presented framework was to ensure ergonomic working conditions to the human operators, such a result reveals that our method is promising in this direction.

The Temporal Demand for the two modalities was perceived to be similar: that is what we expected by fixing the takt time T_{takt} for the two experimental conditions. Nevertheless, subjects considered the *H only* mode less demanding, since they could complete the assembly without waiting the robot intervention.

For what concerning the Performance, since the task was simple, all the subjects were satisfied with their work in both the two conditions (the bar height represents the level of failure, that is very low). Anyway, to reach the same level of Performance, subjects retained to have worked harder for the task accomplishment in the *H only* mode: the difference between the two modalities was again statistically significant (the p-value returned by the comparison for E in the two experimental conditions is $p = 0.0098$, while the test statistic is $W = 7.5$). As for the Physical Demand, to have such a contrast between the required Effort suggests that the framework was improving the human ergonomics. However, since the number of subjects involved in this experiment is limited these results should be considered preliminary, even though promising.

Finally, even if there was not a statistically significant difference, the level of Frustration was higher for the *H only* mode. This could be because, with the robot aid, the assembly rhythm was marked by the robot timing of performing the actions allocated to it and it was steady through the repetitions, while, without any co-worker, when the human finished the repetition before the takt time T_{takt} expired, he had to wait to start a new repetition. Such a waiting might have irritated or annoyed subjects.

By means of the custom questionnaire (see A) we aimed to investigate three main aspects of our framework: the efficiency of the role allocation algorithm, the granted ergonomics conditions and the usability of the presented method in industrial environments. Hereafter the asked questions:

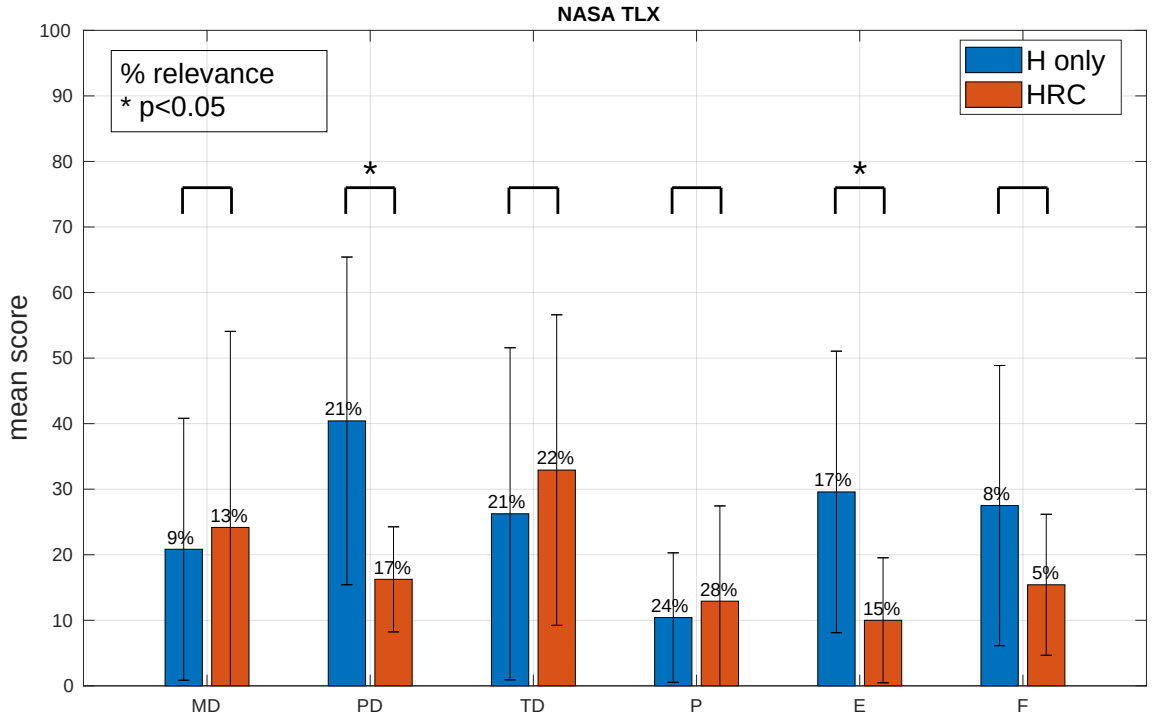


Figure 5.51 Results by collecting the subjects' answers to the NASA (Task Load indeX) TLX questionnaire. The bars height correspond to the mean value of the subjects score for each task aspect: Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Performance (P), Effort (E), Frustration (F). The two matched bars are for the experiment in *H only* (blue one) and *HRC* (red one) mode.

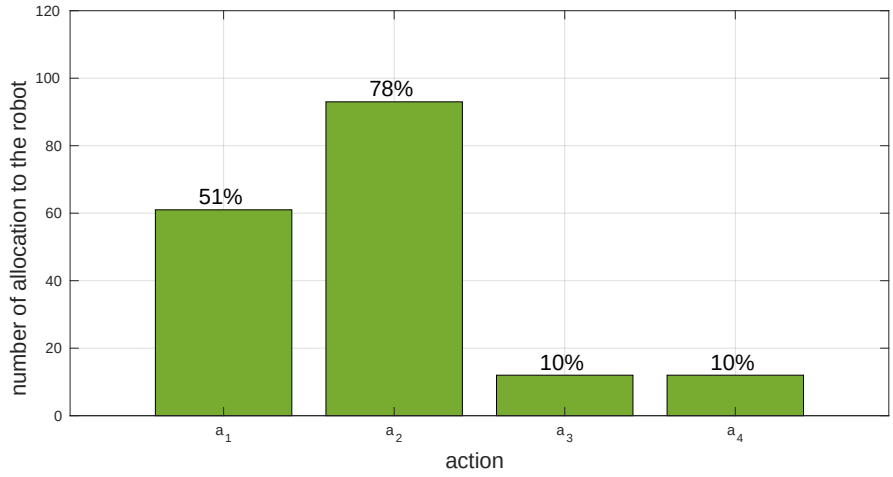


Figure 5.52 How many times each action has been allocated to the robot during the cooperation. Considering all the subjects (twelve), each action was executed either by the human or by the robot for 120 times in total. On the top of each bar the number of allocation to the robot for each action in percentage is expressed.

1. I was glad to take a break when the task was allocated to the robot
2. The robot was hindering me in the task accomplishment.
3. It was easy to understand when I had to take action.
4. When I was tired, the robot was not able to intervene.
5. I feel safe when collaborating with the robot.
6. The robot can choose correctly which action is too demanding for me.
7. I would suggest to use the robot for this kind of tasks.
8. The robot help makes me feel less fatigued at the end of the task.
9. I had to wait for the system to inform me about the next task.

Each one of the nine questions was referred to one or more the just mentioned categories, to which subjects had to answer by means a 7-point Linkert scale, from "Not at all" to "Extremely". In Figure 5.53, on the x-axis there is the question number, while the bar height represents the mean score given to each question. The score is positive when the answer is positive and vice-versa. The bar color represents the category to which each question "belongs". As we expected, the results reveal that subjects were glad to take a break when the task was allocated to the robot, leading them to feel less fatigue at the end of the whole task. Nevertheless, for some subjects the task was so easy that they did not want to be helped. This could be because we set a too low threshold value, but also due to a limited number of repetitions our experiment provided.

However, the majority of subjects perceived also to be helped by the robot in the most stressful tasks. That is proved by the fact that the action allocated the most was a_2 (78% of a_2 was allocated to the robot, that is 93 times on 120, see Figure 5.52), that required to grasp object L, positioned in the furthest location from the human, and inserted it in the upper hollow of J demanding to rise the right arm (i.e. shoulder more at risk). Action a_1 was allocated to the robot 51% of times: this is because a_5 was always executed by the human, then at the end of the assembly repetition, the *kinematic wear* value probably resulted so close to the threshold, that the allocation of a_1 to the robot was triggered. The other two actions a_3 and a_4 were most allocated to the human agent, due to their low ergonomic risk. Overall, the cooperation envisioned a 30% of robot activity, that means a bit more than $\frac{1}{3}$ actions were allocated to it. Furthermore, testers stated that they neither waited for the system

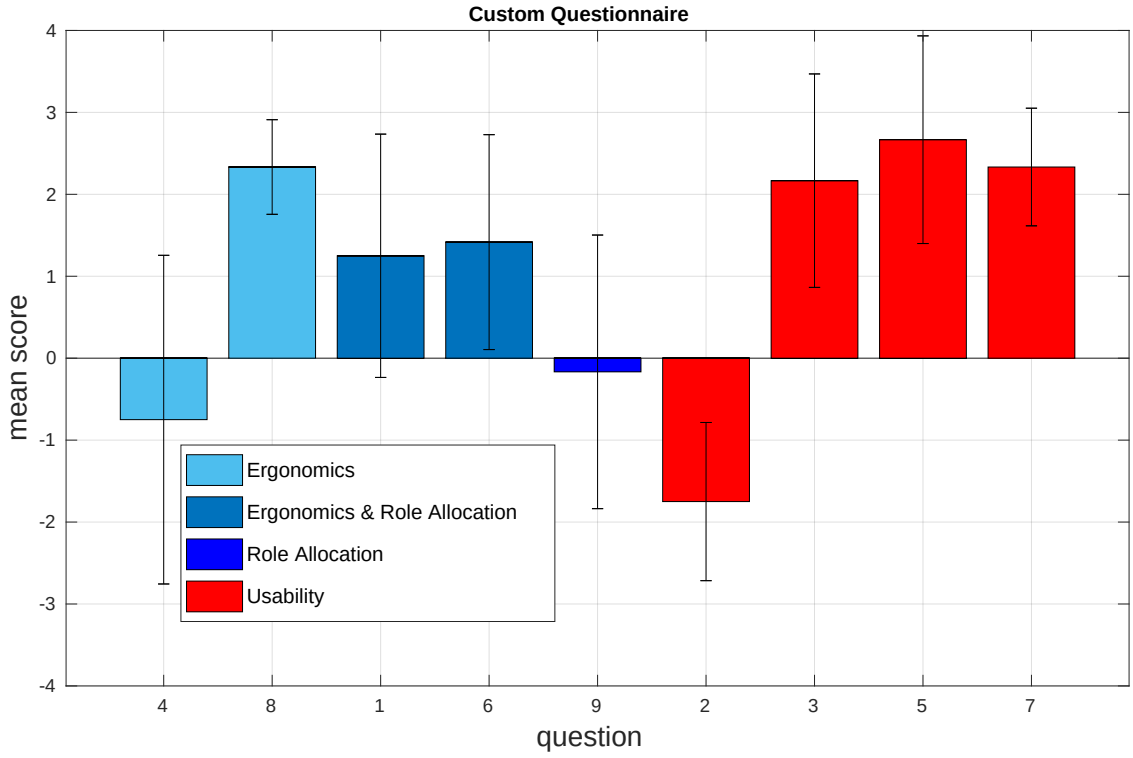


Figure 5.53 Results by collecting subjects' answers to the custom questionnaire. On the x-axis the question number, on the y-axis, the average score. A positive score means a positive answer and vice versa. Questions were designed to evaluate: granted Ergonomics, Role Allocation efficiency and Usability of such a framework.

to inform them about the next task nor considered the information pace too rapid: it is a good outcome, since it means the allocation algorithm computational time does not impede the task flow. Finally, we asked to subjects if they would suggest to employ this framework for this kind of tasks. The answer was positive: they felt to be safe by cooperating with the robot and it did not hinder them in the task accomplishment.

Chapter 6

Conclusions

In this thesis we proposed a method that envisions a dynamic human-robot role allocation based on an online evaluation of the ergonomic risk: by intervening on the roles distribution within the human-robot team, we aim to reduce the human weariness at the end of the collaborative performance.

We tested our framework by arranging an assembly, a very common task in industrial environments. For task decomposition and as allocation model we used an adapted AND/OR Graph (AOG), where the actions that form the task, potentially executed by all the agents, are modeled with different hyper-arcs (section 2.1 and section 2.2). To each hyper-arc, that represents a human action, an ergonomic cost is assigned. Such costs are updated during the whole duration of the task: each of them encodes a predicted physical risk the worker would have by executing the corresponding action, starting from their current conditions. To retrieve such a value, a prediction model is needed (section 3.1). The hyper-arcs costs guide the online AO* search designed to find the optimal path, returning the next action to execute and the allocated agent (section 2.3). According to the simulation results, the computational time necessary to AO* for returning a solution, even with a fairly big AOG size, allows the workers to perform the task without any interruption of the work flow (section 4.2). The outcome of the multi-subject experiment we conducted confirmed that they did not have to wait for the system to inform them about the next action to be performed. A drawback of the implemented AOG to be overcome in the future consists in not providing the possibility of modeling parallelism, i.e. the allocation algorithm cannot return as a solution two (or many) actions to be executed at the same time by the two (or many) agents involved. Another limitation is that the human has not the chance of choosing which action to execute, but he/she submits it to the allocation algorithm. A future work would be to make the workers

"free" as long as their status does not evidence any risk and then oblige them to follow the system instructions.

The costs meaning depends on which human aspect we decide to monitor, which in turn is related to the cooperative task features. To describe the human physical status, kinematics or dynamics indicators are needed: the former evaluates the human body configurations and postures, while the latter takes into account the interaction forces, for instance those with the manipulated tool. On the other hand, to ensure ergonomic HRC in the case the task is mentally stressful, indicators that are able to describe the human cognitive load are more appropriate (section 3.2). Therefore, the designed framework is not coupled with any specific metrics: any risk the human may encounter when he/she executes the task, which can be modelled and predicted can be employed in the proposed framework. Since assembly tasks of lightweight pieces are addressed in this work, to test the functioning of the overall framework we designed a kinematic indicator, named *kinematic wear*, which captures potentially damaging human body configuration. Unlike the traditional indexes, the *kinematic wear* is able to model the history of the human worker's joints wear: when a risk is found, this is due not only to the current joint angles, but also to the past assumed configurations (section 3.3). For the design of such an indicator we have been inspired by the muscle fatigue behavior, that can be modeled as an RC circuit, with charge and discharge trends. Clearly, since we are not dealing with forces, we cannot state that the *kinematic wear* can substitute the information conveyed by a dynamic indicator. However, it may be a good future challenge to find some correlations between the two: in this way we could rely only on a motion-capture system, without using neither sensors for measuring the muscle activity nor force plates for the estimation of the centre of pressure position (section 3.2). As we proved throughout the conducted experiments (section 5.2), each action can be described in terms of *kinematic wear* allowing a distinction among them according to which joint is the most stressed. In this way, it is possible both to efficiently avoid the human to perform damaging actions, and to assign to them the risk-free ones. It would be interesting, in the future, to make this framework able not only to decide on the future actions, but also to inform workers of the risk they are exposed while executing the task.

The reduced number of tested subjects stops us to claim any strong general consideration about the capability of such a framework in understanding the human conditions during the task, and helping workers to end the performance less fatigued and frustrated. Nevertheless, the results we obtained in the multi-subject experiment (section 5.3) look promising: further experiments and studies could render more robust the proposed approach.

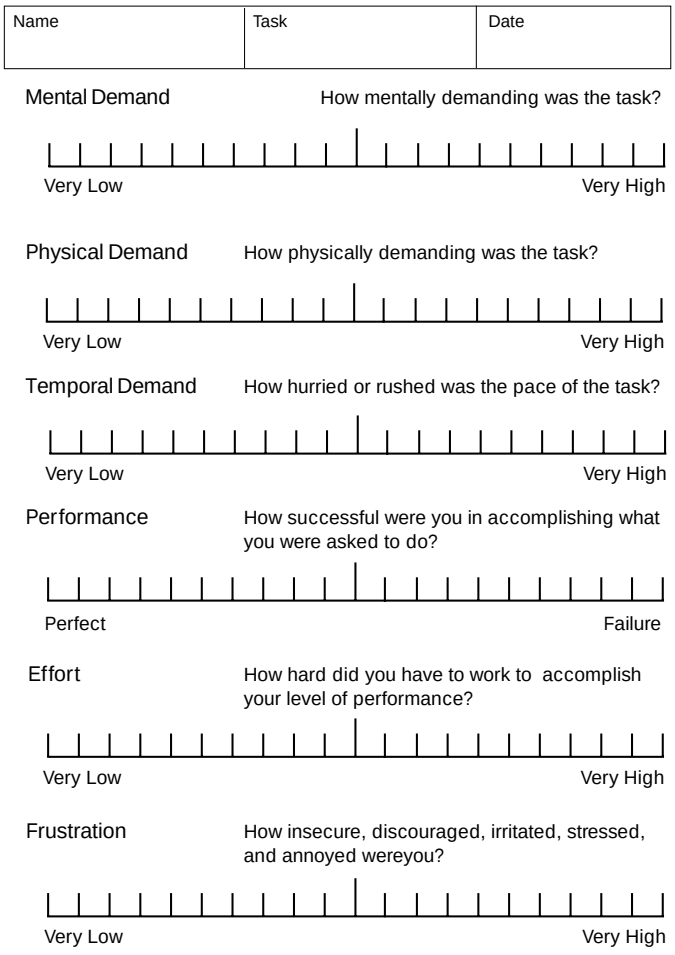
Appendix A

Questionnaires

Figure 8.6

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.



For each couple of items, choose which of the two was more important to <u>your</u> experience of workload in the task that you just performed.

Effort Or Performance	Temporal Demand Or Frustration	Temporal Demand Or Effort	Physical Demand Or Frustration
Performance Or Frustration	Physical Demand Or Temporal Demand	Physical Demand Or Performance	Temporal Demand Or Mental Demand
Frustration Or Effort	Performance Or Mental Demand	Performance Or Temporal Demand	Mental Demand Or Effort
Mental Demand Or Physical Demand	Effort Or Physical Demand	Frustration Or Mental Demand	

Rate the following statements for the HUMAN-ROBOT experiment.

CUSTOM QUESTIONNAIRE

1. I was glad to take a break when the task was allocated to the robot.

2. The robot was hindering me in the task accomplishment.

3. It was easy to understand when I had to take action.

4. When I was tired, the robot was not able to intervene.

5. I feel safe when collaborating with the robot.

6. The robot can choose correctly which action is too demanding for me.

7. I would suggest to use the robot for this kind of tasks.

8. The robot help makes me feel less fatigued at the end of the task.

9 I had to wait for the system to inform me about the next task.

Acronyms

AOG AND/OR Graph. iv, vi, viii, xiii, 1, 7, 12–21, 23, 28, 41–43, 45, 51, 88

HRC Human Robot Collaboration. iv, v, 1–3, 7–13, 19, 25, 38, 41, 78, 80, 84, 89

MRTA Multi-Robot Task-Allocation. 1, 4

NIOSH National Institute for Occupational Safety and Health. 1, 10

OWAS Ovako Working posture Analysing System. 1, 9

REBA Rapid Entire Body Assessment. viii, 1, 9, 10

RULA Rapid Upper Limb Assessment. ix, 1, 9, 13, 34–36, 52, 57, 72

RoM Range of Motion. 1, 34–36, 51, 52, 59, 76

SMEs Small and Medium Enterprises. 1

WISHA Washington Industrial Safety and Health Act. 1, 10

WRMD Work-related musculoskeletal disorders. 1, 8, 9

pHRI physical Human Robot Interaction. 1, 2, 5, 6

References

Abramson, M., Chao, W., and Mittu, R. (2005). Design and evaluation of distributed role allocation algorithms in open environments. Technical report, NAVAL RESEARCH LAB WASHINGTON DC.

Agüero, C., Matellán Olivera, V., Gómez Gómez, V. M., Cañas, J. M., et al. (2006). Switch! dynamic roles exchange among cooperative robots.

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. (2018). Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975.

Akella, P., Peshkin, M., Colgate, E., Wannasuphoprasit, W., Nagesh, N., Wells, J., Holland, S., Pearson, T., and Peacock, B. (1999). Cobots for the automobile assembly line. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 728–733. IEEE.

Albu-Schäffer, A., Ott, C., and Hirzinger, G. (2007). A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The international journal of robotics research*, 26(1):23–39.

Antonsson, E. K. and Mann, R. W. (1985). The frequency content of gait. *Journal of biomechanics*, 18(1):39–47.

Bänziger, T., Kunz, A., and Wegener, K. (2018). Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions. *Journal of Intelligent Manufacturing*, pages 1–14.

Bauer, W., Bender, M., Braun, M., Rally, P., and Scholtz, O. (2016). Lightweight robots in manual assembly—best to start simply. *Examining companies' initial experiences with lightweight robots, Stuttgart*, pages 1–32.

Bestick, A. M., Burden, S. A., Willits, G., Naikal, N., Sastry, S. S., and Bajcsy, R. (2015). Personalized kinematics for human-robot collaborative manipulation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1037–1044. IEEE.

Busch, B., Maeda, G., Mollard, Y., Demangeat, M., and Lopes, M. (2017). Postural optimization for an ergonomic human-robot interaction. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2778–2785. IEEE.

Busch, B., Toussaint, M., and Lopes, M. (2018). Planning ergonomic sequences of actions in human-robot interaction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1916–1923. IEEE.

Campbell, A. and Wu, A. S. (2011). Multi-agent role allocation: issues, approaches, and multiple perspectives. *Autonomous agents and multi-agent systems*, 22(2):317–355.

Casalino, A., Zanchettin, A. M., Piroddi, L., and Rocco, P. (2019). Optimal scheduling of human-robot collaborative assembly operations with time petri nets. *IEEE Transactions on Automation Science and Engineering*.

Chaimowicz, L., Campos, M. F., and Kumar, V. (2002). Dynamic role assignment for cooperative robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 293–298. IEEE.

Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., and Fraisse, P. (2016). Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13.

Damsgaard, M., Rasmussen, J., Christensen, S. T., Surma, E., and De Zee, M. (2006). Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory*, 14(8):1100–1111.

Darvish, K., Simetti, E., Mastrogiovanni, F., and Casalino, G. (2020). A hierarchical architecture for human–robot cooperation processes. *IEEE Transactions on Robotics*, 37(2):567–586.

Darvish, K., Wanderlingh, F., Bruno, B., Simetti, E., Mastrogiovanni, F., and Casalino, G. (2018). Flexible human–robot cooperation models for assisted shop-floor tasks. *Mechatronics*, 51:97–114.

De Mello, L. H. and Sanderson, A. C. (1990). And/or graph representation of assembly plans. *IEEE Transactions on robotics and automation*, 6(2):188–199.

De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A. (2008). An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3):253–270.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., and Thelen, D. G. (2007). Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11):1940–1950.

Dias, M. B., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270.

Doran, J. E. and Michie, D. (1966). Experiments with the graph traverser program. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 294(1437):235–259.

Faber, M., Mertens, A., and Schlick, C. M. (2017). Cognition-enhanced assembly sequence planning for ergonomic and productive human–robot collaboration in self-optimizing assembly cells. *Production Engineering*, 11(2):145–154.

Farinelli, A., Iocchi, L., and Nardi, D. (2004). Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2015–2028.

Fitts, P. M. (1951). Human engineering for an effective air-navigation and traffic-control system.

Fusaro, F., Lamon, E., De Momi, E., and Ajoudani, A. (2021). An integrated dynamic method for allocating roles and planning tasks for mixed human-robot teams. *2021 30th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*.

Gale, D. (1989). *The theory of linear economic models*. University of Chicago press.

Gerkey, B. P. and Matarić, M. J. (2003). On role allocation in robocup. In *Robot Soccer World Cup*, pages 43–53. Springer.

Gerkey, B. P. and Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9):939–954.

Govaerts, R., Tassignon, B., Ghillebert, J., Serrien, B., De Bock, S., Ampe, T., El Makrini, I., Vanderborght, B., Meeusen, R., and De Pauw, K. (2021). Prevalence and incidence of work-related musculoskeletal disorders in secondary industries of 21st century europe: a systematic review and meta-analysis. *BMC musculoskeletal disorders*, 22(1):1–30.

Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. (2009). Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research*, 28(11-12):1507–1527.

Hignett, S. and McAtamney, L. (2000). Rapid entire body assessment (reba). *Applied ergonomics*, 31(2):201–205.

Johannsmeier, L. and Haddadin, S. (2016). A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robotics and Automation Letters*, 2(1):41–48.

Karhu, O., Härkönen, R., Sorvali, P., and Vepsäläinen, P. (1981). Observing working postures in industry: Examples of owas application. *Applied ergonomics*, 12(1):13–17.

Kim, W., Lee, J., Peternel, L., Tsagarakis, N., and Ajoudani, A. (2017a). Anticipatory robot assistance for the prevention of human static joint overloading in human–robot collaboration. *IEEE robotics and automation letters*, 3(1):68–75.

Kim, W., Lee, J., Tsagarakis, N., and Ajoudani, A. (2017b). A real-time and reduced-complexity approach to the detection and monitoring of static joint overloading in humans. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 828–834. IEEE.

Korsah, G. A., Stentz, A., and Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512.

Krieger, M. J. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84.

Lagomarsino, M., Lorenzini, M., De Momi, E., and Ajoudani, A. (2021). An online framework for cognitive load assessment in assembly tasks. *arXiv preprint arXiv:2109.03627*.

Lamon, E., De Franco, A., Peternel, L., and Ajoudani, A. (2019). A capability-aware role allocation approach to industrial assembly tasks. *IEEE Robotics and Automation Letters*, 4(4):3378–3385.

Lennart, L. (1999). System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, 28.

Lorenzini, M., Kim, W., Momi, E. D., and Ajoudani, A. (2019). A new overloading fatigue model for ergonomic risk assessment with application to human-robot collaboration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1962–1968.

Ma, L., Chablat, D., Bennis, F., Zhang, W., and Guillaume, F. (2010). A new muscle fatigue and recovery model and its ergonomics application in human simulation. *Virtual and Physical Prototyping*.

Maurice, P., Padois, V., Measson, Y., and Bidaud, P. (2017). Human-oriented design of collaborative robots. *International Journal of Industrial Ergonomics*, 57:88–102.

Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L., and Sierra, B. (2017). Human–robot collaboration in industrial applications: Safety, interaction and trust. *International Journal of Advanced Robotic Systems*.

McAtamney, L. and Corlett, E. N. (1993). Rula: a survey method for the investigation of work-related upper limb disorders. *Applied ergonomics*, 24(2):91–99.

Merlo, E., Lamon, E., Fusaro, F., Lorenzini, M., Carfì, A., Mastrogiovanni, F., and Ajoudani, A. (2021). Dynamic human-robot role allocation based on human ergonomics risk prediction and robot actions adaptation. *arXiv preprint arXiv:2111.03630*.

Mörtl, A., Lawitzky, M., Kucukyilmaz, A., Sezgin, M., Basdogan, C., and Hirche, S. (2012). The role of roles: Physical cooperation between humans and robots. *The International Journal of Robotics Research*, 31(13):1656–1674.

Pearce, M., Mutlu, B., Shah, J., and Radwin, R. (2018). Optimizing makespan and ergonomics in integrating collaborative robots into manufacturing processes. *IEEE transactions on automation science and engineering*, 15(4):1772–1784.

Peternel, L., Fang, C., Tsagarakis, N., and Ajoudani, A. (2019). A selective muscle fatigue management approach to ergonomic human-robot co-manipulation. *Robotics and Computer-Integrated Manufacturing*, 58:69–79.

Peternel, L., Tsagarakis, N., Caldwell, D., and Ajoudani, A. (2018). Robot adaptation to human physical fatigue in human–robot co-manipulation. *Autonomous Robots*, 42(5):1011–1021.

Punnett, L. and Wegman, D. H. (2004). Work-related musculoskeletal disorders: the epidemiologic evidence and the debate. *Journal of electromyography and kinesiology*, 14(1):13–23.

Rajavenkatanarayanan, A., Nambiappan, H. R., Kyrarini, M., and Makedon, F. (2020). Towards a real-time cognitive load assessment system for industrial human-robot cooperation. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 698–705. IEEE.

Santis, A. D., Siciliano, B., Luca, A. D., and Bicchi, A. (2008). An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3):253 – 270.

Steven Moore, J. and Garg, A. (1995). The strain index: a proposed method to analyze jobs for risk of distal upper extremity disorders. *American Industrial Hygiene Association Journal*, 56(5):443–458.

Urata, J., Hirose, T., Namiki, Y., Nakanishi, Y., Mizuuchi, I., and Inaba, M. (2008). Thermal control of electrical motors for high-power humanoid robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2047–2052.

Van de Perre, G., El Makrini, I., Van Acker, B., Saldien, J., Vergara, C., Pintelon, L., Chemweno, P., Weuts, R., Moons, K., Dewil, R., et al. (2018). Improving productivity and worker conditions in assembly: part 1: a collaborative architecture and task allocation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

van der Spaa, L., Gienger, M., Bates, T., and Kober, J. (2020). Predicting and optimizing ergonomics in physical human-robot cooperation tasks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1799–1805. IEEE.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248 – 266.

Waters, T. R., Putz-Anderson, V., and Garg, A. (1994). Applications manual for the revised niosh lifting equation.

Woolson, R. F. (2007). Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.