



HAL
open science

Towards data-driven predictive control of active upper-body exoskeletons for load carrying

Alexandre Oliveira Souza, Jordane Grenier, François Charpillet, Pauline Maurice, Serena Ivaldi

► To cite this version:

Alexandre Oliveira Souza, Jordane Grenier, François Charpillet, Pauline Maurice, Serena Ivaldi. Towards data-driven predictive control of active upper-body exoskeletons for load carrying. International Conference on Advanced Robotics and its Social Impacts, Jun 2023, Berlin, Germany. 10.1109/ARSO56563.2023.10187548 . hal-04087475

HAL Id: hal-04087475

<https://hal.science/hal-04087475>

Submitted on 4 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Towards data-driven predictive control of active upper-body exoskeletons for load carrying

Alexandre Oliveira Souza^{1,2}, Jordane Grenier², François Charpillet¹, Pauline Maurice¹, Serena Ivaldi¹

Abstract—Upper-limb active exoskeletons are a promising technology to reduce musculoskeletal disorders in the context of load-carrying activities. To assist the user on time, it is crucial to predict the assistance torque required for the future intended movement. In this paper, we propose to predict such a torque with predictive models trained on simulated data. We generate exoskeleton sensor data for training learning-based prediction models from human motion capture data. We design a Quadratic Programming control problem for the exoskeleton to track the human body across its movements. From the data generated using this simulation method, we train two torque command prediction methods for transparent control and load carrying. We show that exoskeleton torque command can be predicted with a relative error below 5% at a horizon of 100ms.

Index Terms—Exoskeleton, Predictive control, LSTM

I. INTRODUCTION

Exoskeletons are a promising solution to reduce work-related musculoskeletal disorders [1] since they provide additional torque to assist users in their movements at work. Exoskeletons can be active or passive: the active ones are actuated, whereas passive ones usually use springs to restitute energy at a suitable moment of movement, making them useful for specific postures and tasks only. The main advantage of active exoskeletons is the higher amount of assistance they can provide, which makes them particularly interesting to assist humans in tasks involving high payloads and higher complexity of movements (such as agonist/antagonist sequences). However, their main challenge is to design controllers that assist the user as needed, detecting the user’s intent of motion and adapting to different payloads. Predicting the user’s intention is required by the exoskeleton to provide a suitable torque on time: any delay in the prediction may cause the exoskeleton to be perceived as a constraint in the motion rather than a help.

Electromyography (EMG) sensors are traditionally used to infer the human intention of motion. Since the excitation-contraction coupling takes delay and a considerable amount of electrical potential, EMG sensors can be used to detect the beginning of a movement in advance. But they are invasive, noisy, and hardly compatible with real-world exoskeletons worn on clothes. An alternative approach to predict user intention is to use data-driven models that use as input the onboard sensors on the exoskeleton (e.g., torque sensors, IMUs), but their use on exoskeletons is constrained by the fact that they require a lot of data to be trained. Acquiring data from a real exoskeleton worn by a human is difficult, expensive, and potentially unsafe if the controller has not yet been validated.

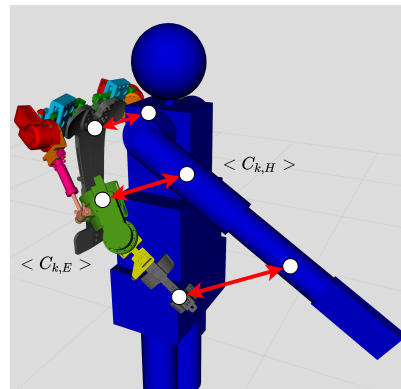


Fig. 1: An upper-body active exoskeleton “worn” by a virtual human. Contact points between the exoskeleton and the body are fed to a Quadratic Programming problem to simulate a transparent motion tracking. This method enables to generate synthetic exoskeleton data for data-driven prediction as well as test exoskeleton controllers in simulation.

Typically, in other areas of robotics, these problems are addressed by data augmentation with simulated data, or directly learning from simulated data. We posit that a similar approach should be applied to active exoskeletons. Therefore, we also need a physical simulator of the exoskeleton worn by a human that enables us to collect data, test and validate controllers.

This paper has two main contributions. First, we propose a simulation of the exoskeleton allowing the generation of virtual sensor data from abundant human motion capture data. We present a method based on Quadratic Programming (QP), where the exoskeleton tracks the human movement as if it were worn by the human, to generate a physical simulation of the exoskeleton (see Fig. 1). Second, we propose and compare two possible ways of predicting the joint torques necessary to assist a human in load carrying:

- 1) Predicting the future joint positions of the exoskeleton from past observations and calculating the corresponding joint torque using a dynamical model of the exoskeleton
- 2) Calculating the past torques from past observation of joint positions and predicting the future joint torque.

To predict these values, we choose to use LSTM-based neural networks for their known performance in predicting future motion [2]. We use the simulation method to generate the data necessary to train our two LSTM-based joint torque predic-

¹ Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

² Safran Electronics & Defense, Valence, France

tion methods and compare their performances in simulation. Our work practically demonstrates how physical simulation is necessary to prototype and study predictive controllers for active exoskeletons, before deployment on real platforms. We believe that it can accelerate the development of exoskeleton controllers, with benefits in the long term for workers.

II. RELATED WORK: HUMAN INTENTION PREDICTION FOR EXOSKELETON CONTROL

Detecting human intention usually requires reading biological signals or interpreting dynamic or kinematic signals. A popular approach for upper-limb exoskeleton control is to use EMG sensors to infer human motor intention and generate a command for exoskeleton control [3] [4] [5]. They can infer the human motion intention 50-100ms before the movement [6], which allows synchronizing the human movements and the exoskeleton assistance. However, using EMG sensors outside of the lab is challenging as it varies depending on experimental conditions such as temperature, sweat, or fatigue [6]. Control strategies of exoskeletons deployed in the real-world (logistics, service, civil...) should rely on less invasive sensors, ideally only on internal sensors of the robotic exoskeleton.

In lower-limb exoskeletons, instead of EMG sensors, it is standard to use Inertial Measurement Units (IMU), positional encoders, or force-sensitive insoles to detect the stride frequency of the user [7], [8]. Once the stride frequency is known, a control law can be defined as a function of the percentage of the stride. Human in the Loop optimization can improve metabolic performance and assist walking, both in the lab [9] and in the real world [10]. However, this technique heavily relies on the cyclic pattern of the human walk, which makes it hard to use for upper limb assistive exoskeleton control where actions are varied and do not follow cyclical patterns.

Another way of predicting for exoskeleton control is to use motion capture data and infer the future kinematic and dynamic behavior of the human from past observations. Several machine learning methods for motion prediction can be used such as Probabilistic Movement Primitives (PROMPS), Recurrent Neural Networks (RNNs) and Encoder-Recurrent-Decoder networks to predict short and long-term human motion from motion capture data and video [2], [11], as well as Generative Adversarial Networks (GANs) [12], [13]. To the best of our knowledge, few works use motion prediction methods to control upper-limb assistive exoskeletons, with few exceptions for an upper-limb rehabilitation exoskeleton [14].

III. METHODS

A. Problem formulation

Our goal is to find the torque command to send to the motor to control the exoskeleton, complying with the human's motion and compensating for a payload. We want the human to move as if they were not wearing the exoskeleton: so the exoskeleton should compensate for its weight and dynamics. The exoskeleton should not constrain human movements, so it should move in perfect "symbiosis" with them, which requires

predicting the human motion intention with accuracy. Finally, the exoskeleton should assist in carrying payloads, which requires knowledge of the payload itself. In the following, we will assume it is known. The torque commanded to the exoskeleton motors, τ_c , must then combine the torque to compensate for the exoskeleton's motion, τ_E and the torque to carry the payload, τ_P . We consider the inverse dynamics for our problem:

$$\tau_c = \tau_E + \tau_P \quad (1)$$

With

$$\tau_E = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (2)$$

and

$$\tau_P = J^T W_P \quad (3)$$

Where τ_c is the joint torque command, q, \dot{q}, \ddot{q} the exoskeleton's joint position, velocity, and acceleration; τ_P the contribution of the external payload; J^T the jacobian transpose of the exoskeleton expressed at the end effector; W_P the wrench applied by the payload; $M(q)$ the inertial matrix, $C(q, \dot{q})$ the Coriolis effects and $g(q)$ the gravity effects. To make the exoskeleton transparent to the user, we compensate for τ_E and for a part of the payload. The payload's contribution is calculated as an external force applied at the end-effector of the exoskeleton's arms, which is an approximation of the reality as all the dynamics of the load are not computed, just its mass contribution. We want to predict $\tau_{c_{t+1 \rightarrow t+h}} = (\tau_{c_{t+1}}, \tau_{c_{t+2}}, \dots, \tau_{c_{t+h}})$ the vector of future joint torques. As explained in section II, data-driven methods are classically used to make such predictions but require examples of past data. In the next sections, we detail how to generate such data with a simulated exoskeleton. The overview of the proposed method is presented in Fig. 2: to generate examples of the exoskeleton's motion, we define an optimization problem that makes the exoskeleton track a simulated human replicating human demonstrations of several tasks, recorded via motion capture. The dataset generated from this method is used to train predictive models. Here, we use models based on LSTM [14]. Online, the model is queried to generate predictions of future torques, according to two methods that will be shown in section III-C.

B. Generating synthetic exoskeleton data from motion capture

Motion capture data of humans are used as references for Quadratic Programming (QP) tasks, which allows animating the simulated exoskeleton moving as if it were worn.

1) *Exoskeleton Description*: We use an upper body active exoskeleton (Designed by Safran Electronics & Defense). It is composed of 4 active actuators, located on the shoulders and elbows joints (respectively in red and green in Fig. 1). There are multiple passive DoFs from the hips to the elbows, to ensure anthropomorphic kinematics with as few constraints as possible while moving. However, in our simulation, all the DoFs need to be controlled as the exoskeleton is not attached to a human that generates the movements of the passive DoFs.

In Fig. 1, we show how we extract from the motion capture data, the Cartesian trajectories of the expected contact points

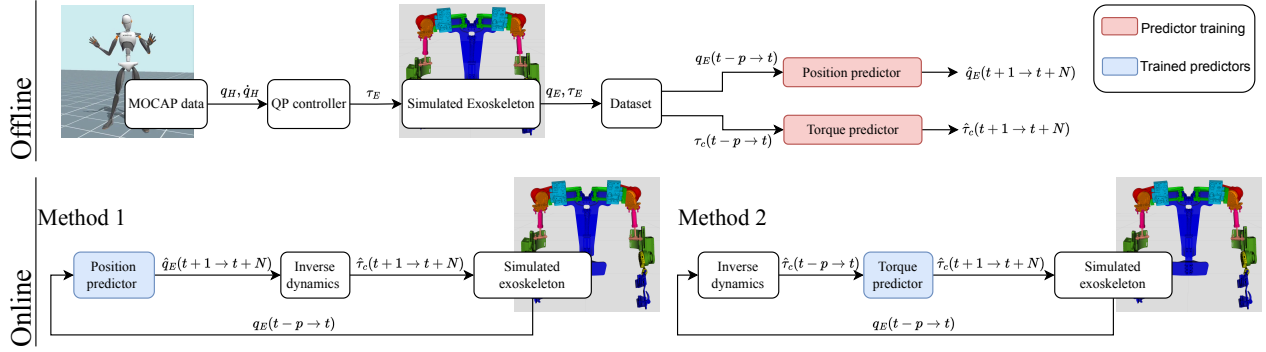


Fig. 2: Pipeline of the torque prediction methods, relying on simulated exoskeleton data. q and τ are respectively the joint positions and the joint torques of the human (H) or the exoskeleton (E), and τ_c the torque command.

between the exoskeleton and the user. For each contact point k , a frame is placed on the Digital Human Model (DHM) $\langle C_{k,H} \rangle$ and another frame on the exoskeleton $\langle C_{k,E} \rangle$.

2) *An optimization approach*: To mimic the transparent behavior of the exoskeleton, i.e., the exoskeleton following the human's movement, we formulate the exoskeleton control as a motion-tracking problem. The objective is to minimize the distance between the contact points on the exoskeleton and the trajectories extracted from the motion capture. The problem consists in solving at every time step of the simulation the following QP optimization problem:

$$\operatorname{argmin}_{\dot{q}, \tau} \sum_k w_k \|\ddot{y}_k - \ddot{y}_k^d\|^2, \quad (4)$$

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 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 with 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 :

$$\ddot{y}^d = \ddot{y}^{goal} + K_d(\dot{y}^{goal} - \dot{y}) + K_p(y^{goal} - y) \quad (5)$$

with y^{goal} the desired Cartesian position of the task.

At each time step, the position, and orientation of the contact points frames of the human (extracted, in our case, using the motion capture data) are given to the QP controller as an objective, i.e., the position and orientation of the human frame $\langle C_{k,H} \rangle$ are given as y_k^d and the position and orientation of the corresponding current state of the frame linked to the exoskeleton $\langle C_{k,E} \rangle$ are given to the controller as y_k , as explained in Fig. 2. The torque calculated from the resolution of the problem allows controlling the exoskeleton in the physics-based simulation. The motion and torque data from the simulation are saved to be used in the training phase.

C. Predicting the exoskeleton torque from past observations

In this section, we present two possible methods for predicting the exoskeleton's torque commands to assist the human. We remind that the exoskeleton must "follow" the human's intended movements and assist with the payload. The first method consists in predicting the future joint positions and feeding these predicted positions to the inverse dynamics to compute the torques. The second method consists in applying first the inverse dynamics on the joint positions, to compute the current and past torques, and then predict the future torque from the past computation. Note that this is similar to having "Virtual torque sensors" and computing predictions out of their estimation, which is often done in humanoid robotics [15].

We expect that the second method will give better results as the number of predicted values is smaller, since only the torques of the motor joints need to be predicted, whereas all the joint positions need to be predicted to perform the inverse dynamics.

1) *PJ+ID: Torque from joint position prediction and inverse dynamics*: Predicting the future torque of the actuated joints of the exoskeleton is decomposed into several subproblems: predicting the future angular position of all the joints of the exoskeleton, using inverse dynamics to compute the torques of the exoskeleton when no payload is added and adding the torque corresponding to the payload. The problem of predicting the future state of the exoskeleton's joints based on their history is formulated as follows:

$$\hat{q}_{E_{t+1 \rightarrow t+h}} = f(q_{E_{t-p \rightarrow t}}) \quad (6)$$

With q_E the angular position vector of the exoskeleton, f the function to determine, h the prediction horizon and p the amount of history used to make the prediction.

An LSTM-based model is used to predict the future angular position of the joints of the exoskeleton based on their past positions. The model is composed of a single-layer LSTM cell and a 3-layer fully connected analyzer. To compute the torques of the actuated joints, we need to predict the future angular positions of all the joints (passive and active). Thus, the input of the neural network is a matrix containing the history of all

the joints of size $(j \times p)$, with j the number of joints and p the amount of history.

The output of the neural network is a matrix of size $(j \times h)$ with h the prediction horizon to make the prediction. The neural architecture is voluntarily light because it is meant to be used on an embedded computer.

To get the torque from the joint angular positions, a numerical derivation is performed to get the angular velocity $\hat{q}_{E_{t+1 \rightarrow t+h}}$, and the angular acceleration $\hat{\ddot{q}}_{E_{t+1 \rightarrow t+h}}$. The Jacobian matrix is calculated using the predicted joint positions. Using Eq. 1 we get $\hat{\tau}_{E_{t+1 \rightarrow t+h}}$ the estimated future torque contribution of the exoskeleton and the future contribution of the load $\hat{\tau}_{P_{t+1 \rightarrow t+h}}$. Thus, we get the future joint torque trajectory over horizon h .

2) *ID+PT: Inverse dynamics and torque prediction:* Instead of predicting the joint's future positions, we can use the joint's past observations and calculate the past joint torques $\hat{\tau}_{c_{t-p \rightarrow t}}$ using the same method as in the previous section. From this calculated torque, we can predict the future joint torque $\hat{\tau}_{c_{t+1 \rightarrow t+h}}$. The input of the neural network is the torques from the four actuated joints, and the output is their future values over the next steps. In this case, the objective is to find the function f that solves:

$$\hat{\tau}_{c_{t+1 \rightarrow t+h}} = f(\tau_{c_{t-p \rightarrow t}}) \quad (7)$$

We use an LSTM-based model, similar in architecture to the model of PJ+ID. It predicts the future torque of the actuated joints of the exoskeleton based on their past values. The model is composed of a single-layer LSTM cell and a 3-layer fully connected analyzer. As we are directly predicting the joint torque, we only need to predict the torque of the actuated joints. Thus, the input of the neural network is a matrix containing the torque history of the motor joints of size $(j \times p)$, with j the number of joint torques and p the number of past timesteps taken into account to make the prediction.

The output of the neural network is a matrix of size $(j \times h)$ with h the prediction horizon to make the prediction.

IV. EXPERIMENTS

A. Dataset

The AndyDataset [16] is a dataset of motion capture movements acquired from 13 individuals performing a variety of industrial-like movements using the Xsens MVN system. It is composed of several trials of 6 industry-like activities performed successively, for each individual (carrying boxes, setting screws at different heights, untying knots. . .).

A DHM is used to replay motion capture data. The DHM is a rigid-body human model with 66 DoFs, inspired by humanoid robots [17]. Differently from other works, here the DHM is a non-physical kinematic puppet, used to replay motion capture data and to calculate interaction points with the exoskeleton, displayed in Fig. 1. The simulation engine used in this work is RobotDart, a C++11 robot simulator based on the DART physics engine [18], [19]. We run inverse dynamics in real-time from the URDF model of the exoskeleton and the angular position, velocity and acceleration of its joints.

The implementation of our QP controller is based on the TSID framework [20], which uses the Pinocchio rigid-body dynamics library [21]. For the QP parameters of Eq. 5, we consider $\dot{y}^{goal} = 0$, $y^{goal} = 0$, $K_d = 2\sqrt{K_p}$ and K_p a parameter that we manually tune to maximize the tracking performances.

We acquire a dataset of motion and torque values from the exoskeleton that is supposed to follow a human performing the movements from the AndyDataset. For each motion, we also acquire the value of the load contribution to the joint torque for different loads. It results in the calculation of torques corresponding to the same movements with weights, without influencing the motion. The dataset contains loads ranging from 0.0kg to 6.0kg with increments of 0.5kg. This dataset was obtained when replaying motion capture movements from the AndyDataSet from 6 individuals, doing 3 types of movements, 5 times each, hence 90 trials overall, corresponding to 10 minutes of movements for each individual.

B. Model training

The input of our neural networks are matrices of size $(j \times p)$; we set p corresponding to a history of 500ms. The j value is different for both methods: for the first method of prediction, all the joint's positions are predicted, thus $j = 20$. For the second method, only the actuated joint torques are predicted, so $j = 4$. The output of our neural networks are matrices of size $(j \times h)$ We choose to use a prediction horizon h corresponding to 100ms in the future, which corresponds to the upper bound of the predicting capacities of an EMG-based predictive controller for exoskeleton [22].

The dataset was divided into training, validation and testing datasets. The testing dataset contains 2 sequences of 2 minutes of motion for each individual, i.e 24min of movement. These 24 minutes are multiplied by the number of weights contained in the dataset, i.e 12 more sets of 24 minutes, i.e 312 minutes of motion for the testing.

C. Comparing prediction methods

The two methods are compared for motion with and without loads in two experiments.

In the first experiment, we compare the prediction methods for torque prediction without payload. We compare the torque relative error between the ground truth and the predicted values for each motor joint for the two methods.

In a second experiment, we compare the prediction methods for torque prediction with a load. We use testing datasets taking into account a known random load for the comparison. We compare the torque error between the ground truth and the predicted values for each actuated joint for each method.

V. RESULTS

A. Dataset quality

The quality of the dataset can be assessed by measuring the tracking performances estimated through the distance between corresponding frames at each time step. The mean distance for the 90 trials is $2.74 \pm 2.32cm$. As shown in Fig. 3, the

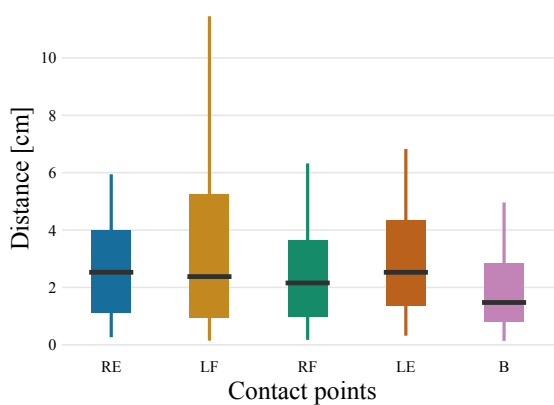


Fig. 3: Distance distribution between the corresponding contact points set on the DHM and the exoskeleton at each time step for every trial. The outliers are not displayed to increase readability and the whiskers correspond to 5% and 95% of the distribution (LE/RE=left/right elbow, LS/RS=left/right shoulder, B=Back).

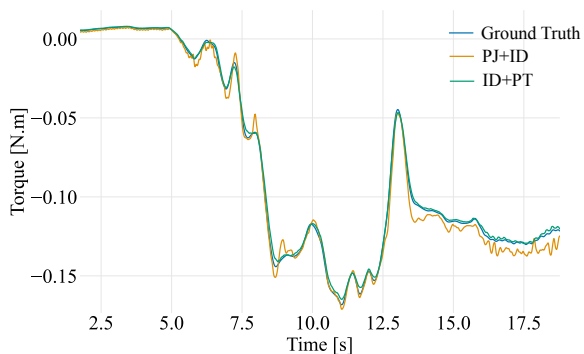


Fig. 4: Continuous prediction. Every 100ms, the past 500ms allow computing predictions of the future 100ms, for the left elbow motor joint. By concatenating the predictions, we get prediction curves that we can compare with the ground truth. (PJ+ID = Predicted Joint Positions + Inverse Dynamics, ID+PT= Inverse Dynamics + Predicted Torques)

overall median is around 2 cm, which means the exoskeleton is following the human but sometimes it makes a small "tracking error". In reality, this could be equivalent to feeling a constrained interaction (e.g. the exoskeleton pushing/pulling).

B. Results of the proposed prediction methods

We present the results of the proposed prediction methods using the dataset gathered from the AndyDataSet and described in the previous section. Fig.4 represents a continuous prediction for a test of 2 minutes of motion. The results of the first experiment are displayed in Fig. 5 where we show the distribution of relative errors compared to the ground truth. The results of this second experiment are displayed in Fig. 6 where we display the distribution of relative errors compared to the ground truth.

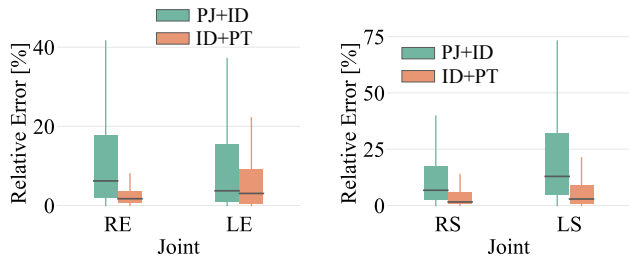


Fig. 5: Comparison of the distribution of the torque prediction errors of the different methods for each motor for a load of 0.0kg. The outliers are not displayed to increase readability and the whiskers correspond to 5% and 95% of the distribution (LE/RE=left/right elbow, LS/RS=left/right shoulder, PJ+ID = Predicted Joint Positions + Inverse Dynamics, ID+PT= Inverse Dynamics + Predicted Torques).

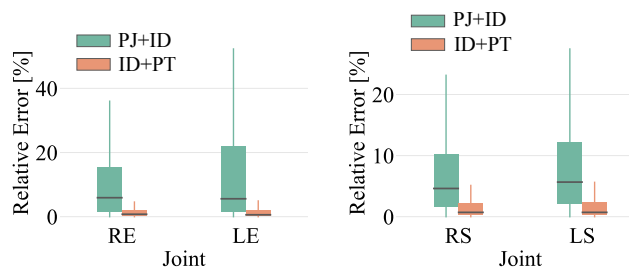


Fig. 6: Comparison of the distribution of the torque prediction errors of the different methods for each motor for a weight of 5.5kg. The outliers are not displayed to increase readability and the whiskers correspond to 5% and 95% of the distribution (LE/RE=left/right elbow, LS/RS=left/right shoulder, PJ+ID = Predicted Joint Positions + Inverse Dynamics, ID+PT= Inverse Dynamics + Predicted Torques).

VI. DISCUSSION

A. Generating synthetic exoskeleton data from motion capture

Generating synthetic data of exoskeletons worn by "virtual humans" is a fundamental challenge to enable to train data-driven prediction algorithms in an application where real-world data are scarce and expensive to obtain. Our method enables us to simulate an exoskeleton "attached" to a human and re-use existing human motion capture datasets, easier to obtain and readily available. The exoskeleton and the digital human are not physically coupled, thus human movements are not influenced by the exoskeleton, which is not the case in real life. In that respect, the dataset generated from this method corresponds to a human wearing an exoskeleton in a "perfect" transparent mode, i.e. without interaction between the human and the exoskeleton.

The quality assessment of the generated dataset enables to train neural networks for exoskeleton torque prediction, the tracking error being smaller than 5 cm 75% of the time for all the frames.

However, this method has several limitations: the tracking errors seem to come mostly from fast movements. The tracking of these fast movements could be improved by adding

a feedforward term in Eq. 5. Moreover, adding a physical human, and physical interaction with the exoskeleton could improve the data as the dynamics of the exoskeleton would impact human movements.

B. Torque prediction

We find that LSTM-based methods can predict torque from past observations with promising results. We compare two possible ways of predicting the future torque needed to control the exoskeleton. The predictions allow overcoming the latency between measurement and action, thus possibly achieving a better transparent control mode. We compare two methods of prediction and show that direct prediction of torque (ID+PT) gives better results than computing the torque associated with joint position prediction (PJ+ID) with external forces (Fig. 6) and without external forces (Fig. 5). These figures show that the ID+PT method has a median error smaller than 3% for all joints and a smaller standard deviation than the PJ+ID method. This difference comes most likely from the higher number of values to predict: for PJ+ID, 20 joint values need to be predicted, whereas in ID+PT only the torques of actuated joints are predicted. Moreover, PJ+ID undergoes two numerical derivations to get velocities and accelerations, which increases the error made in the predictions and leads to noisy torque predictions.

The main limitation of this method is that it relies heavily on the quality of the simulated model of the exoskeleton. Moreover, the model is trained on data that do not take into account interactions between the human and the exoskeleton. Similarly, the load does not modify the movements of the human, which would most likely be the case in reality.

VII. CONCLUSION

In this work, we developed and compared two techniques of torque prediction for exoskeleton control. These techniques rely on kinematic data, inverse dynamics and an LSTM-based neural network to predict the future joint torque needed to control an exoskeleton. We find that predicting the torque from past torque observations gives better results than calculating the torque from predicted motion based on past positions. To achieve these results, we implemented a physical simulation of the exoskeleton that attempts to track precisely the motion of a human, represented by a DHM. This technique has great potential to be used to test controllers and AI-based methods for exoskeletons before deployment on real platforms.

In future work, the possibility of using this technique should be assessed on a real exoskeleton robot. Fine-tuning of models with real data acquired from the exoskeleton will be studied. The reaction of a human wearing an exoskeleton powered by our controller will be investigated as our predictors are trained without knowledge of a possible reaction of the user.

REFERENCES

[1] 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," *Ergonomics*, 2016.

[2] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent Network Models for Human Dynamics," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[3] K. Kiguchi and Y. Hayashi, "An EMG-Based Control for an Upper-Limb Power-Assist Exoskeleton Robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2012.

[4] L. Wang and T. Buchanan, "Prediction of joint moments using a neural network model of muscle activations from EMG signals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2002.

[5] Q. Wu, B. Chen, and H. Wu, "Neural-network-enhanced torque estimation control of a soft wearable exoskeleton for elbow assistance," *Mechatronics*, 2019.

[6] L. Bi, A. G. Feleke, and C. Guan, "A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration," *Biomedical Signal Processing and Control*, 2019.

[7] M. Kim, Y. Ding, P. Malcolm, J. Speckaert, C. J. Sivi, C. J. Walsh, and S. Kuindersma, "Human-in-the-loop Bayesian optimization of wearable device parameters," *PLOS ONE*, 2017.

[8] R. L. Medrano, G. C. Thomas, and E. J. Rouse, "Can humans perceive the metabolic benefit provided by augmentative exoskeletons?" *Journal of NeuroEngineering and Rehabilitation*, 2022.

[9] J. R. Koller, D. H. Gates, D. P. Ferris, and C. David Remy, "'Body-in-the-Loop' Optimization of Assistive Robotic Devices: A Validation Study," in *Robotics: Science and Systems XII*, 2016.

[10] P. Slade, M. J. Kochenderfer, S. L. Delp, and S. H. Collins, "Personalizing exoskeleton assistance while walking in the real world," *Nature*, 2022.

[11] J. Martinez, M. J. Black, and J. Romero, "On Human Motion Prediction Using Recurrent Neural Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017.

[12] J. N. Kundu, M. Gor, and R. V. Babu, "BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[13] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3d human motion prediction via gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[14] J.-L. Ren, Y.-H. Chien, E.-Y. Chia, L.-C. Fu, and J.-S. Lai, "Deep Learning based Motion Prediction for Exoskeleton Robot Control in Upper Limb Rehabilitation," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[15] S. Ivaldi, M. Fumagalli, M. Randazzo, F. Nori, G. Metta, and G. Sandini, "Computing robot internal/external wrenches by means of inertial, tactile and F/T sensors: Theory and implementation on the iCub," in *IEEE-RAS International Conference on Humanoid Robots*, 2011.

[16] P. Maurice, A. Malaisé, C. Amiot, N. Paris, G.-J. Richard, O. Rochel, and S. Ivaldi, "Human movement and ergonomics: an industry-oriented dataset for collaborative robotics," *The International Journal of Robotics Research*, 2019.

[17] W. Gomes, P. Maurice, E. Dalin, J.-B. Mouret, and S. Ivaldi, "Multi-Objective Trajectory Optimization to Improve Ergonomics in Human Motion," *IEEE Robotics and Automation Letters*, 2022.

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

[19] 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," *Journal of Open Source Software*, 2018.

[20] 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.

[21] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiroux, 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.

[22] T. Lenzi, S. M. M. De Rossi, N. Vitiello, and M. C. Carrozza, "Intention-Based EMG Control for Powered Exoskeletons," *IEEE Transactions on Biomedical Engineering*, 2012.