

# Human-Robot Co-Simulation method for upper limb assistive force calculation using polytopes

Maël Gallois<sup>e\*</sup>, Mégane Millan<sup>b</sup>, Nicolas Vignais<sup>c</sup>,  
Sylvain Guégan<sup>d</sup>, Marie Babel<sup>e</sup>, Justin Carpentier<sup>b</sup>,  
Charles Pontonnier<sup>a</sup>

<sup>a</sup> Univ Rennes, ENS Rennes, Inria, CNRS, IRISA, Rennes, France

<sup>b</sup> Inria - Département d'Informatique de l'École normale supérieure, PSL Research University, Paris, France

<sup>c</sup> M2S Laboratory, Rennes 2 Université, Rennes, France

<sup>d</sup> Univ Rennes, INSA Rennes, LGCGM, Rennes, France

<sup>e</sup> Univ Rennes, INSA Rennes, Inria, CNRS, IRISA, Rennes, France

\*Corresponding author: mael.gallois@inria.fr

Received date: 06/04/2025

Accepted date: 27/06/2025

Publication date: 27/10/2025

**Keywords:** force feasible set, upper-limb exoskeleton, constrained dynamics

© 2025 The Authors

Licence CC-BY 4.0

Published by Société de Biomécanique

## 1. Introduction

Assisting the upper limb constitutes a significant challenge in the rehabilitation and readaptation of individuals with neuromuscular and/or neurodegenerative disorders. To address this issue, robotic devices such as exoskeletons have been designed. However, the control of these devices remains intricate and challenging, particularly in the context of the upper limb.

The objective of this study is to define a method to compute the assistance an exoskeleton should provide to the user, according to its capabilities. With the objective to minimize the user voluntary effort, optimal assisting forces that minimize the torque the user must exert along a movement while maximizing the forces provided at the user interfaces are computed. Polytopes (Skiric 2023) are used to determine feasible sets.

## 2. Methods

A human-robot co-simulation was conducted using the Pinocchio robotic simulation library (Carpentier 2019). The proposed exoskeleton (Gallois 2025) provides assistance for shoulder flexion/extension and abduction/adduction, as well as elbow flexion/extension. Internal/external rotation remains free (Fig. 1.a). The human osteo-articular model used is developed by K. Saul *et al.* (2015). Two rigid contact points between the exoskeleton's orthoses and the human body are considered on the upper arm and on the forearm. The necessary assistance forces will be computed at these points (Fig. 1.b).

Firstly, a geometric closure is computed between the human and the exoskeleton models, determining the exoskeleton's joint position and velocity  $(q_r, \dot{q}_r)$  as a function of the user primes joint position and velocity  $(q_h, \dot{q}_h)$ . As with the real system, degrees of freedom are left free between the exoskeleton and the orthosis to enable the geometric closure (Fig 1.b).

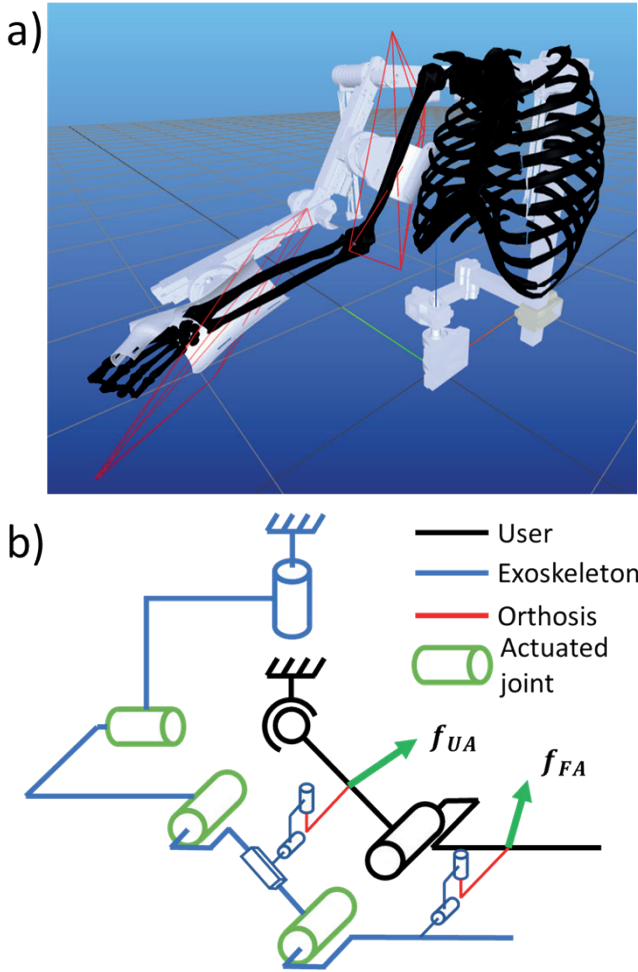
The Python library Pycapacity (Skuric 2023) is used to compute the force-feasible sets of the exoskeleton (Fig. 1.a). These polytopes, denoted  $P_{f_{UA}}$  and  $P_{f_{FA}}$ , are defined at the upper-arm and forearm contact points respectively. Human dynamics can be described by Newton's second law:

$$M_h \ddot{q}_h + C_h \dot{q}_h + \tau_{gh} = \tau_{mus} + J_{h_{UA}} f_{UA} + J_{h_{FA}} f_{FA}$$

where  $M_h$  is the user mass matrix,  $C_h$  is the Coriolis-centrifugal matrix,  $\tau_{gh}$  is the joint gravity torque vector,  $\tau_{mus}$  is the joint muscle torque vector, are respectively the Jacobian matrix and the robot-user interaction force vector at the upper-arm contact point,  $J_{h_{FA}}$  and  $f_{FA}$  at the forearm contact point.

The proposed methodology entails the identification of the  $f_{UA}$  and  $f_{FA}$  actions within the force feasible sets that result in the minimization of

$$(f_{UA}^p, f_{FA}^p) = \min_{f_{UA}^p, f_{FA}^p \in (P_{f_{UA}} \times P_{f_{FA}})} \left( \|\tau_{mus}\|^2 + w(\|f_{UA}\|^2 + \|f_{FA}\|^2) \right),$$



**Figure 1.** (a) Human and robot models for co-simulation. Robot's force polytopes for upper-arm and forearm contacts are represented in red (Mesh python library). (b) Kinematic representation of the geometric linkage between the robot and human models. For visualization purposes, the user's shoulder is simplified as a ball-and-socket joint.

where  $w > 0$  is a scalar factor. It avoids unnecessary effort that would otherwise be compensated between  $f_{UA}$  and  $f_{FA}$ . It must be small enough to avoid interference with the minimization of  $\|\tau_{mus}\|^2$ .

Additionally, we use the minimization function to find the forces  $\{f_{UA}^*, f_{FA}^*\}$  that would allow the exoskeleton to completely take over the motion that is defined as

$$\min_{f_{UA}, f_{FA} \in \mathbb{R}^3} \|\tau_{mus}\|^2 + w(\|f_{UA}\|^2 + \|f_{FA}\|^2)$$

$$\begin{cases} f_{UA}^h = f_{UA}^* - f_{UA}^p \\ f_{FA}^h = f_{FA}^* - f_{FA}^p \end{cases}$$

The differences represent the force that the user must apply at the contact points to perform the movement.

Finally, we performed inverse dynamics to calculate the joint torques of the robot and the human model.

### 3. Results and discussion

In this study, we simulated 3 seconds upper limb realistic movement where the user is moving an object in front of him. The related Saul's coordinates, elevation angle, shoulder rotation, shoulder elevation, and elbow flexion are moving from respectively  $\{115^\circ, 30^\circ, 10^\circ, 35^\circ\}$  to  $\{150^\circ, 70^\circ, 80^\circ, 30^\circ\}$ .

The findings indicate that the exoskeleton's maximum torque capacity is never reached ( $\tau_{mot_{max}} = 20N.m$ ) (Tab. 1).

The user must still exert a portion of the necessary effort. Given that internal-external rotation is not motorized, the user must generate movement around it. However, the required joint torques remain at low levels when compared to the values reported in the literature (Saul 2015).

**Table 1.** Result for a 3 second movement. The names of the human joints refer to the Saul model (2015).

Name	min	max	mean
$\ f_{UA}^p\ $ (N)	14.72	19.32	17.46
$\ f_{FA}^p\ $ (N)	18.28	18.91	18.57
$\ f_{UA}^h\ $ (N)	$\approx 0$	$\approx 0$	$\approx 0$
$\ f_{FA}^h\ $ (N)	0.26	0.62	0.32
Robot shoulder abd. (N.m)	0.18	3.60	2.87
Robot shoulder flex. (N.m)	7.84	13.42	11.61
Robot elbow flex. (N.m)	2.94	5.51	4.05
Human elevation (N.m)	-2.11	-1.71	-1.90
Human shoulder rot. (N.m)	-0.97	-0.76	-0.86
Human shoulder elevation (N.m)	-1.97	1.59	-0.52
Human elbow flex. (N.m)	-0.60	0.26	-0.21

These findings validate the exoskeleton's dynamic capabilities, demonstrating that it can efficiently support upper-arm movements. The next step will consist in using this model as a control model for the exoskeleton. Experimental validation will determine the real forces that can be transmitted to the user. In particular, the chosen contact model (assuming a rigid connection between the exoskeleton and the user) does not allow to anticipate deformation or backlash effects.

#### 4. Conclusion

This co-simulation enables the computation of the required assistance to perform a movement while considering the exoskeleton's capabilities and minimizing human effort. The next step will be to implement it as a robotic assistance.

#### References

- Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamiroux, F., Stasse, O., & Mansard, N. (2019). The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *2019 IEEE/SICE International Symposium on System Integration (SII)* (pp. 614–619) Paris, France. <https://doi.org/10.1109/SII.2019.8700380>
- Gallois, M., Manzano, M., Guégan, S., Vignais, N., Babel, M., & Pontonnier, C. (2025). Effort generation capabilities mapping for personalized robotic assistance of the elbow. In *2025 International Conference On Rehabilitation Robotics (ICORR)* (pp. 1215–1220). Chicago, IL, USA. <https://doi.org/10.1109/ICORR66766.2025.11063068>
- Saul, K. R., Hu, X., Goehler, C. M., Vidt, M. E., Daly, M., Velisar, A., & Murray, W. M. (2015). Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer methods in biomechanics and biomedical engineering*, *18*(13), 1445–1458. <https://doi.org/10.1080/10255842.2014.916698>
- Skuric, A., Padois, V., & Daney, D. (2023). Pycapacity: a real-time task-space capacity calculation package for robotics and biomechanics. *Journal of Open Source Software*, *8*(89), 5670. <https://doi.org/10.21105/joss.05670>
-