Simulating Upper Body Exoskeleton on a Digital Human Model

Alexandre Oliveira Souza[1, 2], Jean Michenaud[1], Raphael Lartot[1], Jordane Grenier[2], François Charpillet[1], Pauline Maurice[1], Serena Ivaldi[1]

***Index Terms*—Dynamic Simulation, Physics-based Simulation, Exoskeleton, QP Control**

I. Introduction

Work-related musculoskeletal disorders (WMSDs) are a huge issue for workers' health and represent an important cost for companies [1]. WMSDs occur when the worker's activity exceeds his physical capabilities. A way to reduce physical strains is to use exoskeletons [2]. In this work, we focus on an active upper-body exoskeleton designed to assist humans in load carrying. For this type of active exoskeleton, the main challenge, once it has been mechanically designed, is to design a good controller. Indeed, it requires many tests with users and a considerable amount of data when learning techniques are involved. In future work, learning will be used to predict human motion intention, in order to assist them as needed. This leads to the necessity of having a simulator, in which data can be acquired and controllers can be tested quickly, while keeping both the user and the hardware safe.

The goal of this work is to animate a Digital Human Model (DHM) equipped with an exoskeleton using Motion Capture (MOCAP) and model the physical interaction between the human and the exoskeleton. This simulator should allow having the exoskeleton interact with the DHM so that control strategies can be tested, as well as allow to acquire virtual sensor data, necessary for the training of Machine Learning (ML) algorithms. As a preliminary milestone, we present a method to have an exoskeleton following the movements of a DHM, as if the digital human was wearing the exoskeleton, using a Quadratic Programming (QP) controller.

II. Methods

MOCAP data are used to animate a non-physical DHM. In the same simulation engine, we simulate a physical exoskeleton following the DHM using a QP controller.

A. Digital Human Model

A DHM is used to replay MOCAP data. The DHM is a numerical rigid-body human model, inspired by humanoid robotics model. Currently, it is a non-physical kinematic puppet, used to replay MOCAP data and to calculate interaction points with the exoskeleton. In this work, we use a database acquired from Xsens, which is a whole body inertial motion capture system. To replay Xsens data in simulation, we use a DHM with 66 DoFs (DHM66). It is a copy of the Xsens avatar, which facilitates the replaying of MOCAP data. The simulation engine used in this work is Robot Dart, a C++11 robot simulator based on the DART physics engine [3], [4].

B. Exoskeleton Description

We use an upper body active exoskeleton. It is composed of: 4 active actuators, located on the shoulders and elbows joints (respectively in red and green on Fig. 1). Multiple passive DoFs from the hips to the elbows, to ensure anthropomorphic kinematics with as few constraints as possible while moving.

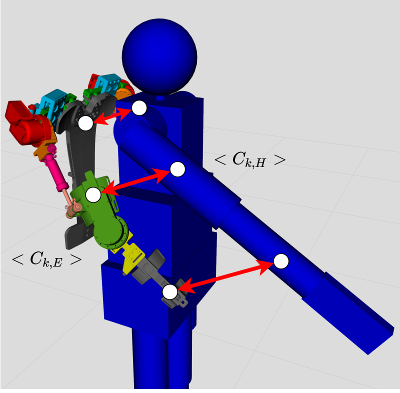

Fig. 1. Contact points between the exoskeleton and the DHM. For each contact point k, a frame is placed on the DHM $< C_{k,H} >$ and another frame on the exoskeleton $< C_{k,E} >$. These frames will serve as "targets" for the controller.

C. An optimization approach

To mimic the transparent behavior of the exoskeleton, i.e., the exoskeleton following the human movement without applying any force, we formulate the exoskeleton control as a motion tracking problem where the objective is to minimize the distance between the contact points on the exoskeleton and the DHM. The problem consists in solving at every time step of the simulation the following QP optimization problem :

$$\underset{\ddot{q},\tau}{\text{argmin}} \sum_k w_k \|\ddot{y}_k - \ddot{y}_k^d\|^2, \tag{1}$$

s.t. dynamics constraints and joints limits

With $\ddot{q}$ the generalized joint angular acceleration, and τ the actuation joint torque. We denote by y_k the current Cartesian positions of frames linked to the parts of the exoskeleton where

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] Safran Electronics & Defense, Valence, France

there should be contact points between the exoskeleton and the DHM (Fig. 1). $\ddot{y}_k$ are their corresponding accelerations. $\ddot{y}_k^d$ is the desired acceleration. The quantity $\|\ddot{y}_k - \ddot{y}_k^d\|^2$ is the acceleration error associated to the Cartesian task k. w_k represents the weight of the task k, i.e. the tasks can be prioritized. $\ddot{y}_k^d$ is calculated using PD control :

$$\ddot{y}^d = \ddot{y}^{goal} + K_d(\dot{y}^{goal} - \dot{y}) + K_p(y^{goal} - y) \qquad (2)$$

With y^{goal} the desired Cartesian position of the task. In this work, we consider $\ddot{y}^{goal} = 0$, $\dot{y}^{goal} = 0$, $K_d = 2\sqrt{K_p}$ and K_p being a tunable parameter.

At each time step, the position, and orientation of the contact points frames of the human (extracted using the MOCAP data applied on the DHM model) are given to the QP controller as an objective, i.e., the position and orientation of the human frame $< C_{k,H} >$ are given as y_k^d and the position and orientation of the corresponding current state of the frame linked to the exoskeleton $< C_{k,E} >$ are given to the controller as y_k, as explained on Figure 2. The torque calculated from the resolution of the problem allows controlling the exoskeleton in the physics-based simulation.

The implementation of our QP controller is based on the TSID framework [5] itself based on the Pinocchio rigid-body dynamics library [6].

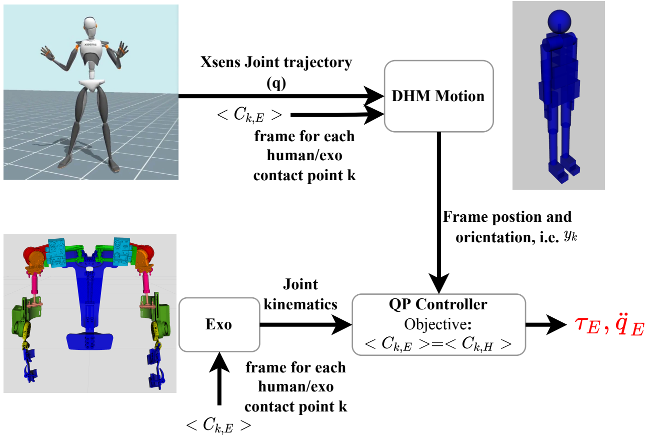
Xsens Joint trajectory (q)
$< C_{k,E} >$
frame for each human/exo contact point k
DHM Motion
Frame postion and orientation, i.e. y_k
Joint kinematics
Exo
QP Controller
Objective:
$< C_{k,E} >=< C_{k,H} >$
$\tau_E, \ddot{q}_E$
frame for each human/exo contact point k
$< C_{k,E} >$

Fig. 2. Flowchart of the control process, at each time step the process is repeated so that the new τ_E can be sent to the exoskeleton joints.

III. Results

The tracking performances can be assessed by measuring the distance between corresponding frames at each time step.

The results shown on Fig. 3 show the distribution of the distances between each corresponding contact points, for every time step of every simulation runs. These results were obtained when replaying MOCAP movements from one individual, doing 3 types of movements, 5 times each, hence 15 trials overall. The mean distance for these 15 trials is $1.55 \pm 1.3cm$. As can be seen on Fig. 3, the overall median is below 2 cm.

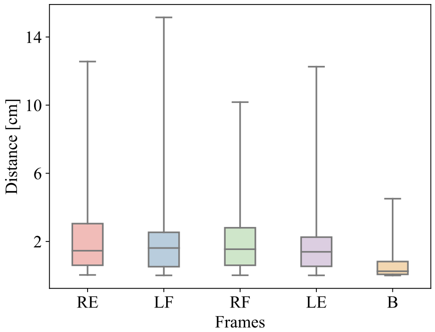

Fig. 3. Distance distribution between the corresponding contact points set on the DHM and on the exoskeleton at each time step for every trial.
LE, LF, RE, RF and B corresponding respectively to Left Elbow, Left Forearm, Right Elbow, Right Forearm and Back.

Some outliers can be as high as 15 cm, it happens especially when the movement is fast. This issue could be improved by adding a feedforward term in equation (2) (ie $\ddot{y}^{goal} \neq 0$).

Conclusion

In this work, we present a method to simulate an active upper body exoskeleton tracking a human movement, emulating a transparent mode. The simulated exoskeleton can be used to test controllers and to generate data.

In future work, it will allow acquiring a dataset of motion data from virtual sensors. This database will prove useful for the training of machine learning algorithms, that will be used in a predictive control method to control the real exoskeleton. This simulation should be improved, allowing the DHM and the exoskeleton to interact with each other, i.e. moving to a physics-based DHM, with its own controller. This will lead to the creation of better datasets and to the possibility of testing control strategies in the simulation, with some limitations (simulation to real gap, no human reaction to the interaction...).

References

[1] E. Schneider, X. Irastorza, E. Safety, S. Copsey, H. Verjans, L. Eeckelaert, V. Safety, and Belgium, *OSH in Figures: Work-related Musculoskeletal Disorders in the EU — Facts and Figures*.

[2] M. P. de Looze, T. Bosch, F. Krause, K. S. Stadler, and L. W. O'Sullivan, "Exoskeletons for industrial application and their potential effects on physical work load," vol. 59, no. 5, pp. 671–681.

[3] K. Chatzilygeroudis and J.-B. Mouret, "Robot dart." [Online]. Available: https://github.com/resibots/robot_dart

[4] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, "DART: Dynamic animation and robotics toolkit," vol. 3, no. 22, p. 500. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.00500

[5] A. Del Prete, N. Mansard, O. E. Ramos, O. Stasse, and F. Nori, "Implementing torque control with high-ratio gear boxes and without joint-torque sensors," in *Int. Journal of Humanoid Robotics*, 2016, p. 1550044. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01136936/document

[6] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.