



Estimation of Upper-Limb Joint Torques in Static and Dynamic Phases for Lifting Tasks

Hasnaa Ouadoudi Belabzioui, Charles Pontonnier, Georges Dumont, Pierre Plantard, Franck Multon

► To cite this version:

Hasnaa Ouadoudi Belabzioui, Charles Pontonnier, Georges Dumont, Pierre Plantard, Franck Multon. Estimation of Upper-Limb Joint Torques in Static and Dynamic Phases for Lifting Tasks. DHM 2023 - 8th International Digital Human Modeling Symposium, Sep 2023, Antwerpen, Belgium. pp.1-10. hal-04140086

HAL Id: hal-04140086

<https://inria.hal.science/hal-04140086v1>

Submitted on 24 Jun 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

Estimation of Upper-Limb Joint Torques in Static and Dynamic Phases for Lifting Tasks

Hasnaa Ouadoudi Belabzioui^{1,2*}, Charles Pontonnier¹ , Georges Dumont¹ ,
Pierre Plantard² , Franck Multon^{1,3} 

¹ Univ Rennes, Inria, CNRS, IRISA

² Moovency, Inc

³ Univ Rennes, M2S Lab

Abstract. In this paper, we propose learning architectures to emulate the inverse dynamics step in motion analysis. Indeed, the in situ motion analysis of a work situation is often based on noisy and/or incomplete motion data (video, depth camera...), requiring the development of methods robust to these uncertainties. Our study focuses on the development and evaluation on reference data (opto-electronic motion capture) of a torque estimation tool for upper limbs. The system was trained to estimate joint torques for static and dynamic one-handed load carrying tasks, based on the estimated position of the joint centers, the mass carried and the mass of the subject. The generalizability of our learning models was tested in inter-subject and inter-task scenarios. The average RMSE (N.m) and the average nRMSE (%) metrics were computed for each type of learning architecture. In a future work, we aim at emulating noisy data as an input of the problem to emulate in situ conditions and improve the robustness of the approach.

Keywords: neural networks, normalization, regression problem, inverse dynamics, sequential data, machine learning

1 Introduction

Evaluating in situ physical risk factors generally relies on noisy and/or incomplete motion data captured by videos or depth cameras. However, standard inverse dynamics cannot handle such inaccuracies and low frequencies to compute reliable joint torques. Instead, recent development in machine learning opens new possibilities to estimate these torques in such a difficult condition, robust to noise. However, designing such a machine learning approach requires extensive tests. In this paper, we propose a preliminary study to explore which machine learning approach would best approximate joint torques estimated with standard approach, firstly with noise-free data. This evaluation has been applied to one-handed load carrying tasks.

In [7], the authors used neural networks to estimate joint torques (namely M_x (N.m) and M_y (N.m) along the X and Y axes) at the L5-S1 joint during static

* hasnaa.ouadoudi-belabzioui@ens-rennes.fr

load handling. The trained algorithm maps the relationships between six features (Load location (X, Y, Z) (cm), hand load (Kg), body height (cm) and body weight (Kg)) and two targets (M_x (N.m) and M_y (N.m)) with a Root Mean Square Error (RMSE) of 16.5 N.m and a correlation of $R^2 = 0.97$. After normalizing the inputs and removing the outliers from static trials, the mean torques error decreased from 16.5 N.m to 11.8 N.m.

The estimation of joint torques in dynamic phases using learning algorithms is commonly performed based on the joint coordinates and velocities. [12] used Random forest approaches to estimate joint torques from motion parameters (joint coordinates and velocities) and acting forces for human gait. They also presented a weakly-supervised learning approach, aimed at inferring human dynamics (Ground Reaction Forces & Moments, and joint torques). To this end, they used an artificial neural network (NN), which architecture incorporated inverse and forward dynamics layers to minimize a pure motion loss. This motion loss consisted in minimizing the difference between the simulated motion generated by the model and the observed one, without using any additional information on the ground reaction forces, moments, or joint torques.

Human motion can also be described in terms of spatio-temporal information. Recently, some authors proposed to design specific NN architectures that consider spatial and temporal information separately [8] to estimate human posture.

Another key point when using machine learning approaches, is to ensure that the inputs have similar range of values. To address this problem, the input data are generally normalized before being processed by the method. This problem has been widely explored when designing action recognition methods [11].

Our preliminary study aims to estimate joint torques using skeletal data (reference data), subject mass, and load mass. We particularly propose two key contributions:

1. A comparison of state of the art machine learning methods to estimate the back and upper limb joint torques from 3D joint positions, the mass carried and the mass of the subject in static poses;
2. An evaluation of a NN architecture to deal with dynamic motions.

2 Overview

Inverse dynamics aims to determine the joint torques τ associated with external forces and motion quantities for a given motion performed by the subject. The movement is fully defined by the joint coordinates $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, and accelerations $\ddot{\mathbf{q}}$. The general inverse dynamics applied to a biomechanical model of the subject can be formulated as follows:

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

where $M(\mathbf{q})$ is the joint-space inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ is the joint-space Coriolis and centrifugal matrix, and $\mathbf{g}(\mathbf{q})$ is the joint-space gravitational vector, and

τ denotes the unknown joint torques. The software platforms, AnyBody [2], OpenSim [3], and the open-source CusToM MATLAB toolbox [9], integrate the inverse dynamics approach. In this study, we utilized the CusToM toolbox to determine joint torques based on a given motion, which served as the reference joint torques for the remainder of the paper. The joint torques estimated by the learning algorithms were compared to the joint torques computed by CusToM for the same motion. To this end, we devised learning architectures for the static and dynamic phases of a motion. The static phase was defined as a phase in which the subject remain still, whereas the dynamic phase was characterized by non negligible motion quantities. For static mass handling poses, we utilized a regression model to compare four architectures, namely, linear regression, decision tree, random forest, and neural network. In the dynamic phase, we first extracted spatio-temporal features from the skeleton data and then incorporated the subject’s mass and the mass of the load to regress the 16 joint torques of the right upper limb and the trunk.

3 Data collection and preparation

In this section, we detail the data used to train and evaluate the machine learning methods used to estimate the joint torques based on static or dynamic input motion data.

3.1 Experimental data and biomechanical model

Experimental data: The study used data from a previous experiment [5] that involved 11 right-handed subjects lifting and placing objects with varying masses, positions, handling height and timing. Each trial was composed of static phases and dynamic phases when displacing the carried object. An optoelectronic motion capture system Qualisys (23 12-Mpixels cameras, sampled at 200Hz) was used to record the motion of the subjects, and 2 force plates were used to record the ground reaction forces during the tasks. Three different loads were used (0Kg, 1Kg, 3Kg), with five final placing positions (175cm height in front, 175cm height on the right, 75cm height in front, 0cm height in front, and 0cm height on the right). By combining the load and the final placing conditions, it leads to $3 \times 5 = 15$ configurations that are repeated 9 times each per subject ($15 \times 9 = 135$ trials per subject). Among the total number of trials ($11 \times 135 = 1485$), 286 were discarded due to unexpected motions.

Static phases lasted 5.38 s in mean per trial, which corresponds to 1076 frames. Hence, a total of $949 \times (1485 - 286) \simeq 1M$ pose samples were available for static phases learning. Trials lasted 11 s in mean, corresponding to 2200 frames. Consequently, $2200 \times (1485 - 286) \simeq 2.6M$ samples were available for the dynamic phase training. As we apply a sliding window with an overlap of 1 and the length of each sequence is 5, the total number of sequences obtained was similar, $2.6M$ 5-frames sequences.

Biomechanical model: The biomechanical model was composed of eleven segments including the head, upper/lower trunks, left/right arms, left/right hands, left/right forearms, articulated with 38 degrees of freedom: a 6 dofs mobile base, and 31 anatomical joint angles following the International Society of Biomechanics recommendations. The joints included in the torque estimation are described in the table 1.

Joint	Corresponding exertion	Joint	Corresponding exertion
1	Lumbar spine flexion/extension	9	Right clavicle axial rotation
2	Lumbar spine lateral flexion/extension	10	Right shoulder plane of elevation
3	Lumbar spine axial rotation	11	Right shoulder depression/elevation
4	Trunk flexion/extension	12	Right Upper arm axial rotation
5	Trunk lateral flexion	13	Right elbow flexion extension
6	Trunk axial rotation	14	Right forearm pronation/supination
7	Right clavicle protraction/retraction	15	Right wrist flexion/extension
8	Right clavicle depression/elevation	16	Right wrist radial/ulnar deviation

Table 1: Joint torques estimated in the study. In particular, the 3 first torques correspond to the classical L5/S1 joint torques.

The CusToM toolbox was used to compute the reference joint torques using the biomechanical model described above.

3.2 Joint centers estimation and data normalization

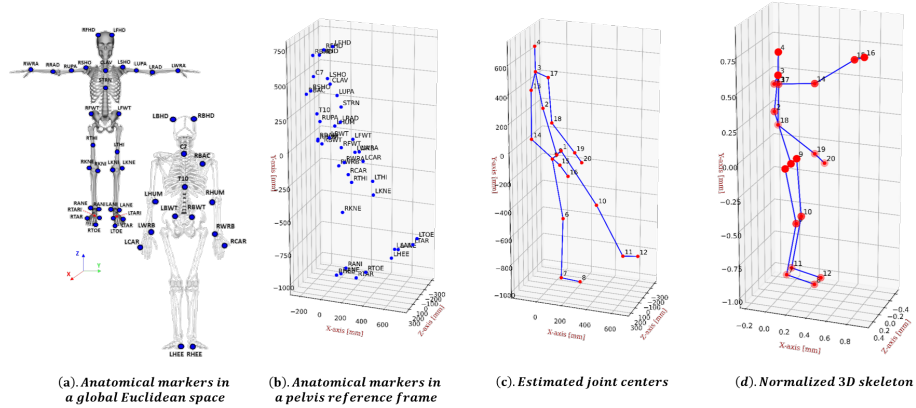


Figure 1: Marker coordinates were expressed in the pelvis reference frame. Joint centers were estimated using regression equations. The 3D human skeleton was normalized using the AABB (Axis-Aligned Bounding Box) approach.

3D anatomical markers positions in global Euclidean space were expressed in the pelvis reference frame. Joint centers were estimated based on the markers positions using regression equations [10, 6]. The resulting human pose (See Figure 1) was modeled as 20 3D joints $\mathbf{J}_{i=1..20}(t)$ at each time t , that can be gathered in a single vector denoted $\mathbf{J}$. 3D coordinate of each joint was normalized by its range of values, using the Axis-Aligned Bounding Box approach [1]. For the $k = [x, y, z]$ component of $\mathbf{J}_i(t)$, denoted $J_i^k(t)$, the normalized value was given by:

$$\hat{J}_i^k(t) = \frac{J_i^k(t) - \min(J_i^k(t))}{\max(J_i^k(t)) - \min(J_i^k(t))}; \quad \hat{m} = \frac{m - m_{min}}{m_{max} - m_{min}}; \quad \hat{M} = \frac{M}{M_{max}} \quad (1)$$

where $\min(J_i^k(t))$ and $\max(J_i^k(t))$ are the minimal and maximal values of J_i^k across all the subjects and all the trials, m_{min} and m_{max} are the minimum and maximum values of the load mass, and M_{max} is the maximum value of the subject mass respectively.

4 Joint torque estimation

In this section, we describe the machine learning approaches developed and evaluated in this paper to estimate the joint torques based on static or dynamic poses.

4.1 Static phases

Estimating the joint torques in a static phase is a particular case of the inverse dynamics problem. Specifically, this problem assumes that the joints velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ are negligible.

The objective of this estimation is to learn the function $\tau = f(\hat{\mathbf{J}}, \hat{m}, \hat{M})$ from the set of samples described in the previous section.

Four classical estimators were benchmarked in the current study to solve this issue: linear regression, decision tree, random forest, and neural network.

To apply the random forest learning, we set the number of trees of the random forest to 10.

The neural network architecture consists of an input layer, a dense layer with 64 units, batch normalization, ReLU activation, and a dropout rate of 10%. A second dense layer with L2 regularization is added to the network having a weight decay of 0.0001. This architecture results in 5072 parameters to optimize. In order to train our deep network f_θ , we adopted the standard mean squared error (MSE) loss function.

4.2 Dynamic phases

The objective of this estimation is to learn the function $\tau = f(G, \hat{m}, \hat{M})$ that maps the input features G (spatiotemporal features, see below), $\hat{m}$ and $\hat{M}$ to the output torques τ at a given time step.

In dynamic phases, velocities and acceleration are not negligible. Thus, we considered a time window of 5 frames prior to the estimation to extract spatiotemporal features G , by applying a one-dimensional convolutional layer to the $\hat{\mathbf{J}}(t_1), \hat{\mathbf{J}}(t_2), \dots, \hat{\mathbf{J}}(t_5)$ 3D joint positions gathered as a table. The layer had 64 filters with a kernel size of 2 and applied the Rectified Linear Unit (ReLU) activation function to its outputs. We also set the padding to 'valid', which means no padding is added to the input, and the stride to 1, which specifies the step size of the convolutional operation. Next, we applied a max pooling layer with a pool size of 2 and stride of 2 to the output of the convolutional layer. This reduced the dimensionality of the output by taking the maximum value in each 2-element segment. Then we applied a dropout rate of 25% to the output tensor from the previous max pooling operation.

Four distinct neural network architectures were tested to estimate the torques from the features: a Long Short-Term Memory (LSTM) layer, an LSTM layer sequentially followed by an attention layer, a Bidirectional LSTM (Bi-LSTM) layer, and a Bi-LSTM layer followed by an attention layer (See the figure 2).

To train these networks, we minimized the standard mean squared error (MSE) loss.

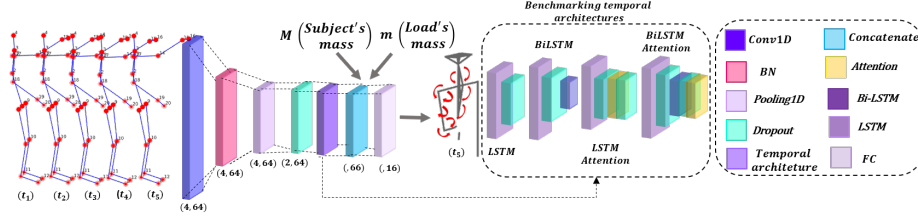


Figure 2: *Learning-based algorithm scheme of dynamic phase.*

The overall architecture used for extracting spatiotemporal features and regressing the 16 joint torques, had an average of 21232 parameters, which was high relative to the number of samples per parameter (See section 3.1). To prevent overfitting, a kernel regularizer with L2 weight regularization of 0.01 was added to the Conv1D layer, and a BatchNormalization layer was added after the Conv1D layer to normalize the output. Dropout regularization with a rate of 25% was added after the max pooling and LSTM layers, and kernel and recurrent regularizers with L2 regularization of 0.01 were added as arguments to the LSTM layer.

4.3 Learning and evaluation

We implemented our learning algorithms in static and dynamic phases using Keras/Tensorflow [4]. Training and validation were executed on an NVidia RTX A3000 GPU. We trained our learning algorithms through stochastic gradient

descent. The optimal model was obtained after 4000 iterations using an early stopping technique. The training was terminated when the loss did not decrease with a minimum delta of $1\text{e-}4$ and a patience value of 50 epochs. We used Adam optimization with a batch size of 64, learning rate $\alpha = 3 \times 10^{-5}$ and hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

We evaluated our learning models by computing the average RMSE and the average nRMSE, resulting from the Leave-One-Out procedure by subject and the Leave-One-Out procedure by task. The first procedure involves iteratively removing each subject from the training set and testing on that subject. The second procedure involves creating five groups of tasks [175cm, Right], [75cm, Front], [0cm, Front], [0cm, Right], [175cm, Front], with tasks iteratively removed from the training set for testing.

The quantitative evaluation of the regression results was done using the root mean squared error (RMSE) between reference and estimated joint torques, represented by ϵ and the normalized root mean squared error (nRMSE) represented by ϵ_r .

5 Results and discussion

Static phase Average RMSEs and standard deviations for the four considered architectures are given in table 2. In the inter-subjects scenario, the neural network algorithm achieved the lowest ϵ (5.54 N.m) and the lowest ϵ_r of 0.03%. In the inter-tasks scenario, the linear regression algorithm achieved the lowest ϵ (5.11 N.m) and the lowest ϵ_r of 0.05%. Even if the neural network achieved a better performance from one subject to one other, the results suggest that it may not generalize to other tasks. Such a behavior may be due to overfitting, that is a quite common issue for such a simple problem. Indeed, the static problem is linear and it was well estimated from the linear regression. The figure 3 shows the estimations made by the neural network on a test set consisting of trials from a subject that was not included in the training set, in both figures, we observe that the learning model encountered difficulties in accurately estimating the torques. The presence of postural variability in the test data in the 0cm trials may have further contributed to the model’s inaccuracy. Further investigation on the impact of postural variability on the model’s performance could be a topic for future research.

Dynamic phase Average RMSEs and standard deviation for the four considered architectures are given in table 3. We observed in inter-subjects scenario that all four algorithms have similar mean values of ϵ , with the lowest mean value and standard deviation achieved by the CNN-LSTM-Attention. This suggests that the CNN-LSTM-Attention algorithm was not only more accurate on average, but also more consistent in its performance across multiple runs. In inter-tasks scenarios the CNN-LSTM algorithm had the lowest mean value of ϵ (5.56 N.m). The CNN-BiLSTM-Attention algorithm has a slightly higher mean value of ϵ (5.85 N.m) but still performs reasonably well with a similar standard deviation

Algorithms	Inter-subjects scenarios		Inter-tasks scenarios	
	$\epsilon \pm \rho$ (N.m)	$\epsilon_r \pm \rho$ (%)	$\epsilon \pm \rho$ (N.m)	$\epsilon_r \pm \rho$ (%)
Decision Tree	8.53 ± 1.62	0.05 ± 0.01	6.80 ± 2.22	0.06 ± 0.02
Linear Regression	6.96 ± 1.49	0.04 ± 0.01	5.11 ± 0.87	0.05 ± 0.03
Neural Network	5.67 ± 0.91	0.03 ± 0.01	5.51 ± 0.67	0.06 ± 0.03
Random Forest	7.48 ± 1.43	0.04 ± 0.01	5.68 ± 1.66	0.05 ± 0.02

Table 2: Static phases Inter-Subjects Scenarios Results and Inter-Tasks Scenarios Results.

of 1.72 N.m. On the other hand, the two attention-based models, CNN-LSTM-Attention and CNN-BiLSTM-Attention, have higher mean values of ϵ (5.80 N.m and 5.81 N.m, respectively) and also have slightly higher standard deviations (1.58 N.m and 1.55 N.m, respectively). In general, these results suggest that the addition of attention mechanisms to the CNN-LSTM and CNN-BiLSTM models did not improve their performance on this particular task. It is noteworthy to mention that the variations in the algorithms' performance are comparatively minor, with a maximum difference of only 0.25 N.m between the best and worst performing algorithms. Both figures 3 show that the learning model for the dynamic phase predicts the static phases better compared to the model designed for the static phase.

Algorithms	Inter-subjects scenarios		Inter-tasks scenarios	
	$\epsilon \pm \rho$ (N.m)	$\epsilon_r \pm \rho$ (%)	$\epsilon \pm \rho$ (N.m)	$\epsilon_r \pm \rho$ (%)
CNN-LSTM	6.04 ± 1.72	$0.018 \pm 5 \times 10^{-3}$	5.56 ± 1.49	$0.015 \pm 5 \times 10^{-3}$
CNN-LSTM-Attention	5.73 ± 1.41	$0.017 \pm 5 \times 10^{-3}$	5.80 ± 1.58	$0.016 \pm 5 \times 10^{-3}$
CNN-BiLSTM	5.82 ± 1.72	$0.017 \pm 5 \times 10^{-3}$	5.66 ± 1.58	$0.016 \pm 5 \times 10^{-3}$
CNN-BiLSTM-Attention	5.85 ± 1.60	$0.017 \pm 5 \times 10^{-3}$	5.81 ± 1.55	$0.016 \pm 6 \times 10^{-3}$

Table 3: Dynamic phases Inter-Subjects Scenarios Results and Inter-Tasks Scenarios Results.

Conclusion This paper evaluates various machine learning approaches to estimate joint torques using skeletal data, subject mass, and load mass, as an alternative to standard inverse dynamics methods for future application to noisy and low-frequency data. The resulting mean torque RMSE for the corresponding L5/S1 moments (τ_2 and τ_3 , see table 1) were equal to 7.29 ± 2.24 N.m and 5.52 ± 1.41 N.m, respectively, for the static phases, and 10.70 ± 2.60 N.m and 5.74 ± 1.41 N.m, respectively, for the dynamic phases. These values are in line with the results of [7]. The results showed a better performance of the estimators used for the dynamic phases than those used for the static phases.

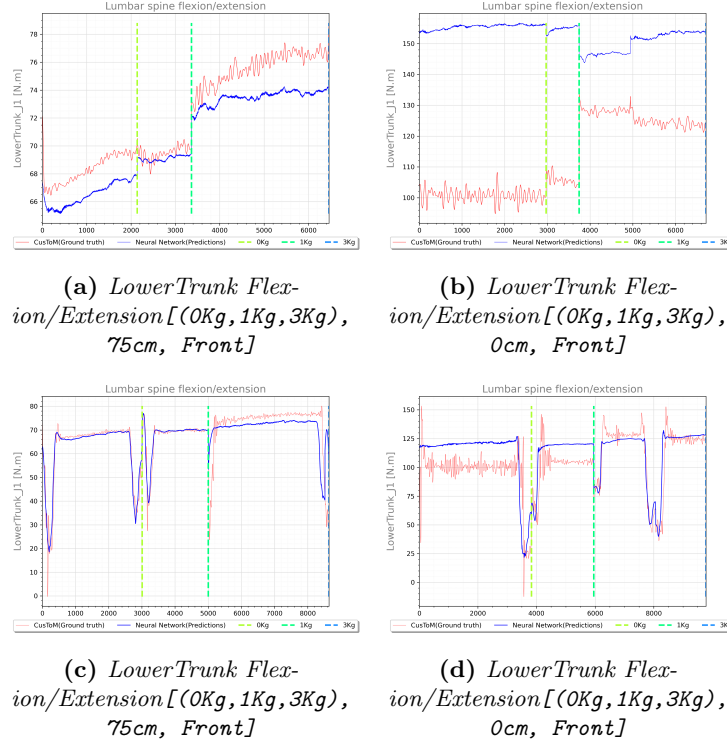


Figure 3: Sample trials. Computed joint torques (red line) and estimated joint torques (blue line) based on the test data using the neural network during (a-b) the static phase and CNN-LSTM-Attention during (c-d) the dynamic phase are represented.

Better performance in dynamic phases may be explained by higher torques in this dynamic phase. Following this first result, we may potentially significantly enhance the performance of our learning model CNN-LSTM-Attention by replacing the spatiotemporal features extraction component with a pre-trained model and by increasing the number of frames. As the inverse dynamics problem is determined by the equation of motion, we may potentially utilize this prior knowledge to employ Physics Informed Neural Networks (PINNs), which integrate known physical laws and principles into their architecture and training processes. A limitation of the machine learning approach is the impact of the quantity and quality of the data used for training. We can increase our training set by using simulated data. Our results tend to show that most of the methods tend to generalize to subjects (inter-subjects tests) that were not used during training, which is promising, and should be confirmed by more extensive tests. In a future work, we plan to incorporate Gaussian noise to evaluate the robustness of our various learning models in a real-time setting.

Bibliography

- [1] Chen, L., Qin, G.: Optimization of the collision detection technology in 3d skeleton animation. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). Volume 10., IEEE (2010) V10–539
- [2] Damsgaard, M., Rasmussen, J., Christensen, S.T., Surma, E., De Zee, M.: Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory* **14**(8) (2006) 1100–1111
- [3] Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G.: Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering* **54**(11) (2007) 1940–1950
- [4] Gulli, A., Kapoor, A., Pal, S.: *Deep learning with TensorFlow 2 and Keras: regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API*. Packt Publishing Ltd (2019)
- [5] Haj Mahmoud, O., Pontonnier, C., Dumont, G., Poli, S., Multon, F.: A neural networks approach to determine factors associated with self-reported discomfort in picking tasks. *Human Factors* (2021) 00187208211047640
- [6] Leardini, A., Benedetti, M., Catani, F., Simoncini, L., Giannini, S.: An anatomically based protocol for the description of foot segment kinematics during gait. *Clinical biomechanics* **14**(8) (1999) 528–536
- [7] Mohseni, M., Aghazadeh, F., Arjmand, N.: Improved artificial neural networks for 3d body posture and lumbosacral moment predictions during manual material handling activities. *Journal of Biomechanics* **131** (2022) 110921
- [8] Mourot, L., Hoyet, L., Le Clerc, F., Schnitzler, F., Hellier, P.: A survey on deep learning for skeleton-based human animation. In: *Computer Graphics Forum*. Volume 41., Wiley Online Library (2022) 122–157
- [9] Muller, A., Pontonnier, C., Puchaud, P., Dumont, G.: Custom: a matlab toolbox for musculoskeletal simulation. *Journal of Open Source Software* **4**(33) (2019) 1–3
- [10] Reed, M.P., Manary, M.A., Schneider, L.W.: Methods for measuring and representing automobile occupant posture. Technical report, SAE Technical Paper (1999)
- [11] Tang, W., van Ooijen, P.M., Sival, D.A., Maurits, N.M.: 2d gait skeleton data normalization for quantitative assessment of movement disorders from freehand single camera video recordings. *Sensors* **22**(11) (2022) 4245
- [12] Zell, P., Rosenhahn, B., Wandt, B.: Weakly-supervised learning of human dynamics. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI* 16, Springer (2020) 68–84