# Planning Ergonomic Sequences of Actions in Human-Robot Interaction

Baptiste Busch[1] and Marc Toussaint[2] and Manuel Lopes[3]

***Abstract*— In this paper, we define the problem of human-robot collaboration as a combined task and motion planning problem which is extended to the multi-agent case (human and robot). Our proposed approach allows us to explicitly take into account ergonomic cost, synchrony and concurrency of behavior in an optimization formulation. We show simulated results as well as an experiment with a real robot combined with a user study. Results show that optimizing over a sequence of actions leads to more ergonomic situations.**

## I. Introduction

As progress is made in the field of Human-Robot Interaction (HRI), the complexity and diversity of tasks humans and robots can accomplish together increases. Within the manufacturing field, efficiency, especially time efficiency, is a crucial expectation. This creates a desire to form highly effective human-robot teams that combines strengths and abilities of both the robot and its human partner [1], [2], while at the same time caring for the well-being of the people working with robots. Compared to a fully automated assembly line, a robot and human worker team offers flexibility and adaptability to changing tasks [3], [4]. This last point is particularly important for small assembly lines of customized products.

“Musculoskeletal disorders” (MSDs) are the single largest category of work-related diseases in many industrial countries [5]. If our goal is to bring human-robot cooperation in industrial workplaces, it should not be at the expense of increasing the MSD rate among industrial workers. Robotics solutions can be deployed to assist people with already diagnosed MSDs [6], but preventive measures need also to be considered. Ergonomic research offers a large variety of postural assessment techniques [7], that can be used as indexes to evaluate and improve the ergonomics of the interactions [8].

Task allocation and coordination between humans and robots is an extensively studied topic in HRI [9], [10]. Some research already includes ergonomic factors in their motion planning algorithms [11], [12], while pre-defining the task allocation beforehand. Planning ergonomic actions is often limited to optimize individual actions [8], [13] rather than performing the optimization over a full sequence. We believe that optimizing simultaneously task allocation and motion while taking into account ergonomic aspects would improve the efficiency of the team and their acceptance and adoption in the real world. For example, if the goal is for the human to grasp a screwdriver lying on a far table, the robot would need to grasp it first and move it to a closer and accessible table. Ergonomic reasoning would benefit the efficiency of the team as it would assign burdening tasks to the robot, releasing humans from excessive muscular overload.

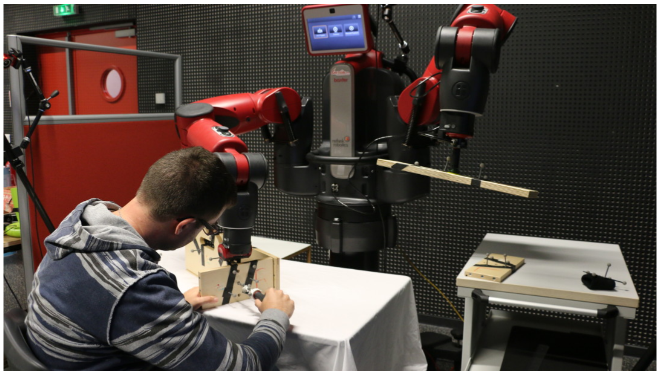


Fig. 1. Toolbox assembly with the help of Baxter robot. The robot hands over parts of the toolbox and hold them for the human coworker to screw them together.

In this paper, we take the approach that ergonomic reasoning can be framed as a combined task and motion planning problem [14], [15], [16], which is extended to the multi-agent case (human and robot) and allows us to integrate an ergonomic objective in an optimization formulation. [9] proposed an optimization-based extension for the multi-agent setting. However, this approach requires all objectives and constraints to be differentiable. The Rapid Entire Body Assessment technique (REBA) to score body postures [17] has previously been applied to select between posture alternatives in HRI [8] and is a widely accepted as a ergonomic measure. However, REBA is a discrete and non-differentiable table of scores (detailed in Section II).

We propose a differentiable surrogate of the REBA score (dREBA) to be integrated in a combined task allocation and motion planning problem. By means of simulated results and a user study on a toolbox assembly collaborative task illustrated in Fig. 1, we show that the introduced dREBA can be used as an enabler to optimize for the original REBA score. We also show that optimizing simultaneously for task allocation and motion planning leads to more ergonomic situations and safer human postures. In this study we have limited the assembly task to the first steps of the assembly

This work was supported by national funds through Fundação para Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 and by the EU FP7-ICT project 3rdHand under grant agreement no 610878.

[1] FLOWERS Team an Inria BSO and ENSTA Paritech joint laboratory, Talence, France. baptiste.busch@inria.fr

[2] Machine Learning and Robotics Lab, University of Stuttgart, Germany. marc.toussaint@informatik.uni-stuttgart.de

[3] INESC-ID, Instituto Superior Técnico, Universide de Lisboa, Portugal. manuel.lopes@tecnico.ulisboa.pt

process, i.e the robot hands over the assembly parts composed of the handle and one side of the toolbox to the human coworker for him or her to screw them together. We argue that a longer assembly of the full toolbox would not provide additionnal benefits and that this shorter task is already enough to observe the effects of our method.

We present the approximation of the REBA cost in Section II and apply the planning technique to two experiments, one in a simulated environment including a Baxter robot and a second one with the real robot as illustrated in Fig. 1.

Results and discussions of the simulated experiment and with the real robot are detailed in Section III and IV respectively and show that planning a full sequence of action for ergonomics is more beneficial than optimizing actions independently.

## II. Ergonomic Task Allocation and Path Optimization

In this section we present the LGP formalism used to simultaneously optimize for task allocation and motion planning. We also introduce the REBA assessment technique we have considered as index for ergonomics and the proposed dREBA surrogate.

### A. LGP formulation and solver

A Logic-Geometric Program (LGP) is an optimization problem over both, a symbolic sequence of actions and a (piece-wise) smooth motion of a system [18]. The logic (e.g., STRIPS-like rules) defines which sequence of actions is feasible; and a sequence of actions defines which geometric constraints the motion has to fulfill. Both are optimized jointly with respect to a cost function, typically control costs of the resulting motion. Formally, an LGP is of the form

$$
\begin{aligned}
\min_{x,s_{1:K}} \quad & \int_0^T c(x(t),\dot{x}(t),\ddot{x}(t))\ dt + f_{\text{goal}}(x(T)) \\
\text{s.t.} \quad & \forall_{t\in[0,T]}\ h_{\text{path}}(x(t),\dot{x}(t)\mid s_{k(t)}) = 0 \\
& \forall_{t\in[0,T]}\ g_{\text{path}}(x(t),\dot{x}(t)\mid s_{k(t)}) \le 0 \\
& \forall_{k=1}^{K}\ h_{\text{switch}}(x(t_k)\mid s_k, s_{k\text{-}1}) = 0 \\
& \forall_{k=1}^{K}\ g_{\text{switch}}(x(t_k)\mid s_k, s_{k\text{-}1}) \le 0 \\
& \forall_{k=1:K}\ s_k \in \text{succ}(s_{k\text{-}1}) \\
& s_K \models \mathfrak{g}_{\text{goal}}\ ,
\end{aligned}
$$

where $x(t)$ is the path, $s_{1:K}$ the symbolic state sequence, $t_{1:k}$ the time points of symbolic state transitions, $h_*, g_*$ the constraints on the motion given $s_{1:K}$, and $\mathfrak{g}_{\text{goal}}, f_{\text{goal}}$ define goal constraints and objectives.

Solving an LGP is hard; most existing combined task and motion planning solvers employ sampling-based methods [14], [15], [16]. As we aim for the optimization w.r.t. an ergonomic measure we adopt the optimization-based LGP formulation. In [9] a multi-bound tree search (MBTS) method is proposed to approximate LGP solutions. MBTS uses several bounds of the full LGP that are themselves optimization problems but with less constraints or only over sub-paths and therefore much faster to evaluate. These bounds are used to prune subtrees when they are found infeasible and prioritize search for symbolic sequences.

To apply the LGP framework to our problem we need to formulate

1) the joint symbolic decision space for the specific human-robot cooperation task,
2) the constraints on the human or robot pose depending on such decisions,
3) the cost function for both, the human and the robot.

Concerning the symbolic decisions, we use STRIPS-like operators to define the action space depending on the specific task. For example, for the first task of our experiment section we will define two possible actions, `grasp` and `place`:

- `grasp(t,e,o)`
  - *description*: at time `t`, the end-effector `e` (robot or agent) grasps the object `o`
  - *precondition*: `free(e) on_table(o)`
  - *effect*: `grasped(e,o) ¬free(e) ¬on_table(o)`
- `place(t,e,o,p)`
  - *description*: at time `t`, the end-effector `e` (robot or agent) places the object `o` on table `p`
  - *precondition*: `grasped(e,o)`
  - *effect*: `placed(o,p) ¬grasped(e,o) free(e) on_table(o)`

These operators define feasible symbolic successor states $\text{succ}(s_{k\text{-}1})$, where $s_k$ is a first-order logic state initialized as `placed(screwdriver, table_left)`. Note that the variable `e` refers to possible end-effectors—in this way these operators define the decision space for all “agents” (end-effectors of human and robot).

The decisions imply geometric constraints: A `grasp` implies a kinematic switch of attaching the object `o` to the end-effector `e`, with a respective equality constraint $h_{\text{switch}}(...)$ ensuring that the object does not jump. A `place` action implies a kinematic switch of attaching the object `o` to the table `p`, detaching from the end-effector `e`, both constrained geometric by not having a jump in the object pose.

The symbolic goal $\mathfrak{g}_{\text{goal}}$ is `grasped(human_right_hand, screwdriver)`.

Concerning the cost function, we optimize for maximal ergonomics in the side of the human, as described below. Additionally we enforce smooth motions by optimizing for the sum-of-squared accelerations of both, the robot and human motion.

### B. The REBA Score as Ergonomic Measure

The REBA assessment technique [17] was initially introduced for pen and paper assessment of industrial workflows. It describes ergonomic preferences with a table as follows. First, a *part table* assigns a score to each parts of the body (trunk, neck, upper arms, . . . ), based on their inclinations at the time of the assessment, as illustrated in Fig. 2. Then, an overall score is calculated from *correspondence tables* that take into account the importance of the body segment. Indeed, a score of 3 for the trunk represents a greater risk than the same score for the upper arm. First, body segments

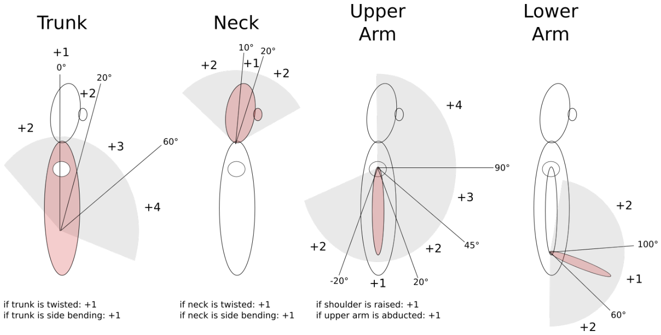


Fig. 2. Example of the REBA calculation for the trunk, neck, upper arm and lower arms.

are divided into groups A (trunk, legs, and neck) and B (upper arm, lower arm, and wrist) and a score is assigned to each group. Finally, the REBA score is calculated from the last table based on the score of both groups.

With our work we want to adopt such meaningful industrial standards, but make them applicable in our optimization framework. This requires us to propose a differentiable version of the REBA cost function, as described in the following.

### C. Differentiable REBA (dREBA)

In its original formulation, the REBA score is a stepwise linear function. We propose to fit a differentiable model to the REBA score, dREBA, and use this as cost function in the LGP formulation. We model dREBA as a sum of weighted polynomial functions,

$$\mathrm{dREBA}(q,t) = \delta_{payload} + \sum_{i=1}^{n} w_i Q_i(q_i,t) \ , \tag{1}$$

where $n$ is the number of joints considered in the REBA techniques and $Q_i(q_i,t)$ is a 2nd order polynomial of the joint $i$ as a function of the joint value $q_i$ at time $t$. First, the coefficients of each polynomial $Q_i(q_i,t)$ are calculated to minimize the squared error to the the *part tables* of the original REBA score table, for each joint separately. Then, the weights $w_i$ are learned from the total REBA score, including the *corresponding tables*. For this, the squared error of our surrogate and the original REBA score is minimized on a set of sample body configuration: We define a set of random poses, ensuring each class of the original REBA cost (from 1 to 10) are evenly represented.

Computing the gradients of the fitted dREBA cost function (1) is straightforward,

$$\nabla_{q_i} \mathrm{dREBA}(q,t) = w_i \nabla_{q_i} Q_i(q_i,t) \ . \tag{2}$$

Table I shows the root mean-square error (RMSE, bottom row) of the dREBA approximation for each of the 10 REBA classes (upper row). As we can see, the RMSE is relatively high for classes of high REBA cost but acceptable for those of low REBA cost.

In our experiments, we use dREBA to enable optimization in the LGP framework. However, as the original REBA score is the accepted standard we will report all the scores w.r.t. the original REBA score; dREBA is only the enabler to optimize for the REBA score.

TABLE I

RMSE FOR EACH REBA CLASSES

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| 0.72 | 0.91 | 0.86 | 1.08 | 1.02 | 1.72 | 1.86 | 2.18 | 2.18 | 1.50 |

### D. Payload cost calculation

The term $\delta_{payload}$ in (1) corresponds to the cost for carrying an object. In the original method, it is calculated using a decision tree as follow,

$$\begin{aligned} if\ load < 5kg: &\ \ \delta_{payload} = 0 \\ if\ load\ between\ 5\ to\ 10kg: &\ \ \delta_{payload} = 1\ , \\ if\ load > 10kg: &\ \ \delta_{payload} = 2 \end{aligned} \tag{3}$$

where *load* corresponds to the weight of the carried objects. Although convenient for pen and paper assessment, as it is simpler to work with integer values, the function of (3) means that carrying objects lighter than $5kg$ is costless and transitions are sharp. As we are working in the continuous domain, we find it more suitable to also consider a continuous function to represent the payload coast. Therefore, we propose the following linear cost term for load,

$$\delta_{payload} = \frac{load}{\omega_{load}} \ , \tag{4}$$

where $\omega_{load} = 5kg$. The function defined in (4) is strictly equivalent to the original calculation in (3) at the boundaries fixed by the REBA method but has the advantages to set a cost even for light weight objects.

## III. Simulation Experiment

We apply the method detailed in Section II in a simulation experiment where the robot has to place the screwdriver on a table for the human to grasp as illustrated in Fig. 3. There are three tables of random heights ranging from 0.6 to 1.6 meters located in front of the human. In the LGP framework, agents and objects positions are variables to be optimized. The environement is, however, considered fixed but can be used for certain actions, e.g. the planner can choose a table to put the screwdriver on.

Our hypothesis is that depending whether the ergonomic cost function introduced in Section II is turned on or off, the choice of table for placing the screwdriver will differ and the overall ergonomic cost will be reduced even if it increases locally. Moreover, the choice of table should impact the human posture cost.

The actions available to the human and robot are `grasp` and `place` as described in Sec. II-A. The MBTS solver return a sequences of actions, optimal in terms of the provided cost functions as in the example given in Fig. 4.

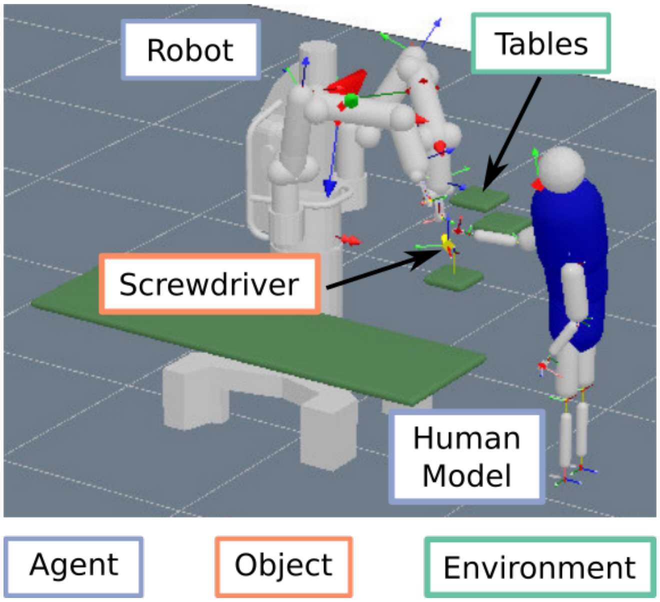


Fig. 3. Simulated view of the experiment. The robot places the screwdriver on one of the three tables for the human to grasp it. Tables heights are randomly set at each run of the experiment.

### A. Experimental setup and hypotheses

To validate the benefits of including the ergonomic cost function in the LGP framework, we have considered three experimental conditions,

1) the *non-ergonomic* condition with ergonomic costs turned off,
2) the *ergonomic* condition with ergonomic costs on,
3) and *re-optimized* condition where the path is optimized with ergonomic cost turned on, but the symbolic action sequence $s_{1:K}$ is fixed to the choice found with non-ergonomic optimizing.

Our main hypothesis is that by optimizing for the dREBA surrogate, the choice of actions and motions will lead the human model to postures that are more ergonomic in terms of REBA score. Therefore, we use the REBA score of the model posture as a measure to validate this effect. Another hypothesis is that optimizing the path along the full sequences of actions will lead to more ergonomic postures compared to stepwise optimization. The objective of the re-optimized condition is then to serve as another baseline that results from methods that have a separation between the high-level task planner and the low-level motion planning. By improving over this baseline we show that it is worth to pay the cost of using the more computationally expensive simultaneous optimization of task and motion planning that is proposed in this work. For example, in this first experiment $s_{1:K}$ refers to the categorical choice of table, which may be chosen sub-optimally with the non-ergonomic optimizer.

In the *non-ergonomic* condition, the planned trajectory of the agent to grasp the screwdriver might not be natural and lead to non-ergonomic postures like an over bending of the spine. Clearly, this will lead to non-favorable REBA scores. In the *re-optimized* condition, re-optimizing such a path with ergonomic costs, but fixing $s_{1:K}$, allows us an easier comparison with the full *ergonomic* condition, which highlights the effects of the choice of the symbolic sequence $s_{1:K}$.

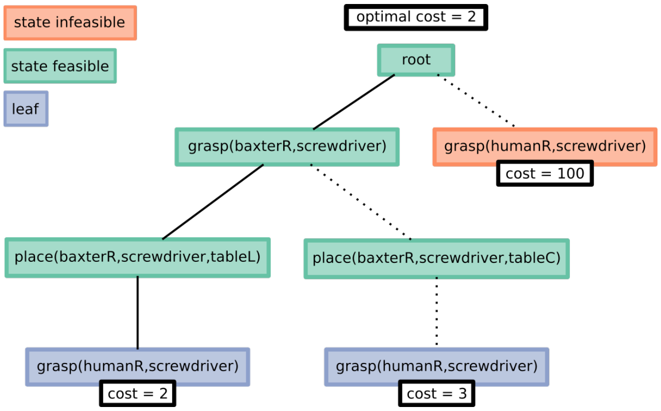

state infeasible
state feasible
leaf
optimal cost = 2
root
grasp(baxterR,screwdriver)
grasp(humanR,screwdriver)
cost = 100
place(baxterR,screwdriver,tableL)
place(baxterR,screwdriver,tableC)
grasp(humanR,screwdriver)
cost = 2
grasp(humanR,screwdriver)
cost = 3

Fig. 4. Example of a tree search made by the MBTS solver to find the optimal sequence of actions under the given constraints. The path in plain lines corresponds to the optimal sequence. A state is declared as infeasible a constraint is broken. In the example case given, the screwdriver is too far away from the human agent which renders it impossible to grasp. The written cost are given for illustrative purposes and do not correspond to real values returned by the solver.

We fix a random height for the three tables and test the three conditions keeping the same table height. The process is repeated 100 times and for each run we collect the table chosen for placing the screwdriver and the human posture at grasping time. We also force the human to be right handed to remove the effects of changing hands to reduce the costs. The initial body posture is set to be at rest according to the REBA assessment (posture of minimal REBA score).

### B. Results

As we observe in the results of Fig. 5 and 7, re-optimizing the path for ergonomics leads, as expected, to a smaller posture score. Still, the best posture scores are obtained in the *ergonomic* condition. Statistical differences are noted directly on the figure and verified with Anova statistical test.

TABLE II
TABLE CHOICE RATIO (%)

| | *non-ergonomic* | *ergonomic* |
|---|---|---|
| left | 89 | 26 |
| center | 7 | 40 |
| right | 4 | 34 |

In terms of table heights, the average height of the chosen table in the *non-ergonomic* condition is $1.08 \pm 0.22$. It is slightly below the average height of the three tables $(1.1 \pm 0.3m)$. The average height of the chosen table in the *ergonomic* condition is $0.95 \pm 0.15m$.

By looking at the optimal choices, summarized in table II, we observe that in the *ergonomic* condition, the most chosen table is the centered one. This makes sense from an ergonomic point of view as deviating the arm on the side is more costly. Nevertheless, this choice also depends on

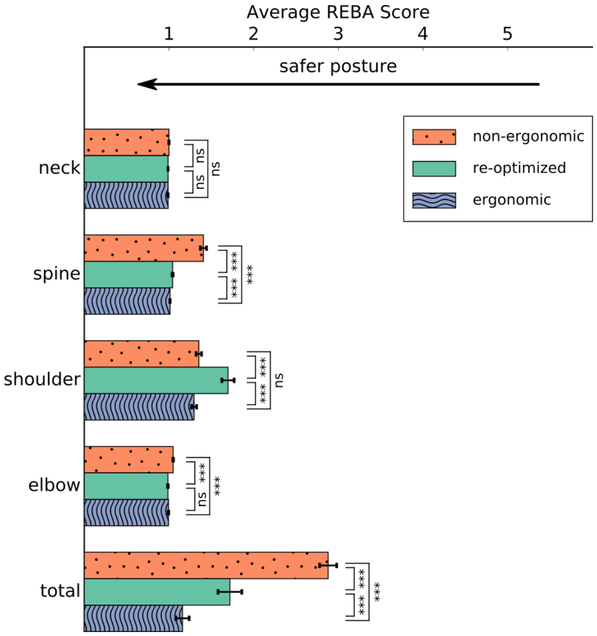


Fig. 5. Average REBA score and standard error of the mean for the three conditions. The lower the score, the safer is the posture. Significance are verified using Anova statistical test and noted according to the standard defined by the APA(American Psychological Association).

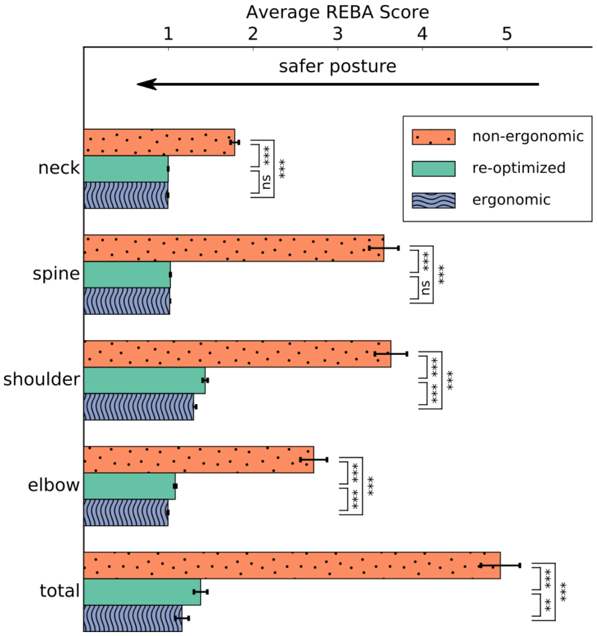


Fig. 6. As Fig. 5, but with the initial body pose randomized.

the table height as suggested by the number of times the left of right table were chosen. On the other side, in the *non-ergonomic* condition, the most chosen table is the one located on the agent's left side. This choice seems to be almost independent from the table height.

This effect arises from the base cost function of the LGP formalism which minimize by default the sum-of-square accelerations of the joints for both the robot and the human. As the time to perform an action is fixed, this leads to select the shortest trajectory for both agents. By extension, placing the screwdriver on the left table often corresponds to the shortest path.

Minimizing the joints acceleration also means that the initial human posture must impact the optimization. To study this effect, we have performed a new experiment, keeping the same experimental setup and reusing the previously generated table heights. The only difference comes from the initial human body posture which is set randomly.

As we observe in Fig. 6 this strongly impacts the posture score of the *non-ergonomic* condition. As the human body possesses a large number of degrees of freedom, some joints are not necessary to move to fulfill the task. Therefore the shortest path is to keep them at their initial value. The left image of Fig. 8 shows an example of a grasping posture impacted by the initial body configuration. In the *ergonomic* and *re-optimized* conditions however, the first human motion is to move back to an ergonomic posture as illustrated in the right image of Fig. 8 and especially in the cost profile of Fig. 7. This action limits the impact of the initial body configuration. As table III suggests, choices of tables is not impacted in the *ergonomic* condition. For the *non-ergonomic* condition, the impact is significant. As the random posture is often bended, such as the one presented in the left image of Fig. 6, the shortest path might be a table of lower height. This point is confirmed by the average height of the chosen tables, $1.02 \pm 0.2m$ which is $6cm$ less than in the normal configuration.

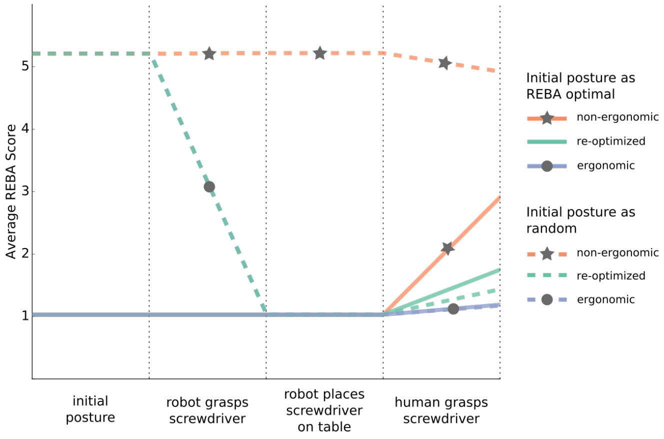


Fig. 7. Evolution of the averaged REBA score during all the phases of the interaction. Plain lines) Initial posture optimal in terms of REBA score. The human starts moving only to grasp the screwdriver which explain the flat score until the last phase. Dashed lines) Initial posutre chosen as random. In the *ergonomic* and *re-optimized* the human moves after the initial phase to get back to an ergonomic posture.

By choosing tables of lower heights, the posture score of the *non-ergonomic* condition slightly gets better. Still, the best REBA scores are obtained in the *ergonomic* condition.

### C. Discussion

From the results of this simulated experiment we conclude that optimizing the sequence of actions for ergonomics leads indeed to safer body posture than only optimizing actions

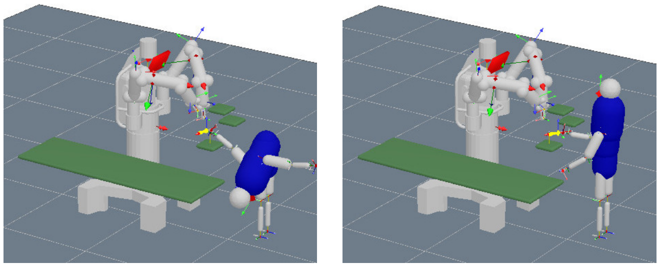


Fig. 8. Left) An example of a grasping posture impacted by the initial body posture in the *non-ergonomic* condition. Without posture correction the shortest path leads to this wrong final posture. Right) The same grasping but in the *re-optimized* condition which counteracts the effects of the initial body posture.

TABLE III

TABLE CHOICE RATIO IN THE RANDOM BODY CONFIGURATION (%)

| | *non-ergonomic* | *ergonomic* |
|---|---|---|
| left | 66 | 26 |
| center | 23 | 40 |
| right | 11 | 34 |

independently. We can also conclude that optimizing the task allocation and motion planning for dREBA induces lower REBA score as expected.

Another interesting result is that considering an ergonomic cost for the body posture allows to remove the effects of the initial body configuration chosen prior to the optimization.

In next section, we introduce a more complex scenario with a longer sequence of actions. As a proof of concept we also implement the generated path and actions on a Baxter humanoid robot.

## IV. Toolbox Assembly Experiment

The second experiment we consider is the assembly of a toolbox as illustrated in Fig. 9. We limit the assembly to screwing the handle (`/toolbox/handle`) and one side of the toolbox (`/toolbox/side_left`) together. We extend the set of possible actions defined in Section II-A with the following decisions:

- `handover(t,e1,o,e2)`
  - *description*: : at time `t`, the end-effector `e1` (robot) hands over object `o` to the end-effector `e2` (agent)
  - *precondition*: `grasped(e1,o) free(t,e2)`
  - *effect*: `¬grasped(e1,o) free(e1) grasped(e2,o) ¬free(e2)`
- `hold(t,e,o)`
  - *description*: : at time `t`, the end-effector `e` (robot or agent) holds object `o`
  - *precondition*: `on_table(o) free(e)`
  - *effect*: `held(o) ¬free(e)`
- `screw(t,e,o1,o2)`
  - *description*: : at time `t`, the end-effector `e` (agent) screws objects `o1` and `o2` together
  - *precondition*: `on_table(o1) on_table(o2) free(e) held(o1)`
  - *effect*: `screwed(o1,o2) ¬held(o1)`

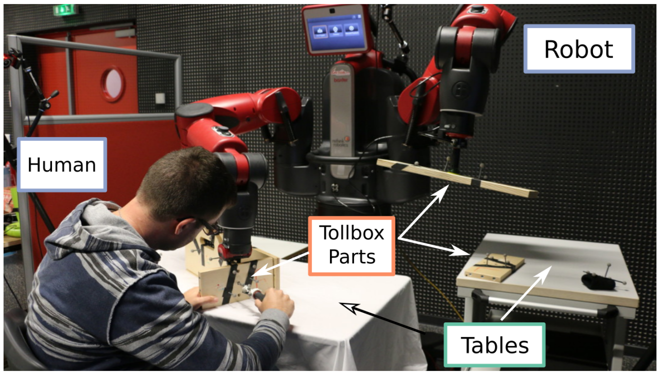


Fig. 9. Picture of the experimental setup of the toolbox assembly annotated using the same convention as Fig. 3.

Initially the `place` action can be accomplished by both the robot and the agent. However, in our setup the robot is equipped with a vacuum gripper on its left end-effector. This type of gripper is convenient to grasp the toolbox parts but cannot be used to precisely place them vertically on the table. Therefore, we limit the `place` decision to only the agent.

To each of those logic decisions we associate a geometric equivalent that defines constraints between objects as kinematic switches. We set the desired goal to `screwed(/toolbox/handle, /toolbox/side_left)` and perform the path optimization. The following is an example of a decision sequence found by the MBTS solver that leads toward the set goal.

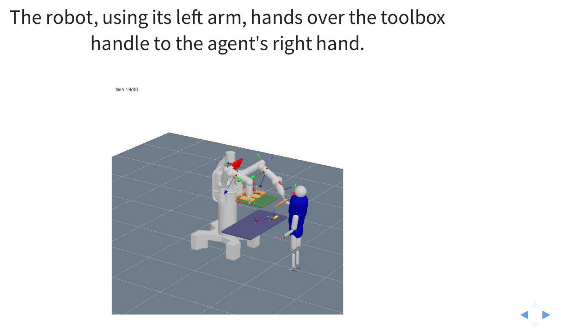
```
grasp(1,baxterL,toolbox/handle)
handover(2,baxterL,toolbox/handle,handR)
grasp(3,baxterL,toolbox/side_left)
place(4,handR,toolbox/handle,tableC)
handover(5,baxterL,toolbox/side_left,handR)
place(6,handR,toolbox/side_left,tableC)
hold(7,baxterR,toolbox/handle)
screw(8,handR,toolbox/side_left,toolbox/handle)
```

The solver also returns a path for both the robot and the agent in terms of joint trajectories and a simulated view of the scenario.

### A. Sequence visualization

In human-robot interaction, communication between the robot and the agent is crucial. For the setup we have considered, the communication is limited to inform the agent about the steps to perform the whole task. If one step is awaiting for an agent's action, it should also detail how this action is performed.

To visualize the planned sequence of actions we generate a webpage with a human readable description of each actions and a simulated view of the last frame of the action path as illustrated in Fig. 10. This webpage is based on *reveal.js* presentation framework [19]. By clicking on the image, the agent launches a video of the full action path. The webpage is accessible offline for the agent to navigate through the sequence of actions or can be seen online to display the current action. We use the *reveal.js* api to automatically



The robot, using its left arm, hands over the toolbox handle to the agent's right hand.

time 19/80

Fig. 10. A representation of an action generated by the solver and displayed on a webpage to simplify the visualization. The agent can click on the image to start a video of the action. Arrows on the right corner are also clickable to navigate between the previous and the next actions.

switch to the current action when the previous action is finished. With this web based approach, we can display this interface on mobile devices (tablets or smartphones) or computer screens.

### B. *Real robot application*

After planning the sequence of actions for the toolbox assembly we convert it into actions to be performed on our Baxter torso humanoid robot. The system also uses a set of predicates to recognize when an action is finished prior to start the next one. Predicates and actions were reused from previous work [20].

At the moment, the system supposes that the agent is do not have the capacity to move freely in the environement, i.e. he or she cannot walk during the execution to avoid re-planning. As in the simulation experiment, the solver planify the motion of both the robot and human agents. However we cannot predict exactly how the real human coworker will act. Therefore, some actions like grasping or handing over toolbox parts are converted directly from the geometric part of the simulation, i.e trajectory of the robot arm to hand over an object are calculated by the solver. For actions following an agent intervention, such as holding a part placed on the table by the agent, we cannot rely on this approach. As the agent might decide to put the object on the table at a different location than the one planned, the trajectory to hold the object might differ. Therefore we track the objects parts using Optitrack motion capture system and generate a new trajectory of the robot arm to the current location of the object. With this approach we do not need to re-plan the full sequence of actions every time the geometry of the scene slightly varies. This greedy approach might, however, be sub-optimal in some cases. Indeed variation of the geometry could lead to changes in the sequence of actions when a full re-planning is performed.

As in the simulation experiment, we calculate the plan in the *ergonomic* and *non-ergonomic* conditions. In this case, we do not consider the *re-optimized* condition as it produces the same sequence of actions as the *non-ergonomic* one. Between the two tested conditions, not only the trajectory of the handing over is affected but also the planned sequence of actions. Changes appear on the first four actions. In the *non-ergonomic* condition the agent is asked to place the handle on the table only after the robot has grasped the next part:

```
grasp(1,baxterL,toolbox/handle)
handover(2,baxterL,toolbox/handle,handR)
grasp(3,baxterL,toolbox/side_left)
place(4,handR,toolbox/handle,tableC)
```

This creates an uncomfortable situation where the agent holds the handle and is unnecessary waiting for his or her next action. Most likely he or she will place it on the table without being told to do so which might create some confusion. In the *ergonomic* condition the sequence is smoother as a hand over is immediately followed by a place action:

```
grasp(1,baxterL,toolbox/handle)
handover(2,baxterL,toolbox/handle,handR)
place(3,handR,toolbox/handle,tableC)
grasp(4,baxterL,toolbox/side_left)
```

This change of the sequence of action is due to the weight of the object being comprised in the calculation of the cost function (1). When holding an object, the cost is slightly higher. Therefore, as the solver minimizes costs over time, placing the object earlier on the table leads to a smaller cost. The rest of the sequence of actions is similar in both conditions. A video of the experiment is available on our Vimeo channel. [1]

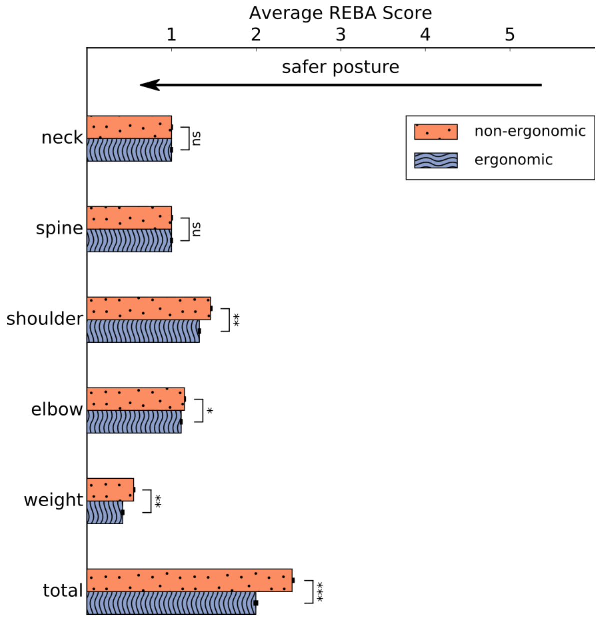


Fig. 11. Average REBA score and standard error of the mean for the two conditions recorded from subjects of the user study.

### C. *User study*

We apply the calculated plans in a user study with 10 participants. Each subject has performed the assembly in the two conditions and we have recorded his or her posture as

[1] https://vimeo.com/232348427

in [8]. Postures score are then calculated and averaged over all subjects and over all the timesteps of the interaction.

In this user study we make the hypothesis that the optimal sequences of actions planned by our system will lead the user toward postures that present a lower REBA score.

Results are presented in Fig. 11 and show that the *ergonomic* condition induces indeed a lower REBA score on average over all the participants. This reduced score is partially explained by the lower *weight* score as objects were held by the subjects for a shorter amount of time. Nevertheless, the impact on the *shoulder* and *elbow* is also significant which indicates that the objects were handed over at a better position.

### D. Discussion

The results of this experiment confirm the two points made with the simulation results. First, optimizing for dREBA correctly leads to lower REBA score, even in a real robot application. Second, in order to generate ergonomic human-robot interaction, considering single actions optimized with ergonomic measures, e.g. ergonomically planning a hand over, is not sufficient. Reasoning in term of sequence of actions leads to more ergonomic situations and improves the overall human comfort.

## V. Conclusion

In this work we have presented a new approach to simultaneously plan task and motion in a collaborative human-robot task while incorporating ergonomic considerations. From the ergonomic field, we took the REBA assessment technique [17] as an index to evaluate the risk induced by the posture of the human in a given situation. Using this index as a cost function we optimize the plan task and the robot motion for maximum ergonomics on the human side.

To accomplish this we had to introduce a differentiable surrogate of the REBA measure (dREBA) suitable to be used in a planning system. Although not strictly equivalent to the original measure, this approximation is an enabler to optimize for the REBA score.

We discuss simulated results, a real human-robot collaborative task and an user study. Results showed that optimizing for the introduced dREBA leads indeed to better human postures according to the original REBA score. They also show that a simultaneous optimization of motion and task is needed to fully minimize ergonomic cost. With this work we aimed to contribute towards safer human-robot collaboration in terms of physical ergonomics.

## References

[1] M. C. Gombolay, R. A. Gutierrez, S. G. Clarke, G. F. Sturla, and J. A. Shah, "Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams," *Autonomous Robots*, vol. 39, no. 3, pp. 293–312, 2015.

[2] O. Ogorodnikova, "Human weaknesses and strengths in collaboration with robots," *Periodica Polytechnica. Engineering. Mechanical Engineering*, vol. 52, no. 1, p. 25, 2008.

[3] H. Bley, G. Reinhart, G. Seliger, M. Bernardi, and T. Korne, "Appropriate human involvement in assembly and disassembly," *CIRP Annals-Manufacturing Technology*, vol. 53, no. 2, pp. 487–509, 2004.

[4] J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals-Manufacturing Technology*, vol. 58, no. 2, pp. 628–646, 2009.

[5] L. Punnett and D. Wegman, "Work-related musculoskeletal disorders: The epidemiologic evidence and the debate," *Journal of Electromyography and Kinesiology*, vol. 14, no. 1, pp. 13–23, 2004.

[6] G. Reinhart, R. Spillner, and Y. Shen, "Approaches of applying human-robot-interaction-technologies to assist workers with musculoskeletal disorders in production," *Intelligent Robotics and Applications*, pp. 74–84, 2012.

[7] E. Takala *et al.*, "Systematic evaluation of observational methods assessing biomechanical exposures at work," *Scandinavian Journal of Work, Environment and Health*, vol. 36, no. 1, pp. 3–24, 2010.

[8] B. Busch, G. J. Maeda, Y. Mollard, M. Demangeat, and M. Lopes, "Postural optimization for an ergonomic human-robot interaction," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'17)*, 2017.

[9] M. Toussaint and M. Lopes, "Multi-bound tree search for logic-geometric programming in cooperative manipulation domains," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'16)*, 2016.

[10] P. Tsarouchi, S. Makris, and G. Chryssolouris, "Human–robot interaction review and challenges on task planning and programming," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 8, pp. 916–931, 2016.

[11] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, "Planning human-aware motions using a sampling-based costmap planner," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5012–5017.

[12] E. A. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1045–1057, 2012.

[13] A. M. Bestick, S. A. Burden, G. Willits, N. Naikal, S. S. Sastry, and R. Bajcsy, "Personalized kinematics for human-robot collaborative manipulation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'15)*. IEEE, 2015, pp. 1037–1044.

[14] E. Plaku and G. D. Hager, "Sampling-based motion and symbolic action planning with geometric and differential constraints," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5002–5008.

[15] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 639–646.

[16] T. Lozano-Pérez and L. P. Kaelbling, "A constraint-based method for solving sequential manipulation planning problems," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3684–3691.

[17] S. Hignett and L. McAtamney, "Rapid entire body assessment (reba)," *Applied Ergonomics*, vol. 31, no. 2, pp. 201–205, 2000.

[18] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *(IJCAI 2015)*, 2015.

[19] "Reveal.js the html presentation framework," http://lab.hakim.se/reveal-js/#/, accessed: 2017-08-28.

[20] M. Toussaint, T. Munzer, Y. Mollard, L. Y. Wu, N. A. Vien, and M. Lopes, "Relational activity processes for modeling concurrent cooperation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'16)*, 2016.