On-line feasible wrench polytope evaluation based on human musculoskeletal models: an iterative convex hull method

Antun Skuric, Vincent Padois, Nasser Rezzoug, David Daney

On-line feasible wrench polytope evaluation based on human musculoskeletal models: an iterative convex hull method

Antun Skuric[1], Vincent Padois[1], Nasser Rezzoug[1,2], David Daney[1]

Abstract—**Many recent human-robot collaboration strategies, such as *Assist-As-Needed* (AAN), are promoting human-centered robot control, where the robot continuously adapts its assistance level based on the real-time need of its human counterpart. One of the fundamental assumptions of these approaches is the ability to measure or estimate the physical capacity of humans in real-time.**

In this work, we propose an algorithm for the feasibility set analysis of a generic class of linear algebra problems. This novel iterative convex-hull method is applied to the determination of the feasible cartesian wrench polytope associated to a musculoskeletal model of the human upper limb. The method is capable of running in real-time and allows the user to define the desired estimation accuracy. The algorithm performance analysis shows that the execution time has near-linear relationship to the considered number of muscles, as opposed to the exponential relationship of the conventional methods. Finally, real-time robot control application of the algorithm is demonstrated in a *Collaborative carrying* experiment, where a human operator and a *Franka Emika Panda* robot jointly carry a 7kg object. The robot is controlled in accordance to the AAN paradigm maintaining the load carried by the human operator at 30% of its carrying capacity.

I. INTRODUCTION

The number of collaborative robots, robots designed to closely coexist with humans, is increasing exponentially in recent years [1], paving the way for more comprehensive and human-centered solutions for human robot collaboration. However, due to the instantaneous nature of the physical interaction, it becomes increasingly challenging to design and tune control laws for the collaborative systems *a priori*. Accounting, in advance, for the high degree of variability induced by humans, usually results in very conservative safety limits and reduces the overall collaboration efficiency considerably. Therefore, new on-line capable and accurate metrics are needed to improve performance of the collaborative systems while ensuring safety for all humans involved. *Assist-As-Needed* (AAN)[2] is one recent example of such paradigm, where a robot continuously adapts the assistance level taking in consideration the capacity of the human operator in real-time.

Several capacity metrics have been developed over the years, first by the robotics community[3][4], and later adapted to the human motion analysis[5][6][7]. The most well known representatives being wrench, twist and acceleration capacity. While twist and acceleration capacity represent the level of dexterity in different directions, wrench capacity represents
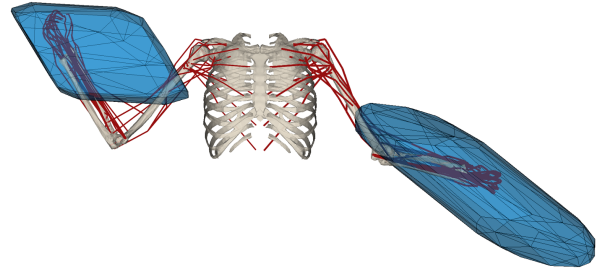
[1] Auctus, Inria Bordeaux, 33405 Talence, France *email: first-name.lastname@inria.fr*

[2] University of Toulon, Toulon, France *email:rezzoug@univ-tln.fr*

Fig. 1. Cartesian force polytope of a musculoskeletal model of both human upper limbs [8] with 7Dof and 50 muscles each, visualised in *pyomeca bioviz* [9]. The polytopes are scaled with a ratio 1m : 1000N

the human ability to apply and resist external wrenches in arbitrary directions, being very important both safety-wise and performance-wise.

The human body capacities are governed by many different factors [10]. Many of them are physiological and biomechanical factors, such as muscle strength, bone length and level of fatigue. The others are purely kinematic such as posture, joint velocities and accelerations. Among the most complete human body modeling approaches are the musculoskeletal models, describing the relationship between the activation of human muscles and the body dynamics. Over the years, many, more or less complete, models have been developed and experimentally validated [11][12][13], as well as different open-source software platforms created for their analysis [14][9].

The exact wrench, twist and acceleration capacity of the musculoskeletal model can be represented in a form of convex polytopes. However due to the relatively high number of muscles usually considered, the exact evaluation of these polytopes is a time consuming process. Therefore, in order to reduce the complexity of their calculation, apart from using less complex models, various approximate techniques have been proposed, using machine learning [15], ellipsoids [16] and approximate polytope estimation techniques [17][18]. However, all of these approaches reduce the precision and confidence in the calculated human capacity.

In this paper, a new algorithm for polytope evaluation is proposed. It allows for the efficient evaluation of the feasible wrench polytope of musculoskeletal human models, given a user specified accuracy. The efficiency is characterized by the low and near-linear complexity with respect to the number of muscles and accuracy. This allows for an online update of the polytope at a rate of 100 ms (10 Hz) for 32 muscles and an accuracy of 5 N (200 ms / 5 Hz for 50 muscles). Beyond these application specific characteristics, the proposed algorithm is generic to a class of implicitly defined linear algebra feasibility

problems

$$Ax = By, \quad y \in \left[\underline{y}, \overline{y}\right] \tag{1}$$

where

$$x \in \mathbb{R}^m, \ y \in \mathbb{R}^d, \ A \in \mathbb{R}^{n \times m}, \ B \in \mathbb{R}^{n \times d}, \ d \geq n \geq m \tag{2}$$

In robotics, this class of problems can be found in many different domains such as when calculating feasible twist capacity of object grasping [19], feasible twist capacity of collaborating robots forming a closed chain [20], or feasible wrench capacity of multilink cable driven parallel robots [21][22].

Section II recapitulates the mathematical formulation of the feasible wrench polytope for musculoskeletal models, followed by the description of the proposed algorithm in Section III. In Section IV the performance characteristics of the algorithm are analysed and compared with state of the art methods. Finally, a demonstrative experiment is shown where the proposed algorithm is used for real-time collaborative robot control.

II. FEASIBLE WRENCH POLYTOPES FORMULATION

The dynamical equation of a general musculoskeletal system expressed in joint space can be formulated as

$$\underbrace{M(q)\ddot{q} + C(q,\dot{q})\dot{q}}_{\tau_d} + \tau_g(q) = \tau - J^T(q)f \tag{3}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ are the joint level generalised coordinates, velocities and accelerations. M is the inertia matrix, C is the Coriolis-centrifugal matrix, τ_g is the joint torque produced by gravity, τ is the joint torque vector generated by the muscle-tendon units, J is the end effector Jacobian matrix and $f \in \mathbb{R}^m$ is an m-dimensional external Cartesian wrench vector. τ_d represents the joint torques induced by the body motion, kineto-static conditions $\dot{q} = \ddot{q} = 0$ yield $\tau_d = 0$.

$$\tau - \tau_d(q, \dot{q}, \ddot{q}) - \tau_g(q) = J^T(q)f \tag{4}$$

From equation (4), it can be seen that the gravity and body motion directly reduce the capacity to resist and apply external forces.

A. Muscle model

The joint torque τ is generated by muscle-tendon units. One of the most well known muscle models was introduced by A. Hill [23] and later refined by F. Zajac [24], where the muscle is approximated by a tensile force composed of an active and a passive component

$$F_i = f_i^A(l_i, v_i)F_{iso,i}\alpha_i + \underbrace{f_i^P(l_i)F_{iso,i}}_{F_{p,i}}, \quad \alpha_i \in [0, 1] \tag{5}$$

f_i^A and f_i^P are active and passive scaling functions depending on the muscle length l_i and contraction/extension velocity v_i. $F_{iso,i}$ is the maximal isometric force the muscle can generate and α_i is the muscle activation level. For a set of d muscles, with specified muscle lengths l and contraction velocities v the achievable set of muscle tensile forces $F \in \mathbb{R}^d$ has a range

$$F \in [F_p, \ F_m] \tag{6}$$

where F_p is the vector of passive forces obtained for $\alpha = 0$ and F_m is the maximal muscle force vector obtained for $\alpha = 1$.

$$\begin{aligned} F_{p,i} &= f_i^P(l_i)F_{iso,i} \\ F_{m,i} &= \left(f_i^A(l_i, v_i) + f_i^P(l_i)\right)F_{iso,i} \end{aligned} \tag{7}$$

The joint torque τ, generated by the muscle tensile force F, can then be calculated using the Moment arm matrix [25]

$$\tau = -L^T(q)F \tag{8}$$

where $L(q)^T$ is the transpose of the moment arm matrix, which corresponds to the muscle length Jacobian relating the space of joint and muscle length velocities

$$\dot{l} = L(q)\dot{q}, \quad L_{ij} = \frac{\partial l_i}{\partial q_j} \tag{9}$$

Finally, the negative sign in equation (8) makes the force applied in the length shortening direction of the muscle positive.

Combining equations (4) and (8), one obtains the relationship relating the d muscle tensile forces F and the m dimensional Cartesian wrench f

$$J^T(q)f = -L^T(q)F - \tau_d(q, \dot{q}, \ddot{q}) - \tau_g(q) \tag{10}$$

B. Residual muscle forces

For any given joint configuration q, velocity $\dot{q}$ and acceleration $\ddot{q}$, the dynamics τ_d and the gravity τ_g torques are constants and can be considered as a bias. Their equivalent bias muscle force F_b can be expressed as

$$-L^T(q)F_b = \tau_d(q, \dot{q}, \ddot{q}) + \tau_g(q) \tag{11}$$

This force F_b can be assessed by finding the minimal muscle activation α that generates the desired torque vector [26]. This can be resolved using a quadratic problem formulation where the muscle activation vector norm $||\alpha||^2$ is minimised, with equation (11) as equality constraint

$$\begin{aligned} \min_{F_b} \quad & \frac{1}{2}F_b^T P F_b \\ \text{s.t.} \quad & -L^T F_b = \tau_g + \tau_d \\ & F_p \leq F_b \leq F_m \end{aligned} \tag{12}$$

where P is a diagonal matrix with elements on diagonal $p_{ii} = 1/(F_{m,i} - F_{p,i})^2$.

Furthermore, the bias muscle forces F_b can then be used to calculate the range of possible residual muscle forces F' from equation (6)

$$F' = F - F_b, \quad F' \in [0, \ F_m - F_b] \tag{13}$$

Finally, without loss of generality, the relationship of the muscle-tendon tension forces F and the Cartesian wrenches f can be expresses as

$$J^T(q)f = -L^T(q)F, \quad F \in \left[\underline{F}, \ \overline{F}\right] \tag{14}$$

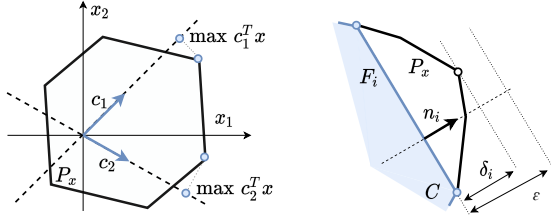where the $\underline{F}$ and $\overline{F}$ are the minimal and maximal muscle-tendon tension forces F.

Fig. 2. Left figure demonstrates the LP based vertex search by choosing two different vectors c. Right figure exemplifies the polytope face condition from equations (24) and (23), where face $\mathcal{F}_i$ of the convex-hull $\mathcal{C}$ is considered to be the face of $\mathcal{P}_x$, because $\delta_i < \varepsilon$.

C. Wrench capacity polytope

A general achievable set of muscle-tendon tensile forces $F \in [\underline{F}, \overline{F}]$ forms a d-dimensional hyper-rectangle with side lengths equal to the ranges of each of the d muscle forces. Using the moment arm matrix $L(q)$ and equation (8) this hyper-rectangle can be projected into the n-dimensional space of joint torques τ, forming the convex polytope of achievable joint torques $\mathcal{P}_\tau$

$$\mathcal{P}_\tau = \left\{ \tau \in \mathbb{R}^n \mid \tau = -L^T(q)F, \; F \in [\underline{F}, \overline{F}] \right\} \quad (15)$$

Furthermore, the dual relationship of the m-dimensional Cartesian wrenches f and the generalised joint torques τ is given through the Jacobian transpose matrix $J^T(q)f = \tau$ and defines the achievable Cartesian force polytope

$$\mathcal{P}_f = \left\{ f \in \mathbb{R}^m \mid J^T(q)f = \tau, \; \tau \in \mathcal{P}_\tau \right\} \quad (16)$$

Combining the definitions (15) and (16) the implicit definition of $\mathcal{P}_f$ can be defined as

$$\mathcal{P}_f = \left\{ f \in \mathbb{R}^m \mid J^T(q)f = -L^T(q)F, F \in [\underline{F}, \overline{F}] \right\} \quad (17)$$

III. POLYTOPE EVALUATION FORMULATION

Given the generic formulation of problem (1) the set of all feasible x forms a convex polytope

$$\mathcal{P}_x = \left\{ x \in \mathbb{R}^m \mid Ax = By, \quad y \in [\underline{y}, \overline{y}] \right\} \quad (18)$$

The polytope evaluation can either be done by finding the set of all the vertices ($\mathcal{V}$-rep) or by finding its half-space representation ($\mathcal{H}$-rep). Most of the commonly used polytope evaluation algorithms are very efficient for finding the exact solution to very specific problems: [27][28][5] calculate the $\mathcal{V}$-rep of $Ax = b, b \in [b]$ and [29][30][31] calculate the $\mathcal{H}$-rep of $x = Ay, y \in [y]$. However since the number of faces and vertices of the polytopes augments exponentially with the size of the system, when the size of the problem becomes high, searching for the exact solution of the combined sub-problems becomes practically intractable [32].

To overcome this issue, various approximative approaches, such as the *Ray Shooting Method* (RSM) [33] and the *Convex-Hull Method* (CHM) [34], have been developed, reducing the complexity and improving the execution time. Although RSM algorithms are relatively simple to setup and have already been used for evaluating the capacity of musculoskeletal models [17], these algorithms do not provide any bound on their estimation error and highly rely on hand tuned initial parameters.

CHM [34] algorithms on the other hand, have shown to be very efficient while at the same time maintaining the bound of the user defined estimation error. CHM uses *linear-programming* (LP) repetitively to find new vertices and the convex-hull algorithm to group them to faces, augmenting the precision of the approximation successively. Additionally, CHM finds the $\mathcal{V}$-rep and the $\mathcal{H}$-rep of the polytope at the same time. However in its standard formulation, the CHM is not suitable for the family of problems given with equation (1).

Therefore, Section III-A introduces an LP formulation adapting the CHM approach to the implicit problem formulation (1) and in Section III-B, the proposed algorithm overview is given, as well as the pseudo-code and the geometrical representation.

A. Linear programming formulation

From a geometrical point of view, solving a linear programming (LP) problem boils down to finding a vertex of a polytope [35]. However, in order for the polytope $\mathcal{P}_x$ to be suitable for LP optimisation in the m-dimensional output space, its implicit formulation (18) needs to be expressed in explicit form.

In the general case, $n \geq m$ and A is not square. To find an explicit form of equation (1), the pseudo-inverse A^+ can be used as a solution only if By belongs to the *image* of A [36]. Therefore, equation (1) can be transformed to

$$x = A^+ By, \quad By \in \mathcal{I}m(A) \quad (19)$$

Using the *Singular value decomposition* [36] of $A = U\Sigma V^T$, an equality constraint can be devised for By to belong to the *image* of A

$$U_2^T By = 0 \quad (20)$$

where rotation matrix $U \in \mathbb{R}^{n \times n}$ is separated into $U_1 \in \mathbb{R}^{n \times m}$, a projector to the *image* of A, and $U_2 \in \mathbb{R}^{n \times (n-m)}$, a projector to its *null-space*.

Finally, combining equations (20), (19) and (1), one can create a linear program [35] capable of reaching all vertices of the polytope $\mathcal{P}_x$

$$
\begin{aligned}
\max_{y} \quad & c^T A^+ By \\
\text{s.t.} \quad & U_2^T By = 0 \\
& \underline{y} \leq y \leq \overline{y}
\end{aligned}
\quad (21)
$$

by appropriately choosing the projection c, as demonstrated on figure 2.

B. Proposed algorithm overview

Given the polytope definition (1) and (18) and the LP problem defined in (21), the proposed method provides a structured way to choose vectors c to find all the vertices ($\mathcal{V}$-rep) and facets ($\mathcal{H}$-rep) of the polytope $\mathcal{P}_x$. In order to do so, the method leverages the iterative convex hull $\mathcal{C}$ calculation, where each newly found vertex extends $\mathcal{C}$. The normal vectors of the faces of the newly obtained $\mathcal{C}$ are used to decide for the new vectors c to use in the LP (21). This iterative process is performed until some specified level of accuracy is reached.

More precisely, the first step of the algorithm constructs an initial set of vertices x_v and an initial convex hull. In a m-dimensional space, the minimal number of points to create a
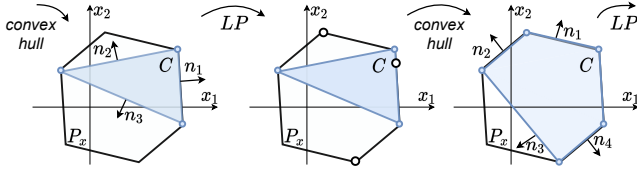
Fig. 3. This figure shows the procedure of successive approximation of the polytope $\mathcal{P}_x$ using the proposed CHM algorithm. The face normal vectors $\boldsymbol{n}_i$ of the convex-hull $\mathcal{C}$ are used with LP (21) to find new vertices which are then used to update the convex-hull $\mathcal{C}$ and furthermore improve the approximation of the polytope.

volume is $m+1$. Since $\boldsymbol{x}$, defined by relation (19), can be expressed as a linear combination of the m base vectors $v_i \in V$ of A,

$$\boldsymbol{x} = a_0 \boldsymbol{v}_0 + \cdots + a_m \boldsymbol{v}_m$$

a way to get the initial set of vertices is to solve LP (21) that minimises ($\boldsymbol{c} = -\boldsymbol{v}_i$) and maximises ($\boldsymbol{c} = \boldsymbol{v}_i$) the projection of the polytope $\mathcal{P}_x$ on the axes of each one of the base vectors $\boldsymbol{v}_i$. Once the initial set of vertices $\boldsymbol{x}_v$ is found, the initial convex hull $\mathcal{C}$ can be calculated. Its centroid position is given by $\boldsymbol{x}_c = \frac{1}{N} \sum \boldsymbol{x}_{v,i}$.

The next step of the algorithm is to iterate over all the faces $\mathcal{F}_i$ of the convex hull $\mathcal{C}$. For each new face $\mathcal{F}_i$, the normal $\boldsymbol{n}_i$ is found and used as a candidate $\boldsymbol{c}_i$ in the direction pointing out of polytope $\mathcal{P}_x$. The normal vector direction is verified by projecting the vector going from the centroid $\boldsymbol{x}_c$ to any vertex $\boldsymbol{x}_j$ of face $\mathcal{F}_i$, onto the normal $\boldsymbol{n}_i$ and verifying if the scalar product is positive or negative.

$$\boldsymbol{c}_i = \begin{cases} \boldsymbol{n}_i, & \text{if } \boldsymbol{n}_i^T(\boldsymbol{x}_j - \boldsymbol{x}_c) \geq 0, \\ -\boldsymbol{n}_i, & \text{otherwise.} \end{cases} \tag{22}$$

Solving equation (21) with $\boldsymbol{c}_i$, the obtained $\boldsymbol{x}_i$ can be either a vertex of the polytope $\mathcal{P}_x$ or be coplanar with face $\mathcal{F}_i$. To determine if $\boldsymbol{x}_i$ is coplanar with the face, a simple check can be devised, which verifies if the orthogonal (normal) distance from $\boldsymbol{x}_i$ to any vertex $\boldsymbol{x}_j$ of face $\mathcal{F}_i$

$$\delta_i = \boldsymbol{n}_i^T(\boldsymbol{x}_j - \boldsymbol{x}_i) \tag{23}$$

is within a certain user defined accuracy ε:

$$\boldsymbol{x}_i = \begin{cases} \text{vertex}, & \text{if } |\delta_i| \geq \varepsilon, \\ \text{on the face}, & \text{otherwise.} \end{cases} \tag{24}$$

Figure 2 provides a graphical interpretation of this check and of the values δ_i and ε.

If $\boldsymbol{x}_i$ belongs to face $\mathcal{F}_i$ of the convex-hull $\mathcal{C}$ then $\mathcal{F}_i$ is considered to be a face of the polytope $\mathcal{P}_x$. In that case the algorithm updates the $\mathcal{H}$-rep of $\mathcal{P}_x$

$$H \leftarrow \begin{bmatrix} H \\ \boldsymbol{n}_i^T \end{bmatrix}, \quad \boldsymbol{d} \leftarrow \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{n}_i^T \boldsymbol{x}_j \end{bmatrix} \tag{25}$$

where $\boldsymbol{n}_i$ is the face normal vector and $\boldsymbol{n}_i^T \boldsymbol{x}_j$ is the orthogonal distance from the origin to the face $\mathcal{F}_i$. On the other hand, if $\boldsymbol{x}_i$ is a vertex of the polytope it is appended to the $\mathcal{V}$-rep list $\boldsymbol{x}_v \leftarrow [\boldsymbol{x}_v, \boldsymbol{x}_i]$.

Once all the faces of the convex hull $\mathcal{C}$ are evaluated, the convex hull is updated using the new vertex list $\boldsymbol{x}_v$. If the vertex list $\boldsymbol{x}_v$ has no new elements or, in other words, if the maximal

distance δ_i for all the faces $\mathcal{F}_i$ is lower than the user defined accuracy ε

$$\max\{|\delta_i|\} \leq \varepsilon \tag{26}$$

the procedure is completed. Otherwise the procedure restarts by iterating over all the new faces $\mathcal{F}_i$ of the updated convex hull $\mathcal{C}$.

A pseudo-code of one example implementation of the proposed algorithm can be found in Algorithm 1. Furthermore, figure 3 visually demonstrates several iterations of the algorithm for the 2-dimensional polytope $\mathcal{P}_x$ example. The code of the proposed algorithm used in this paper is publicly available[1].

Algorithm 1 Proposed CHM algorithm pseudo-code

Require: $A, B, \underline{\boldsymbol{y}}, \overline{\boldsymbol{y}}, \varepsilon$
  $U, \Sigma, V^T \leftarrow svd(A)$
  init $\mathcal{H}$-rep: $H, \boldsymbol{d} \leftarrow [\,]$ and $\mathcal{V}$-rep: $\boldsymbol{x}_v \leftarrow [\,]$
  for all $\boldsymbol{v}_i$ in V **do**
    $\boldsymbol{y}_i \leftarrow$ LP (Eq. 21) with $\boldsymbol{c} = \boldsymbol{v}_i$ and $\boldsymbol{c} = -\boldsymbol{v}_i$
    $\boldsymbol{x}_v \leftarrow [\boldsymbol{x}_v, \ A^+ B \boldsymbol{y}_i]$
  repeat
    calculate the convex hull $\mathcal{C}$ of $\boldsymbol{x}_v$
    calculate the centroid $\boldsymbol{x}_c = \frac{1}{N} \sum_i \boldsymbol{x}_{v,i}$
    for all new face $\mathcal{F}_i$ in $\mathcal{C}$ **do**
      find normal $\boldsymbol{n}_i$ and one vertex $\boldsymbol{x}_j$ of $\mathcal{F}_i$
      $\boldsymbol{y}_i \leftarrow$ LP (Eq. 21) with $\boldsymbol{c} = \boldsymbol{c}_i$ (Eq. 22)
      $\boldsymbol{x}_i = A^+ B \boldsymbol{y}_i$
      calculate δ_i (Eq.23)
      if $|\delta_i| \leq \varepsilon$ **then**
        new face: update $\mathcal{H}$-rep (Eq. 25)
      else
        new vertex: update $\mathcal{V}$-rep $\boldsymbol{x}_v \leftarrow [\boldsymbol{x}_v, \ \boldsymbol{x}_i]$
  until $\max\{|\delta_i|\} \leq \varepsilon$
  return $\boldsymbol{x}_v, H, \boldsymbol{d}$

C. Feasible wrench polytope

The application of the proposed method to compute the feasible Cartesian wrench polytope $\mathcal{P}_f$ is direct. $\boldsymbol{x}$ is chosen as the Cartesian wrench $\boldsymbol{f}$, $\boldsymbol{y}$ with the range $\boldsymbol{y} \in [\underline{\boldsymbol{y}}, \overline{\boldsymbol{y}}]$ is the vector of muscle-tendon tensile forces $\boldsymbol{F} \in [\underline{\boldsymbol{F}}, \overline{\boldsymbol{F}}]$. A is the Jacobian transpose matrix $J^T(\boldsymbol{q})$ and B is the moment arm matrix $-L^T(\boldsymbol{q})$. Finally, the accuracy ε can be interpreted as the maximal underestimation error of the Cartesian wrench $\boldsymbol{f}$ allowable, representing the desired accuracy of the polytope estimation.

IV. EXPERIMENTS AND RESULTS

A. Performance analysis

To evaluate the performance of the proposed CHM algorithm, a comparative experiment is performed. The experiment consists in finding the $\mathcal{V}$-rep of the cartesian force $m = 3$ polytope for a randomised mockup musculoskeletal model with $n = 7$ degrees of freedom and a number of muscles ranging from $d = 20$ to 100.

[1] https://gitlab.inria.fr/auctus-team/people/antunskuric/human_wrench_capacity
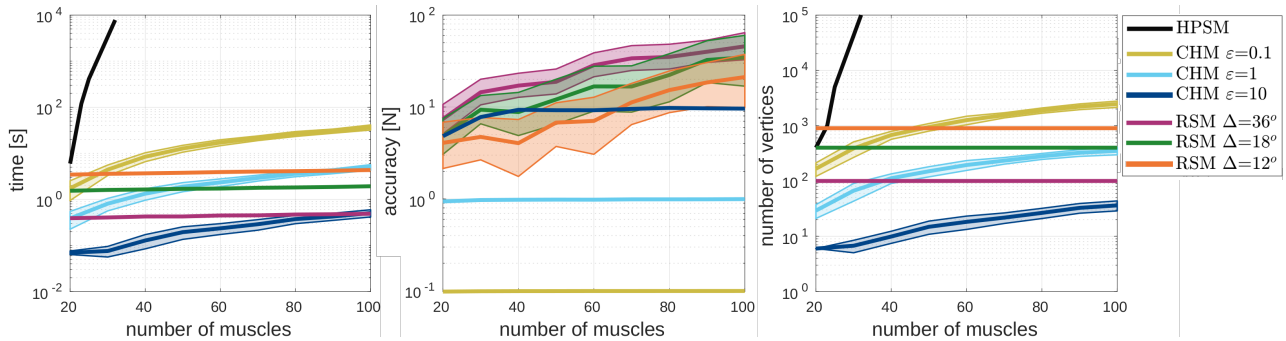
Fig. 4. Figure presenting the performance analysis results in the logarithmic scale for three different algorithms: HPSM, RSM and proposed CHM with respect to different number of muscles. Left figure shows the evolution of the execution time. Middle figure shows the evolution of the underestimation error, the maximal distance in between the polytope $\mathcal{P}_x$ and the acquired polytope, calculated as $\max\{|\delta_i|\}$. Right figure shows the evolution of the number of vertices found. All the plots show the averaged results, with variances, over 100 runs, where each run corresponds to one randomly generated model.

Apart form the proposed method, two additional algorithms are tested. The first algorithm presents an exact approach. It combines the *Hyper plane shifting method* (HPSM) [29], to evaluate the $\mathcal{H}$-rep of the polytope $\mathcal{P}_\tau$, and the *primal-dual* method introduced by Bremer et al. [37], to find the $\mathcal{V}$-rep of $\mathcal{P}_f$. The second method is the RSM algorithm introduced by Carmichael et al. [17] as an example of approximative algorithm used in the context of musculoskeletal models.

The RSM algorithm [17] requires uniform sampling of the *ray directions* in the 3D space. This sampling can be performed based on a two Euler angles parametrization. For this method, three linearly increasing levels of granularity are tested for each Euler angle: $\Delta=36^o$, 18° and 12°; making for $n_r=10, 20$ and 30 *rays* per angle. Overall number of *ray directions* in 3D space ($N_r = n_r^2$) is then 100, 400 and 900. For the proposed CHM algorithm, three exponentially increasing levels of accuracy are tested $\varepsilon=0.1$, 1 and 10 N. All the algorithms are implemented in Matlab and run on a 1.90GHz Intel i7-8650U processor.

Results of the experiments, averaged over 100 algorithm runs, are shown on figure 4. The results confirm that the exact approach using the HPSM method has an execution time exponentially related to the number of muscles. Even for only 30 muscles it already takes more than 4.5 h to calculate, therefore this method is not tested on more than 30 muscles, where it finds over 10^4 vertices.

The RSM algorithm shows nearly constant time of execution for the full range of tested muscles and the constant number of vertices found, which is expected. The results show however, that for low muscle numbers $d<25$, when using a fine granularity $\Delta=12^o$ the RSM algorithm finds more vertices than the exact solution found by the HPSM. Furthermore, the middle plot of figure 4 shows that the RSM estimation error increases considerably with the number of muscles d, followed by a very high variance. These results confirm that, as the polytope shape is not spherical, by uniformly covering the space of ray directions, RSM based algorithms will necessarily estimate certain areas of the polytope better than the others.

The graphs show that the proposed CHM method's execution time depends near-linearly of the number of muscles considered d and the estimation error bound parameter ε. The accuracy graph, shown in the middle plot, shows that the proposed CHM algorithm is capable of limiting the estimation error of the polytope evaluation under desired value ε regardless of

the number of muscles. Furthermore, considering the vertex number found by the algorithms, it can be seen that the number of vertices has a nearly-linear relationship with the number of muscles d and the variable ε.

The demonstrated efficiency of the proposed CHM algorithm opens many doors for its applications in real-time systems, providing the user with an easy to understand trade-off between speed and accuracy.

B. Collaborative object carrying

An example experiment is conducted, designed as a simplified heavy object manipulation task common in the industrial assembly procedures. In the experiment, shown in the figure 5, a human operator and a *Franka Emika Panda* robot jointly carry $m_o=7$ kg object, each one compensating for a part of the total weight $f_h+f_r=m_o g=G_o$. The human operator visually navigates the object through the common workspace, passing through the set of 4 via-points. The experiment is repeated two times, for two different operator positions, equally distant from the via-points, but rotated by 90° in space.

Naive gravity compensation strategy would be the usual choice for these experiments, making the robot carry the full weight of the object and provide the operator with full movement transparency. However, several studies have shown that the human's physical feedback during task execution improves significantly his engagement in the task [38], improves the task execution efficiency and reduces the risk of the human error [39]. Since the human's carrying capacity varies significantly throughout the workspace, allocating any fixed weight to the operator requires *a priori* analysis of the task and determination of the, often conservative, *"worst-case"* weight limits [40][41], often highly specific to the task in question.

As the human operator's carrying capacity can be expressed as the maximal force the operator's arm can generate in the z-axis (vertical) direction, proposed CHM algorithm can be used to calculate it in real-time. Inspired by the *Assist-As-Needed* (AAN)[2] paradigm, a simple adaptive weight distribution strategy is designed, with goal to ensure that the human operator's relative load remains constant with respect to its real-time capacity. The fixed ratio chosen for the experiments is 30% of the human's carrying capacity, ensuring operator's safety and maintaining the constant level of engagement.

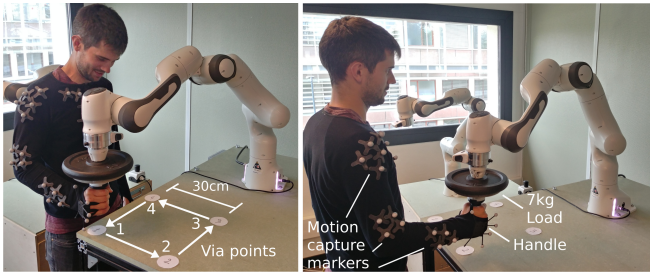$$f_h = 0.3 f_{h,max}, \quad f_r = G_o - f_h$$

Fig. 5. These figures show the experimental setup. The robot end-effector is fixed to the carried object and the human operator is holding the object by the handle. The motion capture system is used to acquire the pose of the human arm. The via-points are visually indicated to the human operator with numbered stickers on the table placed at the corners of a 30 cm square. The images shows the experiment 1 (left) and the experiment 2 (rigth).

The human upper limb configuration is inferred in real-time using an *Optitrack* motion capture system. The musculoskeletal model analysis is developed using the efficient C++ library *pyomeca biorbd* [9]. The musculoskemetal model used in this experiment is the 50 muscles, 7 degrees of freedom MOBL-ARMS model [8] developed by Holzbaur et al. [11]. The human subject (male, 182 cm, 80 kg) and the model belong to the same 50th percentile anthropomorphic group [42] so no model scaling [43] is necessary.

The robot is controlled using the direct force control

$$\boldsymbol{\tau}_r = J_{ee}^T(\boldsymbol{q})f_r + \boldsymbol{\tau}_{g,r}(\boldsymbol{q})$$

where J_{ee} is the end-effector Jacobian, $\boldsymbol{\tau}_{g,r}$ is robot's gravity torque vector and $\boldsymbol{\tau}_r$ are the joint torques applied to the robot. The carrying capacity of the robot is calculated using the real-time algorithm [28], making sure that its capacity is not exceeded as well. All the algorithms are implemented in Python, integrated using the *Robot operating system* (ROS) and run on a computer with 1.90GHz Intel i7-8650U processor.

Figure 6 confirms that the evolution of robot's carrying capacity does not change in both experiments, as the via-points' position remains fixed, whereas human's carrying capacity profile changes significantly. The human's carrying capacity in both experiments peaks around 20 kg demonstrating that the human's force capacity is not negligible with respect to the robot's, however its large variability, from 20 kg to less than 5 kg, emphasizes the importance of measuring it in real-time.

Figure 6 further shows that, by calculating the operator's carrying capacity in real-time, using the same simple control law, the robot is able to adapt its assistance strategy despite the substantial change in the operator's capacity profile. To demonstrate this adaptability even further, in addition to experiments 1 and 2 , a third experiment is included in the video attachment, where the human operator moves freely in space.

V. Prospective for human-robot collaboration

The proposed algorithm presents a new and efficient way of calculating human's wrench capabilities which, as opposed to the known methods (ellipsoids [3][5] or RSM algorithm [17]), can guarantee user defined level of accuracy. It opens many possibilities for wider use of human wrench polytopes in the area of the human-robot collaboration, by both reducing the computation time and enabling the use of more complete human models, better describing human subjects [44].
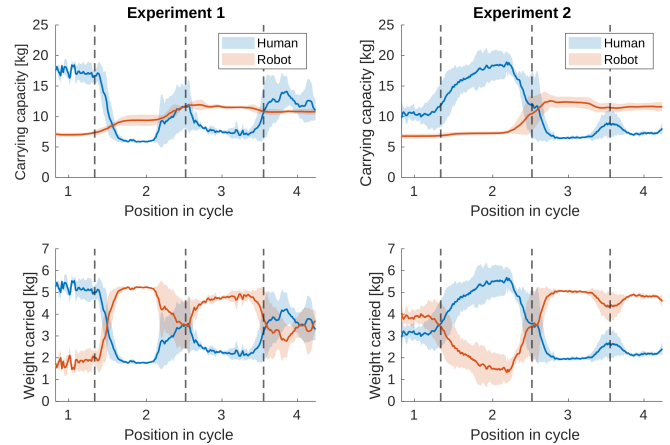

Fig. 6. This figure shows the evolution of the carrying capacity and weight carried by the operator and the robot over the course of the via-point cycle, for two experiments. The plots show mean values and the variances of the curves calculated over 10 successive cycles. The graph regions belonging to the different via-points are separated by vertical lines. All the values are expressed in kilograms for easier readability.

Analysis based on the more complete human models and their pathology could enhance the exoskeleton and prosthesis design, while the efficiency might accelerate analysis of wider anthropomorphic groups of people and improve the design of collaborative workspaces. The algorithm's real-time ability, on the other hand, may enable higher degree of human-centered robot control. The operators' accurate real-time capacity, based on the detailed musculoskeletal models, could be used not only to enforce safety and improve engagement, but to provide the operator-specific assistance profiles. Preventing the development of the work related injuries and potentially even enabling the rehabilitation in the factory context.

VI. Conclusion

This paper introduces a new generic iterative convex hull based algorithm for evaluating the polytope associated to the class of over-determined problems $A\boldsymbol{x} = B\boldsymbol{y}$, where $\boldsymbol{y}$ is a set. The efficiency of the method is demonstrated on the evaluation of the feasible wrench polytope of human musculoskeletal models. In that context, the execution time of the proposed method is shown to be nearly-linear with respect to the number of muscles in the model as well as to the user defined accuracy.

The potential of the method to be used for collaborative robot control is demonstrated in a collaborative carrying experiment, where the human operator and a robot jointly carry a 7kg object. The experiment shows that by combining the real-time human's capacity evaluation with a simple control law, a highly flexible collaboration scenario can be created.

Even though the proposed algorithm itself is independent of any musculoskeletal model, it relies entirely on the model's real-time representation of the operator. Therefore, for practical implementations, the key challenges become: evaluating human posture in the real-time, precise enough and in a minimally intrusive way, as well as performing the individual scaling and calibration of the used musculoskeletal models for a given operator [43].

References

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[2] M. G. Carmichael and D. Liu, "Admittance control scheme for implementing model-based assistance-as-needed on a robot," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 870–873, IEEE, 2013.

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

[4] P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot, "Evaluation of force capabilities for redundant manipulators," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, (Minneapolis, MN, USA), pp. 3520–3525, 1996.

[5] M. Sasaki, T. Iwami, K. Miyawaki, I. Sato, G. Obinata, and A. Dutta, "Vertex search algorithm of convex polyhedron representing upper limb manipulation ability," *Search algorithms and applications*, p. 455, 2011.

[6] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," *Journal of physiology-Paris*, vol. 103, no. 3-5, pp. 211–219, 2009.

[7] N. Rezzoug, V. Hernandez, and P. Gorce, "Upper-limb isometric force feasible set: Evaluation of joint torque-based models," *Biomechanics*, vol. 1, no. 1, pp. 102–117, 2021.

[8] K. R. Saul, X. Hu, C. M. Goehler, M. E. Vidt, M. Daly, A. Velisar, and W. M. Murray, "Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model," *Computer methods in biomechanics and biomedical engineering*, vol. 18, no. 13, pp. 1445–1458, 2015.

[9] B. Michaud and M. Begon, "biorbd: A c++, python and matlab library to analyze and simulate the human body biomechanics," *Journal of Open Source Software*, vol. 6, no. 57, p. 2562, 2021.

[10] NASA, "Man-systems integration standards." Volume 1. 2021, Online: https://msis.jsc.nasa.gov/Volume1.htm.

[11] K. R. Holzbaur, W. M. Murray, and S. L. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," *Annals of biomedical engineering*, vol. 33, no. 6, pp. 829–840, 2005.

[12] W. Wu, P. V. Lee, A. L. Bryant, M. Galea, and D. C. Ackland, "Subject-specific musculoskeletal modeling in the evaluation of shoulder muscle and joint function," *Journal of Biomechanics*, vol. 49, no. 15, pp. 3626–3634, 2016.

[13] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, and S. L. Delp, "Full-body musculoskeletal model for muscle-driven simulation of human gait," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 10, pp. 2068–2079, 2016.

[14] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[15] V. Hernandez, N. Rezzoug, P. Gorce, and G. Venture, "Force feasible set prediction with artificial neural network and musculoskeletal model," *Computer methods in biomechanics and biomedical engineering*, vol. 21, no. 14, pp. 740–749, 2018.

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

[17] M. G. Carmichael and D. Liu, "Towards using musculoskeletal models for intelligent control of physically assistive robots," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 8162–8165, 2011.

[18] S. Chen, J. Yi, and T. Liu, "Strength capacity estimation of human upper limb in human-robot interactions with muscle synergy models," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1051–1056, IEEE, 2018.

[19] D. Prattichizzo and J. C. Trinkle, *Grasping*, pp. 955–988. Cham: Springer International Publishing, 2016.

[20] A. Bicchi and D. Prattichizzo, "Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms," *IEEE transactions on robotics and automation*, vol. 16, no. 4, pp. 336–345, 2000.

[21] D. Lau, J. Eden, Y. Tan, and D. Oetomo, "Caspr: A comprehensive cable-robot analysis and simulation platform for the research of cable-driven parallel robots," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3004–3011, 2016.

[22] X. Sheng, L. Tang, X. Huang, L. Zhu, X. Zhu, and G. Gu, "Operational-space wrench and acceleration capability analysis for multi-link cable-driven robots," *Science China Technological Sciences*, vol. 63, no. 10, pp. 2063–2072, 2020.

[23] A. V. Hill, "The heat of shortening and the dynamic constants of muscle," *Proceedings of the Royal Society of London. Series B-Biological Sciences*, vol. 126, no. 843, pp. 136–195, 1938.

[24] F. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control," *Critical reviews in biomedical engineering*, vol. 17, no. 4, p. 359—411, 1989.

[25] M. G. Pandy, "Moment arm of a muscle force," *Exercise and sport sciences reviews*, vol. 27, no. 1, pp. 79–118, 1999.

[26] F. C. Anderson and M. G. Pandy, "Static and dynamic optimization solutions for gait are practically equivalent," *Journal of Biomechanics*, vol. 34, no. 2, pp. 153–161, 2001.

[27] K. Fukuda and A. Prodon, "Double description method revisited," in *Combinatorics and Computer Science* (M. Deza, R. Euler, and I. Manoussakis, eds.), (Berlin, Heidelberg), pp. 91–111, Springer Berlin Heidelberg, 1996.

[28] A. Skuric, V. Padois, and D. Daney, "On-line force capability evaluation based on efficient polytope vertex search," in *ICRA 2021 - IEEE International Conference on Robotics and Automation*, (Xi'an, China), May 2021.

[29] M. Gouttefarde and S. Krut, "Characterization of parallel manipulator available wrench set facets," in *Advances in Robot Kinematics: Motion in Man and Machine* (J. Lenarcic and M. M. Stanisic, eds.), (Dordrecht), pp. 475–482, Springer Netherlands, 2010.

[30] G. B. Dantzig and B. C. Eaves, "Fourier-motzkin elimination and its dual," *Journal of Combinatorial Theory, Series A*, vol. 14, no. 3, pp. 288–297, 1973.

[31] C. Jones, E. C. Kerrigan, and J. Maciejowski, "Equality set projection: A new algorithm for the projection of polytopes in halfspace representation," tech. rep., Cambridge University Engineering Dept, 2004.

[32] T. Huynh, C. Lassez, and J. Lassez, "Practical issues on the projection of polyhedral sets," *Annals of Mathematics and Artificial Intelligence*, vol. 6, pp. 295–315, 2005.

[33] P. K. Agarwal and J. Matoušek, "Ray shooting and parametric search," *SIAM Journal on Computing*, vol. 22, no. 4, pp. 794–806, 1993.

[34] C. Lassez, "Quantifier elimination for conjunctions of linear constraints via a convex hull algorithm," *Symbolic and Numerical Computation for Artificial Intelligence*, pp. 103–122, 1992.

[35] S. Vajda and S. I. Gass, "Linear programming. methods and applications," *OR*, vol. 15, no. 4, p. 412, 1964.

[36] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on Automatic Control*, vol. 25, pp. 164–176, Apr. 1980.

[37] D. Bremner, K. Fukuda, and A. Marzetta, "Primal—dual methods for vertex and facet enumeration," *Discrete & Computational Geometry*, vol. 20, no. 3, pp. 333–357, 1998.

[38] P. Rani and N. Sarkar, "Operator engagement detection for robot behavior adaptation," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, p. 1, 2007.

[39] E. A. Byrne and R. Parasuraman, "Psychophysiology and adaptive automation," *Biological Psychology*, vol. 42, no. 3, pp. 249–268, 1996. Psychophysiology of Workload.

[40] C. Shoaf, A. Genaidy, W. Karwowski, T. Waters, and D. Christensen, "Comprehensive manual handling limits for lowering, pushing, pulling and carrying activities," *Ergonomics*, vol. 40, no. 11, pp. 1183–1200, 1997.

[41] Health and Safety Commission, *Manual Handling: Manual Handling Operations Regulations 1992, Guidance on Regulation L23*. HSE, 2016.

[42] C. C. Gordon, T. Churchill, C. E. Clauser, B. Bradtmiller, and J. T. McConville, "Anthropometric survey of us army personnel: methods and summary statistics 1988," tech. rep., Anthropology Research Project Inc Yellow Springs OH, 1989.

[43] T. A. Correa and M. G. Pandy, "A mass–length scaling law for modeling muscle strength in the lower limb," *Journal of Biomechanics*, vol. 44, no. 16, pp. 2782–2789, 2011.

[44] M. H. Sohn, D. M. Smith, and L. H. Ting, "Effects of kinematic complexity and number of muscles on musculoskeletal model robustness to muscle dysfunction," *PLOS ONE*, vol. 14, pp. 1–16, 07 2019.