2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Towards a Prolonged Productivity in Industry 4.0: A Framework for Fatigue Minimisation in Robot-Robot Co-Manipulation

Edoardo Lamon[1,2], Luka Peternel[1], and Arash Ajoudani[1]

Abstract— **Industry 4.0 envisions the integration of flexible and quickly reconfigurable robotic systems in assembly lines. This has led to the development of light-weight and adaptive collaborative robots with limited power and payload constraints. Hence, repetitive tasks or those that demand high forces may exceed such limits, resulting in robot damage and lost productivity. To address this problem, we propose a novel framework to prolong the lifetime of collaborative robots while guaranteeing the desired level of productivity. To address this, we propose a method that minimises a function of robot fatigue, an index able to model the robot motor usage at the joint level. This index features the desired external force, task duration, hardware parameters, and fatigue history. Moreover, future tasks are considered through fatigue thresholds imposed on specific joints, computed according to the robot safety requirements. We proved the effectiveness of the approach by comparing its results in terms of fatigue and torque with the well-known minimum effort approach. The results showed that our method ensures that the fatigue thresholds are not exceeded.**

I. INTRODUCTION

One of the challenges of Industry 4.0 aims to integrate flexible and reconfigurable manufacturing solutions into the industrial process. Reconfigurable manufacturing systems embody this new tendency where the production line is able to quickly adapt its capacity and functionality based on frequent changes in product demand. In such systems, every work cell layout is designed by the principles of functionality, simplicity of work flow, optimisation of transit time between each workstation, and sequence of tasks.

The above-mentioned solutions often require cooperation and interconnections between the different agents (e.g., humans and robots). In particular, the presence of smart collaborative robots (cobots) opens a possibility to accomplish complex tasks while sharing the workload with each other. In small and medium-size companies the number of robots may be limited and therefore each robot is expected to be able to perform a variety of different tasks while dynamically changing its role with other robots or humans within the same work cell.

The main advantages of robots compared to human workers are high-precision motion and considerable power capacity. Nevertheless, modern versatile work cells tend to pair robots with human workers, which can complement the robots with their superior cognitive capabilities and task understanding. When cobots are working together with humans, we have to consider human safety as one of the priorities [1]. To make working conditions safe for the human from the physical interaction perspective, cobots employ variable impedance control [2], [3] that can facilitate a compliant interaction. Another important safety aspect is the ergonomics of the human partner, which can be addressed by either reduction of human body joint torques [4] or muscle fatigue [5] during human-robot collaboration. Such methods can prevent degradation of human performance and ensure that the tasks are performed in specified time schedules.

Nevertheless, some tasks may also overload the cobot's power or endurance capacity, which would also prevent the tasks to be executed in the desired schedule. Some common physical stress cases include: working in ineffective configurations, exerting high forces to produce the task, carrying heavy loads and using vibrating tools for a prolonged time. These cases induce both mechanical and thermal fatigue to the robot structure/motors and may lead to performance degradation or even damage.

When cooling down the motors alone is not sufficient to prevent the increase of thermal fatigue, the robot can use the redundant degrees of freedom (DOF) to optimise the joint configuration [6] and achieve the task production with less joint effort. The most common approach is the classic minimum effort technique [7] that gives a minimum overall torque in all joints. However, such an approach fails when specific joints need to be offloaded due to the fatigue. To address this issue, a method for fatigue management through the robot joint reconfiguration was recently proposed [8]. This method monitors the current states of motor temperature and redistributes the load from the specific joints that are under a high fatigue to the joints that do not have a considerable fatigue. Nevertheless, the approach in [8] worked only based on the current state of the fatigue and did not have the ability to predict future states, which is an important factor when considering time-constrained multi-task production process.

In this paper, we introduce a method that computes the optimal configuration to accomplish a particular task in a specified time within the desired fatigue limitations. On one hand, the time constraint is dictated by the optimal production process consisting of a sequence of tasks. On the other hand, staying within the fatigue limitations during the current task is crucial to be able to successfully finish the next tasks in the process. The proposed method includes an optimisation approach to find the best solution to execute a certain task by using the fatigue model and specifications about previous and next tasks. As a comparison, the traditional minimum effort solution does not provide the best

[1] HRI[2] Lab, Dept. of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy.
[2] Robotics and Automation group, Dept. of Information Engineering, Universita' degli Studi di Pisa, Pisa, Italy.
edoardo.lamon@iit.it

978-1-5386-7282-2/18/$31.00 ©2018 IEEE

configuration estimate for the stack of tasks, since it does not consider the existing fatigue as a result of previous tasks, nor can it predict the fatigue in future, which can potentially prevent the execution of next tasks.

To validate the proposed method we perform experiments on KUKA LWR4+ (main robot) equipped with Pisa/IIT SoftHand and Franka Emika Panda (assistant robot). We selected object polishing with a polishing machine as the current task of the production process. The optimisation process considers the fatigue from the previous tasks and fatigue limitations concerned with future tasks. While the role of the main robot is to accomplish force-demanding task requirements, the presence of the assistant robot is exploited for improving the system performance in terms of reconfigurability. In the experimental setup, the main robot holds the polishing tool, while the assistant robot holds and moves the object that has to be polished. For these proof-of-concept experiments we only consider the fatigue analysis of the main robot. However, the method can be employed on the assistant robot as well.

II. METHODS

First, we need to describe clearly what is the task of the robot and which quantities describe it. In general, the considered task consists on the application of some force vector $\boldsymbol{f}_{ext}$ in a position $\boldsymbol{x}_f$ (both defined in Cartesian space), and duration T for which it has to be applied. The external force is related to the joint torques of a n-DOFs fixed-base robot through the *joint space dynamic model*:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{ext}, \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint angles vector, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is the vector of Coriolis and centrifugal torques, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ denotes the gravity component and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of input joint torques. The last term, $\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{ext} \in \mathbb{R}^n$, represents the torque due to the action of an external force $\boldsymbol{f}_{ext} \in \mathbb{R}^m$, with $m \leq 6$, projected at joint level by the geometric Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$.

A. Minimum Effort

A classic method based on optimal control computes the robot configuration $\boldsymbol{q}$ according to the minimisation of an index function, called "effort" [7]. If the duration T is fixed, a possible effort ε is defined by

$$\varepsilon(\boldsymbol{\tau}) = \int_0^T \boldsymbol{\tau}^T(t)\boldsymbol{\tau}(t)dt = \int_0^T \sum_{i=1}^n \tau_i^2(t)dt. \tag{2}$$

In our problem, the force $\boldsymbol{f}_{ext}$ should be produced in a point in space $\boldsymbol{x}_f$ that belongs to a certain region $\mathcal{D}$ in Cartesian space, for the whole duration of the task in the same configuration. In this way, the torques needed to produce it become just a function of the current joint configuration ($\boldsymbol{q}$) and are constant over time. For this reason, the joint space model (1) is reduced to the quasi-static relation

$$\boldsymbol{\tau}(\boldsymbol{q}) = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{ext} + \boldsymbol{g}(\boldsymbol{q}), \tag{3}$$

For an established configuration $\boldsymbol{q}$, the effort function is defined as

$$\varepsilon(\boldsymbol{\tau}(\boldsymbol{q})) = T\sum_{i=1}^n \tau_i^2(\boldsymbol{q}). \tag{4}$$

Since T is defined by the production process and is therefore constant, it does not affect the computation of the optimal configuration and the optimisation process is given as

$$\boldsymbol{q}_\varepsilon = \arg\min_{\boldsymbol{q}} \varepsilon(\boldsymbol{\tau}(\boldsymbol{q})) = \arg\min_{\boldsymbol{q}} \sum_{i=1}^n \tau_i^2(\boldsymbol{q}) \tag{5}$$

subject to $\boldsymbol{x}_f \in \mathcal{D}$.

This configuration represents the optimum only with respect to the torques required for the current task. However, this optimisation does not consider the history of the torques from the previous tasks that may have caused a considerable fatigue to specific joints. According to this algorithm the torque load is just distributed through all the joints, even if some motors are more powerful than others. While hardware specifics, like motor strength and cooling capacity, can be included into the optimisation by an additional weighting matrix, the fatigue inherited from the previous tasks and future fatigue trends, as a result of producing the current task, are not considered. Furthermore, in industrial assembly line scenarios, robots are required to produce high forces and change configuration rapidly and repetitively. Consequently, for the purpose of finding a configuration based on the current task, and the history of previous tasks, it is reasonable to solve a different optimisation problem. Such a problem should consider also past and future fatigue, the end-effector external force, the duration of the task production and parameters based on the hardware of the system.

B. Robot Joint Fatigue Model

In order to solve such an optimisation problem, we will introduce the concept of fatigue in a robotic framework. In the case of humans, the fatigue reduces the force production capacity of muscles [9]. Similarly, we can consider the fatigue of a robot when the motors are not able to effectively produce the required force or when the thermal/physical state might damage the hardware. The fatigue model of the robot joint that we devise is based on a first-order dynamic model inspired by [5]. This dynamics also correspond to the dynamics of some other thermal models found in the literature [10]. The differential equation of the proposed model is given as

$$C_i\frac{dV_i(t)}{dt} = \tau_i(t)(1 - V_i(t)) \tag{6}$$

where V_i, C_i, and τ_i are the fatigue level, endurance capacity, and the torque of joint i, respectively. The capacity is a hardware-related parameter that represents the specifics of the motor (i.e. power, cooling, etc.). It basically regulates the slope of the fatigue curve under a given torque; the bigger the capacity is, the longer it takes for the fatigue to take effect. The solution of this first-order differential equation is

$$V_i(T) = 1 - \exp\left(-\frac{\int_{T_0}^{T} \tau_i(t)dt}{C_i}\right) \tag{7}$$

where T_0 and T represent the initial and final time instant of the force generation ($T_0 < T$) and the level of fatigue is defined in the range $[0, 1]$.

C. Minimum Fatigue Configuration

We propose a method that estimates, for each task, the optimal configuration of the robot in which it can produce the desired end-effector force without reaching the predefined fatigue thresholds in different joints. The fatigue thresholds can be dictated by the desired performance and amount/type of future tasks that the robot still needs to produce. For example, if the robot still has to produce several tasks after the current task (without a break), the fatigue thresholds should be set low enough in order to keep some reserve for the future. If the sequence of tasks is defined in advance, the whole multi-task process can be optimised by using the proposed method. Note that since the model is scalar, each joint fatigue level is calculated independently. In a quasi-static case, i.e. fixed configuration vector $\boldsymbol{q}$ throughout force duration (for simplicity $T_0 = 0$ and $T = t$), the model in (7) can be simplified into

$$V_i(t) = 1 - \exp\left(-\frac{\tau_i t}{C_i}\right). \tag{8}$$

The desired end-effector force is mapped to the joint level according to (3). For each robot configuration $\boldsymbol{q}$, the respective joint torque is constant throughout the duration of the force generation and leads to a raise of fatigue level according to (8). Note that for this proof-of-concept case we use a constant end-effector force. Nevertheless, if force is not constant and if its trajectory is known in advance, we can instead use the model in (7) in the proposed optimisation approach. In order to consider the fatigue history of the joint, we have to include the initial condition of the fatigue index in the model used for the current task k. The initial condition of the fatigue index is equal to the fatigue inherited from the previous task $k-1$. The modified model in (8) that includes the fatigue history is defined as

$$V_{k,i}(t) = V_{k-1,i} + (1 - V_{k-1,i})(1 - \exp\left(-\frac{\tau_i t}{C_i}\right)), \tag{9}$$

where $V_{k,i}(t)$ still remains within the range $[0, 1]$.

Since the task duration T is defined by the production process and since we consider a constant end-effector force, $V_{k,i}$ can be regarded only as a function of joint configuration $\boldsymbol{q}$. Such a function can then be used in the optimisation process that searches for the optimal joint configuration. For the optimisation, we define a cost function that penalises the exceeding of fatigue thresholds as

$$h(\boldsymbol{q}) = \sum_{i=1}^{n} V_i(\boldsymbol{q}) + \alpha P_i, \tag{10}$$

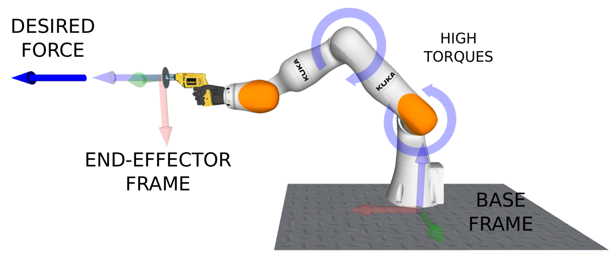

Fig. 1. Orientation of the tool axis with respect to the end-effector frame and to the base frame. In the current configuration, joint 2 and 4 are mainly exploited to obtain the desired force.

where P_i is the function that indicates when the threshold is exceeded and is defined as

$$P_i = \begin{cases} 0, & \text{if } V_i(\boldsymbol{q}) < V_{th,i} \\ 1, & \text{if } V_i(\boldsymbol{q}) \geq V_{th,i} \end{cases}. \tag{11}$$

Since functions (10) and (11) are the same for all tasks in the sequence (with different parameters of course), the task dependency notation k is dropped. The value of α is a weight that sets the amount of penalty. A general rule is that it should be much higher than the sum of fatigues of all joints. Since $\max V_i(\boldsymbol{q}) = 1$ and there are n joints, we should adhere to $\alpha \gg n$.

At this point, it is straightforward to define the process that gives us the optimal configuration as

$$\boldsymbol{q}_{opt} = \arg\min_{\boldsymbol{q}} h(\boldsymbol{q}), \tag{12}$$

subject to constraint $\boldsymbol{x}_f \in \mathcal{D}$. The force application point $\boldsymbol{x}_f$ that corresponds to the position of end-effector of the robot in Cartesian space is also one of the results of the optimisation.

Since we want to search through different directions of the end-effector force in order to find the minimum fatigue configuration, we do not set any Cartesian orientation constraint in the optimisation. To be able to produce the task, the desired force should naturally be aligned with the tool/end-effector (see Fig. 1). The desired force $\boldsymbol{f}_{ext}$ expressed in the end-effector frame is transformed into the base frame by

$$\boldsymbol{f}_{ext}^b(\boldsymbol{q}) = \boldsymbol{R}_e^b(\boldsymbol{q}) \boldsymbol{f}_{ext}^e. \tag{13}$$

In case the tool axis is pointing in the z-axis of the end-effector frame, the desired force $\boldsymbol{f}_{ext}^e$ would be given as $f_r \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, where f_r is a scalar that represents the magnitude of the reference force.

The torque calculation in (3) is therefore updated as

$$\boldsymbol{\tau}(\boldsymbol{q}) = \boldsymbol{J}^T(\boldsymbol{q}) \boldsymbol{R}_e^b(\boldsymbol{q}) \boldsymbol{f}_{ext}^e + \boldsymbol{g}(\boldsymbol{q}), \tag{14}$$

where $\boldsymbol{R}_e^b$ is the transformation matrix from end-effector frame to base frame. This approach keeps the tool axis aligned with the force produced by the robot end-effector.

The cost in (10) $h(\boldsymbol{q})$ is dependent on $V(\cdot)$ through joint torques $\boldsymbol{\tau}$ in (14). To minimise the cost, the optimisation in (12) can reduce the torque coming from the gravity or

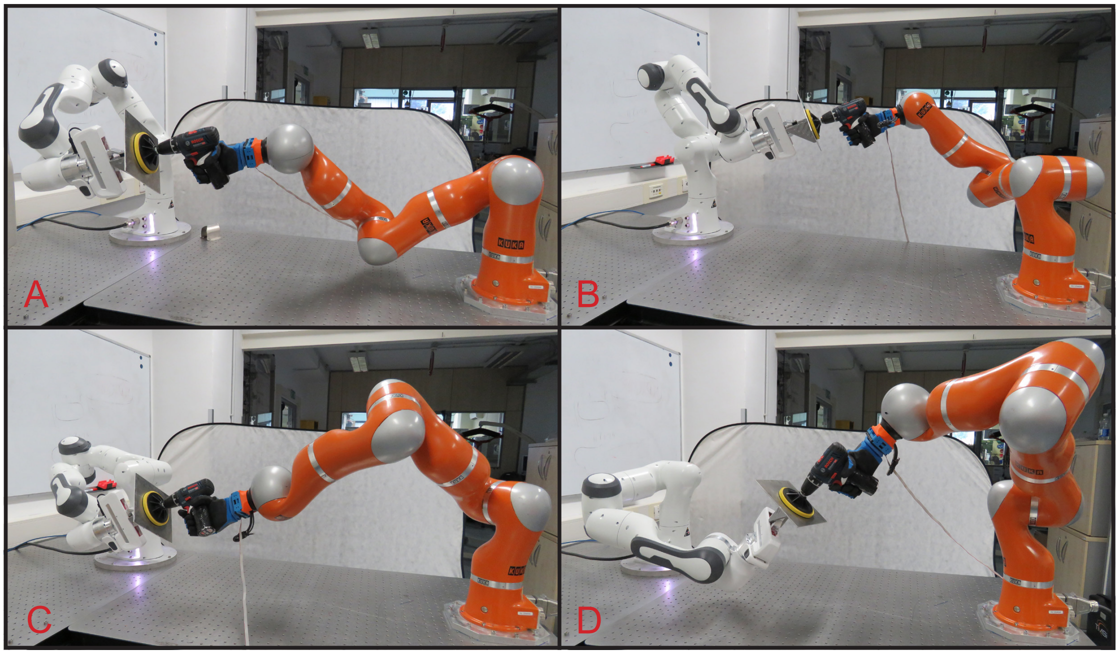

Fig. 2. Different configurations of the KUKA LWR4+ related the polishing task. Fig. 2A: minimum effort and $w_{th} = 0.1$, Fig. 2B: minimum fatigue and $w_{th} = 0.1$, Fig. 2C: minimum effort and and $w_{th} = 0.05$, Fig. 2D: minimum fatigue and $w_{th} = 0.05$. While in the minimum effort configurations (Fig. 2A and 2C) the end-effector force production involves joint 2 and 4 considerably, the minimum fatigue configurations (Fig. 2B and 2D) involve other joints more (joint 1 and joint 3, respectively).

the torque coming from the task force production. Note that mathematically the force at the end-effector can be reduced by moving the robot into singular configurations, where the Jacobian matrix is rank-deficient. These configurations are fine when some external force has to be only sustained by the robot. However, such configurations are not desirable when the robot has to perform manipulation tasks and when itself has to produce the end-effector force on some external object. In fact, the end-effector forces $\boldsymbol{f}_{ext} \in \mathcal{N}(J^T)$, where $\mathcal{N}(J^T)$ is the null subspace of J^T, are compensated by the reaction forces of the constraints of the mechanical structure [11]. This means that in this case the force cannot be actively produced, but just sustained.

In order to avoid the singular configurations and maintain a decent degree of manipulability, we use the manipulability measure [12] defined as

$$w(\boldsymbol{q}) = \sqrt{\det(J(\boldsymbol{q})J^T(\boldsymbol{q}))}, \tag{15}$$

where $w(\boldsymbol{q})$ represents a scalar index that is joint configuration dependent. To ensure a reasonable manipulability in the optimised configuration we modify the cost function in (10) as

$$h(\boldsymbol{q}) = \sum_{i=1}^{n} (V_i(\boldsymbol{q}) + \alpha P_i) + \beta \bar{P}, \tag{16}$$

where P_i is defined as before and $\bar{P}$ is the indicator function

$$\bar{P} = \begin{cases} 1, & \text{if } w(\boldsymbol{q}) < w_{th} \\ 0, & \text{if } w(\boldsymbol{q}) \geq w_{th} \end{cases} \tag{17}$$

with $n < \beta < \alpha$ in order to ensure that the fatigue minimisation has a priority over the manipulability optimisation.

III. EXPERIMENTS

In the experiments, we focused on an industrial scenario that requires adaptability and reconfigurability characteristics to complete different tasks in stack similarly to assembly line production. The experimental setup involved two robots performing a sequence of tasks. For this proof-of-concept experiments, we physically performed only one task, i.e. object polishing (see Fig. 2). The previous tasks were represented by fatigue history parameters, while the future tasks were represented by fatigue threshold parameters.

The main robot (KUKA LWR) held the polishing machine, while the assistant robot (Franka Emika) held the object that had to be polished. The main and the assistant robot had to move their end-effectors to the optimised position in Cartesian space in order to obtain a collaborative co-manipulation. The two robot manipulators shared the workspace that we defined by a Cartesian end-effector constraint and was equal to a sphere of 20 cm of radius, centred in $\boldsymbol{x}^b_{\mathcal{D}} = \begin{bmatrix} 0.486 & 0 & 0.317 \end{bmatrix}$ expressed with respect to the base frame of the main robot.

The software architecture for the main robot consists of a Real-Time (RT) platform for control and software middleware, XBotCore (*Cross-Bot-Core*) [13]. It provides, among other features, a simple API for kinematic and dynamic computations (*Model Interface*) in different robots and it allows the re-usability of the code thanks to the Robot Hardware Abstraction Layer (R-HAL) [14]. The input torques for the main robot are computed according to the hybrid force/impedance controller proposed in [8], where robot redundancy was exploited in the same way to ensure the desired joint behavior.

TABLE I

VALUE OF PARAMETERS USED IN THE EXPERIMENTAL SETUP

Parameters	Value	Description
$\lVert \boldsymbol{f}_{ext}^{e} \rVert$	200	Magnitude of task force $[N]$
C_i	10000	Capacity of joint i $[N \cdot m \cdot s]$
T	60	Duration of the task $[s]$
$r_{\mathcal{D}}$	0.2	Radius of Cartesian sphere constraint $[m]$
$V_{th,i}$	0.6	Fatigue threshold of joint i
α	9999	Fatigue penalty
$w_{th,1}$	0.1	Manipulability threshold of trial 1
$w_{th,2}$	0.08	Manipulability threshold of trial 2
$w_{th,3}$	0.05	Manipulability threshold of trial 3
β	999	Manipulation penalty

TABLE II

COMPARISON OF CONFIGURATION COMPUTED BY MINIMUM EFFORT AND BY MINIMUM FATIGUE.

w_{th}	Estimate	Joint configuration (rad)
0.1	$\boldsymbol{q}_{\varepsilon}$	$\begin{bmatrix}-0.32 & -2.09 & 2.64 & 1.40 & 0.16 & 0.93 & 0\end{bmatrix}^T$
	$\boldsymbol{q}_{opt}$	$\begin{bmatrix}-0.82 & -1.74 & 2.31 & 1.63 & -1.65 & -1.05 & 0\end{bmatrix}^T$
0.08	$\boldsymbol{q}_{\varepsilon}$	$\begin{bmatrix}-0.66 & -1.63 & 1.65 & 1.40 & 0.49 & 0.70 & 0\end{bmatrix}^T$
	$\boldsymbol{q}_{opt}$	$\begin{bmatrix}0.66 & -1.05 & -1.32 & 1.51 & -2.97 & 0.12 & 0\end{bmatrix}^T$
0.05	$\boldsymbol{q}_{\varepsilon}$	$\begin{bmatrix}0.49 & -1.05 & -0.82 & 1.40 & -0.49 & 0.35 & 0\end{bmatrix}^T$
	$\boldsymbol{q}_{opt}$	$\begin{bmatrix}-1.81 & 0.93 & -1.98 & -2.09 & 1.48 & 1.05 & 0\end{bmatrix}^T$

To prove the effectiveness of the proposed method compared to existing methods, we focus on a particular situation. Let us suppose that the manipulator has already accomplished some tasks in the sequence. Because of that, the robot joints have reached a certain level of fatigue, and some of them are already close to the thresholds dictated by the next task. In the experimental setup, we set the fatigue threshold V_{th} of the next task for all the joints to the 60% of the maximum, which corresponds to 0.6, while the history of fatigue was set to

$$\boldsymbol{V}_{past} = \begin{bmatrix}0.2 & 0.59 & 0.2 & 0.59 & 0.2 & 0.2 & 0\end{bmatrix}^T.$$

We assumed that the previous tasks in the sequence used the second and fourth joints considerable more than the rest. Therefore, the fatigue of the second and the fourth joint were already really close to the threshold.

In the experiments we examined cases with three different manipulability thresholds: 0.1, 0.08 and 0.05, that corresponds to the 70%, 55% and 35% of the maximum manipulability[1]. The values of all parameters used in the experimental setup are summarised in Tab. I.

The optimal robot configurations computed by the proposed method and by the classic minimum effort are displayed in Tab. II (also see Fig. 2 for photos during the experiments). For each configuration, we estimated the related fatigue by the fatigue model (9). It is obvious from the Fig. 3 that for all three manipulability threshold cases the minimum effort method fails to keep the fatigue below the thresholds. On the other hand, our method is able to redistribute the joint torque in such a way that the fatigue indexes satisfy the predefined requirements. Basically, the proposed method accomplished the minimisation of the penalty function $h(\boldsymbol{q})$ by exploiting the less fatigued joints. On the other hand, the minimum effort method simply tries to obtain an overall minimum joint torque in all joints. The joint torques for both methods in all cases are displayed in Fig. 4. The proposed optimisation method could redistribute the load between different joints by either avoiding the load from gravity or changing the direction end-effector task force with respect to joint configuration. In the latter case, the method can manipulate the moment arms between end-effector force and joints. If we want to make a certain joint uninvolved in the end-effector force production, we can align its motor axis with the direction of the force[2]. For instance, this is well illustrated in Fig. 2B that shows the optimisation result of the proposed method. In this case, the axes of the second and fourth joints are almost aligned with the end-effector force direction.

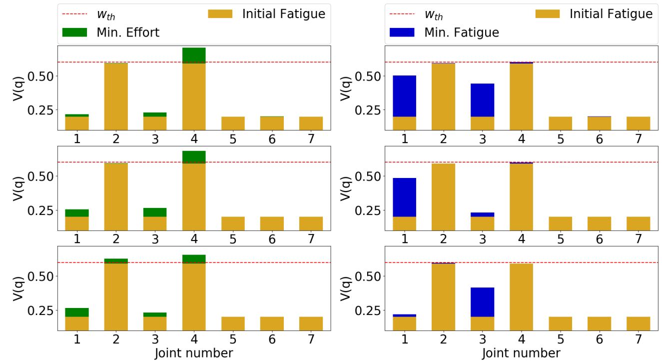

Fig. 3. Different levels of fatigue estimated for the current task computed for the three different manipulability thresholds $w_{th} = 0.1$, $w_{th} = 0.08$, and $w_{th} = 0.05$ (see Tab. I). Yellow bars indicate the fatigue inherited from the previous tasks. Green and blue bars show the fatigue gained during the current task for minimum effort method and our method, respectively. Red lines indicate a fatigue threshold of the current task.

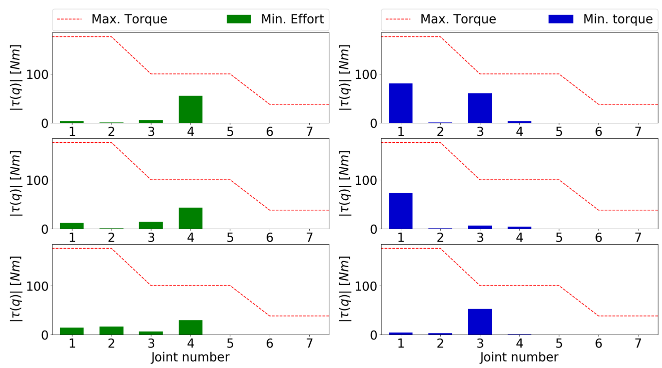

Fig. 4. Joint torques of estimated configurations, computed for the three different manipulability thresholds $w_{th} = 0.1$, $w_{th} = 0.08$, and $w_{th} = 0.05$ (see Tab. I). Red lines indicate the joint torque limit as specified by the robot's technical manual.

[1]We pre-calculated the maximum manipulability within the given end-effector position constraint, which was 0.14. The average manipulability inside the constraint was 0.08.

[2]Since the axis of the last joint is always aligned with the tool axis (see (14)), the rotation of the last joint does not affect the optimisation.

IV. DISCUSSION

The proposed method offers different advantages with respect to the existing methods. The most important is that it computes, if feasible, configurations of the robot for the task production that do not exceed the defined fatigue thresholds. In industrial scenarios, this means that the robot can produce the task effectively by minimising the possibility of having to take a break due to the fatigue and avoid the risk of hardware damage. The optimal configuration for a given task depends on the sequence of tasks in the production process. This means that the optimisation considers past and future events that are defined by various parameters, such as existing fatigue, fatigue thresholds, task duration, etc.

Each fatigue value, indeed, is decoupled from the others; in this way, it is possible to control the fatigue dynamics of each joint individually. One should set different fatigue thresholds and capacity depending on the particular tasks and hardware. For instance, the torque limits and cooling capacity for most robots are not equal for every joint. The joint capacity, indeed, regulates the rate of fatigue growth and because of that more powerful or motors with better cooling systems should be given higher capacity values.

Another advantage of the proposed approach is flexibility. First, it is possible to exploit different fatigue models. In this work we used a torque-based fatigue model. However, the fatigue model can easily be substituted with other models (current-based, temperature-based, etc.). Second, the framework, that has been tested in an industrial scenario with manipulators, can also be applied to different setups, i.e. robots and tasks. For instance, the method could be used, in similar fashion, to compute the optimal joint configuration of a humanoid robot. In this particular case, ground reaction forces should be considered in equation (3). Moreover, since the model has been validated for human fatigue as well, the whole framework could be applied also to humans in human-robot collaborative tasks. In this way, not only robot configuration but also human posture could be optimised.

On the other hand, one of the drawbacks of the proposed method is the presence of several numerical parameters that condition the result of the optimisation. These parameters, such as joint capacity, fatigue thresholds, and manipulability thresholds, basically reflect the system hardware and tasks. Without a proper knowledge of the hardware and tasks, the strength of the proposed method can be limited.

Another potential drawback of the proposed method is that the algorithm has been developed to estimate the configuration offline, during the task planning phase. The optimisation has nonlinear terms and the search of the minimum is global, which makes it time-consuming. Nevertheless, the industrial processes are usually known in advance and because of that the optimisation can be performed previously.

Lastly, in this proof-of-concept study, we considered only fatigue models without the recovery. A possible extension of the current model consists in adding a recovery model, that can be found in the literature [9], [5]. The recovery will lower the fatigue level of the relaxed joints, i.e. the joints whose torque is below a certain level. Moreover, the effectiveness of method could further be tested on humanoid robots and humans. With these models, the optimisation could exploit a larger number of redundant degrees of freedom in order to achieve joint offloading.

REFERENCES

[1] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schffer, "On making robots understand safety: Embedding injury knowledge into control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.

[2] N. Hogan, "Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications," *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, Mar. 1985.

[3] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *Int. J. Rob. Res.*, vol. 26, no. 1, pp. 23–39, Jan. 2007.

[4] W. Kim, J. Lee, L. Peternel, N. Tsagarakis, and A. Ajoudani, "Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 68–75, 2018.

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

[6] V. Potkonjak, D. Kostic, M. Rasic, and G. S. Dordaevic, "Motion in human and machine: A virtual fatigue approach," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 32, no. 5, pp. 582–595, Sep 2002.

[7] L. W. Neustadt, "Minimum effort control systems," *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 1, no. 1, pp. 16–31, 1962.

[8] L. Peternel, N. Tsagarakis, and A. Ajoudani, "A method for robot motor fatigue management in physical interaction and human-robot collaboration tasks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, p. in press.

[9] L. Ma, D. Chablat, F. Bennis, W. Zhang, and F. Guillaume, "A new muscle fatigue and recovery model and its ergonomics application in human simulation," *Virtual and Physical Prototyping*, vol. 5, no. 3, pp. 123–137, 2010.

[10] J. Urata, T. Hirose, Y. Namiki, Y. Nakanishi, I. Mizuuchi, and M. Inaba, "Thermal control of electrical motors for high-power humanoid robots," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 2047–2052.

[11] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[12] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[13] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A real-time cross-robot software platform," in *Robotic Computing (IRC), IEEE International Conference on*. IEEE, 2017, pp. 77–80.

[14] G. F. Rigano, L. Muratore, A. Laurenzi, E. M. Hoffman, and N. G. Tsagarakis, "A mixed real-time robot hardware abstraction layer (r-hal)," *Encyclopedia with Semantic Computing and Robotic Intelligence*, 2018.