NeuroErgo: A Deep Neural Network Method to Improve Postural Optimization for Ergonomic Human-Robot Collaboration

Merikh Nejadasl, Atieh; Gheibi, Omid; Van de Perre, Greet; Vanderborght, Bram

Published in:
IEEE international conference on Robotics and Automation

DOI:
10.1109/ICRA46639.2022.9812460

Publication date:
2022

NeuroErgo: A Deep Neural Network Method to Improve Postural Optimization for Ergonomic Human-Robot Collaboration

Atieh Merikh Nejadasl[1,3], Omid Gheibi[2], Greet Van De Perre[1,4], Bram Vanderborght[1,4]

***Abstract*— Collaborative robots can help industry workers to improve their ergonomics. They can propose a safe and ergonomic posture to the workers to reduce the risk of musculoskeletal disorders. Proposing an ergonomic stance needs postural evaluation and optimization. To optimize the workers' posture, we need to run the optimization on a cost function representing the ergonomic status. The tabular ergonomic assessment methods are the most common methods used by ergonomists, but they are linear stepwise functions that are not differentiable and not suitable for optimization purposes. We propose NeuroErgo, a deep neural network model that can approximate the tabular ergonomic assessment methods more precisely than existing methods. By solving the task constraints optimization problem for any task in industry and NeuroErgo as posture cost function, a safe and ergonomic posture can be derived and recommended to the workers while accomplishing their job.**

***Index Terms*— Deep neural network, Ergonomic, Optimization, Human-robot collaboration, REBA**

I. Introduction

Human-robot collaboration is a growing line of research to reduce Work-related MusculoSkeletal Disorders (WMSD) [1], [2]. WMSDs are the most prominent reason for sick leaves among industry workers. The people who suffer from WMSDs may never return to their previous condition, reducing their performance and satisfaction in life [3]. From studies, we know that the most critical factor responsible for WMSDs is not ergonomically friendly repetitive tasks [4]. It seems crucial to consider industrial working conditions and workers' working requirements to reduce this compensation cost and increase industrial proficiency.

The studies show working in an ergonomic posture can reduce the risk of WMSDs, especially in tasks that need repetitive movements [5], [6]. However, achieving an ergonomic posture is not always trivial. It sometimes requires instruction or sacrificing the ergonomic posture to do the job quicker or more comfortably, but instant comfort may contradict an ergonomic posture.

Collaborative robots can help the worker to work ergonomically in many ways [7], [8]. They can be an assistant to evaluate, instruct, and remind an excellent posture to the worker while he/she is working. As a first step, the worker's body posture should be monitored by depth cameras. Then, the camera data is assessed by software that evaluates the ergonomic status of the worker's posture. This software judges the ergonomic condition of the posture based on the joints' degree or any other necessary data and a specific evaluation method. Eventually, ergonomic posture is proposed based on the optimization method to reduce the risk under some task constraints, such as the job to be done or the worker's body morphology. Thus, the collaborative robot changes the workpiece location to cause the optimized posture.

Many researchers attempted to tackle this optimization problem with different ergonomic assessment methods [9]–[12]. Among the ergonomic assessment methods, the standard methods like Rapid Entire Body Assessment (REBA) method [13] and Rapid Upper Limb Assessment (RULA) method [14] have been mostly utilized [12], [15] These studies fitted a second-order polynomial function on the tabular data of REBA and RULA to derive a differentiable model of the methods. Because the tabular methods are discrete functions and computationally make complicated optimization of problems in which utilized them.

However, polynomial estimation suffers from inaccuracy for approximating the tabular assessment methods. Thus, the problem statement of the current paper is:

> *How to approximate tabular ergonomic assessment methods more precisely as a differentiable function with applicability in task constraints optimization in ergonomic human-robot collaboration?*

As authors of [16] state, using feedforward neural networks is a fundamental method to approximate functions. On the other hand, we know from [17] that by adjusting *depth* and *width* of neural networks, we can increase the capability of the network to learn and approximate more complex functions. Consequently, due to the complex discrete structure[1] of the tables, it seems that we can obtain a more accurate continuous approximation of these tables by a deep[2] feedforward neural network.

Our novel method, called NeuroErgo, tackles this problem by learning the corresponding tables, REBA or RULA, using a deep feedforward network [16] in a pre-processing phase. Then, similar to existing methods such as [15] and [12], optimize the corresponding objective functions that are using the

[1] Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium, [2]Computer Science Department, KU Leuven, Belgium ,[3] Flanders Make,[4] imec, Belgium

[1]The REBA/RULA table's structure is hierarchical, and each hierarchy has multiple conditions. For example, if we express the REBA table's structure by a rooted tree, this tree will almost have 1.4 million separate paths from the root to the tree's leaves. (The counting the number of the different paths is straightforward by multiplying the number of parts for each body part.)

[2]Besides the high complexity of the tables and the necessity of a high capacity neural network to approximate them, the high depth requirement of the neural network will empirically be approved in Section IV based on the grid searching hyper-parameter tuning. As the high number of layers is suitable for low approximation error, we use "deep" as an adjective of the network in this paper.

result of the network as an approximation function. In other words, polynomials in those methods are replaced by the proposed network in NeuroErgo that gives a higher precision approximation from corresponding ergonomic metrics.

The rest of this article is as follows. In Section II, there is a literature review about the related works. Then, Section III gives an overview of the context of the problem. Afterward, in Section IV, the main idea of using deep neural networks for training the REBA tables is elaborated. In Section V, the experimental results are described to compare our proposed method versus the current estimation method based on two synthetic and real-world datasets. Finally, we conclude with some suggestions for future works in Section VI.

II. Related Works

Many researchers tackled human-robot collaboration for improving the ergonomic condition of industrial workers. In [10], [18], the main idea is to detect a non-ergonomic posture by measuring the joint overloading caused by an external load. In this method, after handling an external load by the industrial worker, the worker's body center of pressure is changed. This movement causes an external load to his/her joints. If the external load exceeds the allowed range, the collaborative robot changes the workpiece position to a better place to drive the reduction in this overloading. By measuring the movement of the center of pressure, authors succeed in getting rid of the dynamic modeling for estimating the risk of handling the object by workers.

In [19], the authors reduced the risk of WMSDs by proposing a generic algorithm for improving the ergonomics. This algorithm is a search-based optimization that reduces the ergonomic score measured by REBA. Then, it checks the reduction feasibility in each step until it reaches a point that more ergonomic score reduction is impossible.

These studies mentioned earlier only consider a single cause of WMSDs, i.e., the problem in the existing external load. Hence, the risk measurement is not happening in case of a sub-optimal stance, and the score could decline other possible causes of WMSDs.

In [9], the authors first assessed the posture of the industrial worker based on a personalized kinematic model. In case of danger in posture, the robot changes the place of the workpiece to be manipulated easier by the worker; correcting the pose is done by solving an optimization problem that minimizes the REBA score while accomplishing a specific task in an industrial environment. In this method, the author defined a variable called dREBA which is derived by fitting some second-order polynomial on the values of the REBA table. dREBA is differentiable and suitable to be used in the optimization problem for finding ergonomic posture. Although this polynomial method is novel and straightforward, it is inaccurate. Because in some situations, for the existence of some joints' degree, the REBA method specifies a score of one, and for not having that degree, the score is zero, e.g., shoulder raise in REBA tables. In these situations, the second-order polynomial is not a suitable fit. Note that this inaccuracy will also experimentally be justified later in Section V (particularly see Figure 2).

Authors of [12] used several ergonomic assessment methods to do the postural optimization. They used the RULA score, joints effort, shoulder, lumbar torque, and back flexion values for assessing the ergonomic condition of the worker's posture and did a multi-objective optimization for a task-specific problem in the industry. For this optimization, they fitted a second-order polynomial on the RULA tables data to differentiate from it in the optimization function for the RULA evaluation. Similar to [9], this study has a poor estimation of the RULA score.

Apart from related works to the human-robot collaboration-, some related works are studies that used machine learning methods to facilitate the ergonomic assessment procedure (the REBA and the RULA methods). In these methods, ergonomists calculate some of the specific joints' degree of workers, and by using some tables, evaluate the ergonomic status of the workers. To be able to use these methods in human-robot collaboration, many researchers tried to automate this process. In [20]–[22], the authors used a neural network and inverse kinematic to predict joints' angles of the workers from digital video snapshots or depth cameras like Kinect. Finally, the obtained joints' degree can be used for the RULA or the REBA assessment. These studies only covered the joints measurement part, and there is a need to improve the score estimation.

Moreover, from the machine-learning field, First, authors of [23] utilized a simple two-layer neural network for approximating non-linear objective functions in optimization tasks. Although the essence of their idea is the same as the current paper, their method handles straightforward scenarios and is not suitable for more complex cases in real applications. Second, a pre-printed paper [24] that attempted to apply neural networks to obtain a differentiable RULA and postponed REBA modeling as future work. Although the essence of their work is quite similar to NeuroErgo, they did not discuss the corresponding neural network's hyper-parameters selection, training, testing, data generation, and potential problems for ReLU activation function as it is not differentiable at point zero. Moreover, we cannot find the details of empirical results and comparison with existing methods such as dREBA [15].

III. Problem Context

Inspired by [9], in order to improve the human partner's ergonomic condition using human-robot collaboration, the corresponding objective cost function comprises a risk for the human's ergonomic condition, and task constraints that the human accomplishes should be minimized. Now, suppose we denote the ergonomic condition of the human body at time t by $q^{(t)}$, and denote the risk of the ergonomic condition and the cost of the particular task by $C_{posture}(q^{(t)}, t)$ and $C_{task}(q^{(t)}, t)$, respectively. Then, we can formalize the objective function for the collaboration, denoted by C_{total}, by a weighted sum of the two denoted cost functions like the following:

$$C_{total}(q^{(t)}, t) = w_1 \times C_{posture}(q^{(t)}, t) + w_2 \times C_{task}(q^{(t)}, t) \quad (1)$$

In this definition, weights of w_1 and w_2 are domain-oriented and determined based on the importance of each type of cost, i.e., the risk of the human's ergonomic condition and the cost of the task.

Now, based on the specified objective function (C_{total}), we can define the aim of the specified human-robot collaboration at time-step t; minimizing the objective function at time t, based on an optimum human's body joints' position. In formal definition, we can write the following minimization:

$$\begin{aligned} \min_{q^{(t)}} \quad & C_{total}(q^{(t)}, t) \\ \text{s.t.} \quad & \text{body joints' motion ranges.} \end{aligned} \quad (2)$$

Solving the above aggregated optimization problem will propose an ergonomic posture for accomplishing a specific task. By different definitions of task constraints and cost functions, the method can suggest the ergonomic posture for various jobs in an industrial environment. This ergonomic posture can be proposed to the worker to remember using the lowest risk posture while doing their job.

After giving an overview of the human-robot collaboration, we elaborate on details of the risk of the human's body ergonomic condition and the cost of the task in the following.

A. Ergonomics

To score the risk of the human's ergonomic condition , we use the REBA assessment method [13]. It is a tabular pen and paper method that assesses human posture in six steps. In each step, based on the joints' degree of neck, trunk, legs, upper arm, lower arms, and wrists including their possible flexion and extension, a score is assigned; finally, by putting together all these partial scores, a final score, an integer value between 1 to 15, shows the ergonomic status of the body. The final score which is higher than a specific value, shows a danger in posture, and if the worker continues to stay in that posture, there will be a risk of musculoskeletal disorders in the future. However, the REBA score below that value shows an ergonomic and safe posture that needs no change or correction.

As mentioned in the previous paragraph, the final REBA score results from some tables that their rows and columns values are related to the partial REBA scores, which are the values gathered by measuring the different parts of the joints' degree of the worker. These values cannot merely be expressed by a well-known continuous form of mathematical functions, such as polynomials, due to existing hierarchical conditions in the tables' definition .

Moreover, the REBA score as a step-wise linear function is not differentiable. Consequently, this non-differentiability will be a barrier to solving the optimization problem of Equation 2. To that end, as suugested in [9], the idea is approximating the REBA score by a differentibale function, D. Therefore, we define $C_{posture}$ in C_{total} by the following:

$$C_{posture}(q^{(t)}, t) = D(q^{(t)}) \quad (3)$$

A simple suggestion for the function D has been proposed by [9]. The suggested function is a sum of the weighted polynomial functions, called differentiable REBA (dREBA), fitted on REBA tables. The number of polynomials equals the number of human's joints considered in the standard REBA evaluation (see inputs Figure 1). Each polynomial is a second-order polynomial as the function of the corresponding joint's value.

Note that in the above equations, as we utilize the REBA assessment method, the human's ergonomic condition $q^{(t)}$ is interpreted as a set of 21 variables $q_1^{(t)}$, $q_2^{(t)}$, ..., $q_{21}^{(t)}$ that are the body joints' degree and their possible flexion and extension at time-step t (more details in Section IV). Hence, we characterize the "body joints' motion ranges" in the optimization problem of Equation 2, based on the allowable ranges defined by [25] for a human body, and we instantiate the equation like the following:

$$\begin{aligned} \min_{q^{(t)}} \quad & C_{total}(q^{(t)}, t) \\ \text{s.t.} \quad & q_1^{(t)} \in [-60°, 30°], q_2^{(t)} \in [-54°, 54°], \ldots, \\ & q_{21}^{(t)} \in [-90°, 90°] \end{aligned} \quad (4)$$

B. Task constraints

For tasks in industry, there is usually a workpiece or tool located in a specific location, and the worker should reach that tool while doing the job. We consider the workers' body as a serial chain of arms, upper body, lower body, and legs. The worker's hand should be positioned in a desirable location, i.e., instrument location; therefore, the goal here is to minimize the Euclidean distance between the forward kinematics of the body denoted by $FK(q^{(t)}, t)$ and the desired forward kinematic denoted by FK_{des}, which is the instrument location. Thus, the cost of the task at time t for joints' position $q^{(t)}$ is defined as the following[3]:

$$C_{task}(q^{(t)}, t) = \|(FK(q^{(t)}, t) - FK_{des}\| \quad (5)$$

To enlighten the problem context, all details of the optimization problem in Equation 2 have been explained. Accordingly, we are ready to restate the problem statement of the current paper.

C. Problem statement

The state-of-the-art methods such as [15] and [12] suggest a quadratic function as of approximation function D. Although their suggestion is differentiable, it is not accurate enough. The consequent problem is "*how to propose a differentiable method that is more accurate than the existing approximation method and applicable at runtime by a feasible computation time?*"

To tackle the problem, in the following section, we describe NeuroErgo as a differentiable model for approximating REBA tables to be in place of the function D in equation 6.

[3]Our implementation for FK function is accessible via this GitHub repository: `https://github.com/VUB-RMM/NeuroErgo`

IV. Methodology

Based on [16], we know that deep forward networks (DFNs) as a type of deep neural networks (DNNs) aim to approximate a function. On the other hand, tabular ergonomic assessment methods, e.g., REBA and RULA, use a table to score a set of specified body joint angles. Hence, we can identify these tables as a non-linear function applying over the body joint angles. Therefore, the main idea of NeuroErgo is learning these tables based on many analytical generated data for the future task optimizations (Equation 4).

To elaborate the idea, *we instantiate NeuroErgo for the REBA assessment method*, without loss of generality.

A. REBA approximation by DNN

Figure 1 represents the complete topology of the DNN for learning REBA tables, i.e., an instantiaition of NeuroErgo on REBA tables. The network's input layer size is 21, which equals the number of different degrees of freedom of human's joints[4], as mentioned in the previous paragraph. In this way, each element of the input is meaningful in terms of the REBA score. For an instance of $q_1, q_2, \dots, q_{21}$ in the input layer, the output layer returns the approximated total REBA. Note that the activation function for all neural units is not the same. In the following paragraphs, we elaborate on the rationale behind this topology and its hyper-parameters.

1) Topology: The first obvious fact in the represented topology is that the network is not fully connected. The reason behind this fact is the hierarchical structure of the REBA method. As found in [13], the score of each part of the body, e.g., neck and leg, is computed locally first. Then, the total REBA score is computed based on those local REBA scores. Accordingly, the same implemented logic in this topology can be observed.

Based on this logic, local networks in Figure 1 aim to compute the REBA score for each part, and the goal of the aggregator network is learning how to compute the total REBA score based on the output of the local networks. Note that each part's q_is do not affect the local REBA score of other body parts. Therefore, connecting unrelated q_is to other input neurons could increase the number of training variables and cause late convergence. Besides, these connections could not help to boost the performance of the network in terms of accuracy. At this point, a feasible question is whether we can take advantage of this locality to get performance in training the proposed DNN?

2) Training: To achieve an accurate model, we need to train the whole network with as many data samples as possible. However, as the order of data sample is 10^9 (explained in the following paragraph), training the whole network from scratch to reach an acceptable accuracy level can be very time-consuming. To resolve this issue, as a rational heuristic in the initial step, we train local networks based on the corresponding features, e.g., for the neck's local network, using features of $\{q_1, q_2, q_3\}$ from the full feature set $\{q_1, q_2, \dots, q_{21}\}$]. Then, based on the achieving weights for local networks, we train the whole network (with aggregator network) with the complete set of features, i.e., $\{q_1, q_2, \dots, q_{21}\}$]. Now, we know how the network's training method is, but what about the required data for doing it?

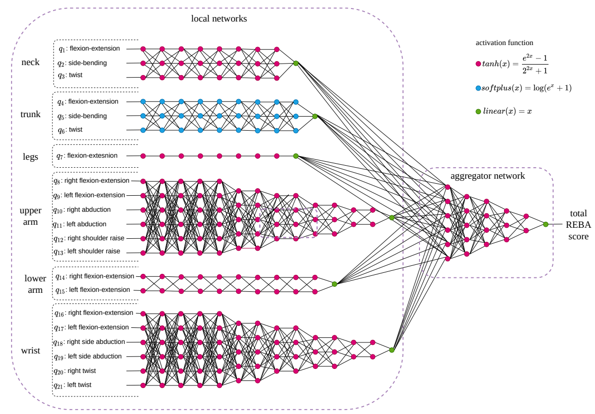

Fig. 1. Complete topology of the DNN for learning REBA tables, as an instance of NeuroErgo.

3) Data: Despite most typical applications of DNNs that need real-world collected data, e.g., image processing applications, we have the generator for training and testing data. In other words, REBA tables play the supervision role, and we do not need any extra captured and collected datasets. However, REBA tables [13] have been defined based on a sub-range of q_is (input features). For example, for the leg joint, a partial REBA score of 0, 1 or 2 is assigned, for a joint angle situated in, respectively, the interval of [0, 30], [30,60] and [60,130].

Thus, we need to sample these ranges for generating the required data. Although increasing the number of samples can improve the model's accuracy, if we select only ten values for each q_i, we will have 10^{21} data samples that required powerful hardware to train the whole network in a reasonable time.

A heuristic for data sampling is including at least one sample from each range in REBA tables such that the REBA score is shifted in each of these ranges. Including the ending points of these ranges can also be informative for the learner, e.g., for the leg joints, these ending points are 0, 30, 60, and 130 degrees. Note that we can increase the samples' granularity with less pressure on data size for training local networks (as the matter of vigorous initialization for the whole network [26]).

In more details, for training our proposed DNN, we used about 10^6 data samples for local networks and in order of 10^9 samples for training the whole network[5]. Now, we have

[4]It is assumed that both feet are all attached to the ground, and the person will not walk. Consequently, both knees will have the same angle. The mentioned hypothesis will not reduce the value of the current work as most cases in the industry need a well-shaped posture of the human. Even for safety reasons, people should keep in mind to put their feet on solid ground and stand firmly moreover most cases in industry person is standing in front of a work cell placed on the table and we only care about the upper body.

[5]Note that for training the DNN by a dataset of RGB-D images of the human partner with different poses, we need at least 1 billion images with enough variants. However, providing this collection is very time- and resource-consuming. This issue indicates the value and the applicability of the data generator.

all the required materials to build the network. However, returning to the topology paragraph, we have many options for network hyper-parameters, e.g., neurons' activation function, the number of layers in each local network and the aggregator, and the number of neurons in each of these layers. How should we select optimum options among the existing hyper-parameters' value space?

4) Hyper-parameters selection: Our general strategy for hyper-parameter optimization is grid searching, i.e., exhaustive searching, besides some heuristics to narrow the search space. These heuristics are 1) Based on [17], [27], [28], depth and width of DNNs[6] can affect their convergence rate, learning power, and capacity. Hence, we should cover a legitimate range of depth and width for options of each network topology. 2) We aim to minimize the difference between the network result and REBA score for the input data. Mean squared error can accordingly be a logical differentiable choice for the loss function.

After grid searching on the specified options, we achieved an optimal set of hyper-parameters for each network[7].

5) Utilization: Ultimately, we finished the last step of building the network, and all required details of the DNN have been elaborated. Now, for use it in optimization tasks (Equation 4), we need a method for taking gradient the trained DNN with respect to its input variables q_is (elements of $q^{(t)}$ in Equation 4). This problem has been solved and enhanced through the years [16], [29], what is called Back-Propagation (BP). It is an efficient backward derivative method that is worked based on the derivative chain rule. Accordingly, using a recent BP algorithm for DNNs [16], we can efficiently take the derivative of the trained network proposed by the current paper, based on its input variables q_is. Therefore, we propose the following instantiation for $C_{posture}$ in Equation 6:

$$C_{posture}(q^{(t)}, t) = NE(q^{(t)}). \tag{6}$$

In the above equation, $NE : \mathbb{R}^{21} \to \mathbb{R}$ is a DNN differentiable function proposed by NeuroErgo.

V. Validation

As mentioned in Section I, a popular method for REBA tables approximation (for function D in Equation 6) is dREBA that fits two degree polynomial curves for each table [12], [15]. Thus, we compare the empirical results of NeuroErgo with dREBA in the followings.

A. Evaluation setup

To evaluate NeuroErgo, we used two types of dataset. First, a synthetic dataset with one million valid random body postures was created by the data generation explained in Section IV-A. Second, a real-world dataset [30] which is humans' postures from 10 people (aged between 20 and 35 years) who were asked to do 14 different activities like grabbing an object from the ground or taking an object from a shelf. Meanwhile, their joint position and orientation have been captured by the Microsoft Kinect V2 as a data acquisition device. To utilize this provided data for validation, we have extracted joints' angle information from the joints' pose data by some geometric calculation.

Moreover, to investigate the application of the method on the human-robot collaboration scenario, it is assumed that each human posture is a set of kinematic chains connected consecutively. By applying forward kinematic on the overall human chain, we can calculate the position of the human hands as the end effector of the overall chain. The hand's position is the place that the workpiece is placed. The new angles that are the output of the optimization algorithm[8] result in a new feasible position of the workpiece that may differ from the previous workpiece's position (cf. Equation 4). In this step, a collaborative robot can transfer the workpiece to the new place for better handling by the worker.

To evaluate NeuroErgo, we used Python 3.7. Also, dREBA method and its evaluation have been developed in Matlab because of the required symbolic computation in this method. We use the Franka Emika Panda robot arm to validate the methodology as a collaborative robot that can change the workpiece position as stated earlier. The Orocos Kinematics and Dynamics Library [32] is used for kinematic inversion to give the workpiece pose to the robot. The robot is modeled in Gazebo simulation, and the motion between the initial and final location of the item is planned in Moveit [33] employing Robot Operating System (ROS) as middleware. The details of motion planning and the robot control are beyond the scope of this article. Nonetheless, their implementations used for validation is available online.[9]. Moreover, all these implementations run on an i7-3770 @ 3.40GHz processor with 16GB RAM. Note that to analyze the noise-resistance level of the methods, we added unbiased uniform noise in the range of $[-1, 1]$ into this dataset.

B. Evaluation goals

The evaluation's main purpose is to clarify the performance of NeuroErgo against existing polynomial methods for approximating the REBA score by a differential function. As a result, the evaluation sub-goals are representing: 1) distribution of true errors, i.e., the absolute difference between the true REBA score and the approximated score for dREBA and NeuroErgo based on the generated validation data, 2) statistical parameters of these distributions such as mean and standard deviation, 3) a proper statistical test to prove the significant difference between these two distributions, and 4) the resistance level of each method under a noisy data capturing situation.

[6]Depth and width of a neural network mean the number of layers and the number of neurons in each layer, respectively.

[7]Details of these optimal hyper-parameters represented in `https://github.com/VUB-RMM/NeuroREBA`.

[8]In this optimization algorithm, we use GradientTape in Tensorflow to compute the derivative of the NeuroErgo's DNN for specific input and use LocalSolver [31] (under license number 009256) as a BlackBox optimizer to find a minimum gradient point of the objective function under the specified body joints' movement constraints.

[9]`https://github.com/VUB-RMM/NeuroErgo`

C. Evaluation results

1) Absolute error: As the first empirical result, Figure 2 represents the distribution of absolute errors on each datset, i.e., the absolute difference of the predicted and true value of the REBA score of each validation data, for NeuroErgo and dREBA. This figure indicates that NeuroErgo surpasses dREBA in terms of the Mean Absolute Error (MAE) on both datasets. (The more detail of the MAE with its standard deviation can be found in Table I.)

2) Running time: Although the proposed network can be established over GPU to get faster results, the difference of the prediction time for these two models, i.e., NeuroErgo and dREBA, is negligible on CPU. Note that the training time is not a serious issue here; the training is done only once as a pre-processing task.

3) Inversion: As the main application of this approximation is in optimization problems, another informative metric is *inversion*, i.e., the number of times that a lower score is approximated higher than the approximation of a higher score [34]. In a formal language, suppose two joint vectors $[q_1, q_2, \ldots, q_{21}]$ and $[q'_1, q'_2, \ldots, q'_{21}]$ denoted by q and q', respectively. We also denote the REBA score of these two body joints by r and r' such that $r < r'$. Take an approximation method that estimates the REBA score of these two vectors by $\tilde{r}$ and $\tilde{r}'$. Now, in the case of $\tilde{r} > \tilde{r}'$, we call an inversion occurred if $r < r'$. Accordingly, the percentage of inversion among the one million validation samples is 31.8 % for NeuroErgo and 32.6 % for dREBA approximation method. We can also observe a significant improvement for NeuroErgo in the percentage of inversions on real-world data set (2.8 % and 23.0 % for NeuroErgo and dREBA, respectively).

4) Statistical inference: These statistics and median errors in Figure 2 shows decidedly that NeuroErgo is far superior to dREBA. However, we should justify it statistically. As these distributions are independent and not normal, we use "Mann–Whitney U" test [35] for this aim. Here, the null hypothesis is that "randomly drawn sample from the dREBA's error distribution is more significant than a randomly drawn sample from the NeuroErgo's error distribution". By running the test on each dataset, we can accept this null hypothesis by confidence 1.0 and 0.95 on synthetic and real-world datasets, respectively (with p-value 0.0 and 0.05, respectively). Therefore, we could justify that NeuroErgo gives us a much more reliable approximation of the REBA score than dREBA.

5) Measuring noise-resistance level: As sensors for measuring joint angles can suffer from noise, a valid question about these approximation errors seems to be on how much they are noise resistance. To answer this question, we used the noisy real-world dataset (explained in the evaluation setup). Then, we observed its effect on the error distribution of each method (error distibtuions of methods on the normal real-world dataset in Figure 2). As a result, we obtained the same error distribution with more than 99.99 % confidence, and with less than 0.1 change in the mean and the standard deviation that have been reported for the real-world dataset in Table I. (The expected noise effects on shifting the error values of NeuroErgo and dREBA in real-world dataset are 0.18 ± 0.44 and 0.14 ± 0.41, respectively.)

In sum, we can get an insight that both methods seem to be resistance to the unbiased capturing joints' angle noise.

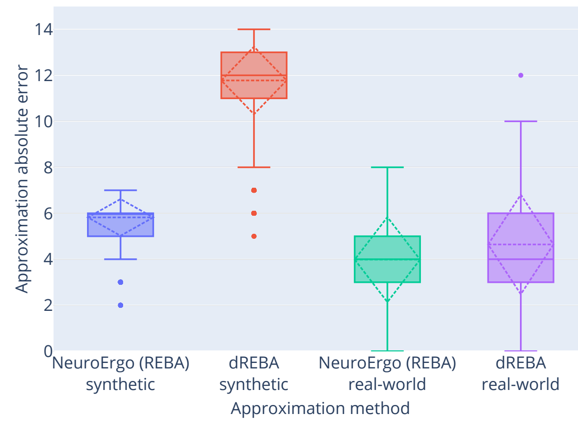

Fig. 2. Distribution of absolute errors for NeuroErgo and dREBA based on the synthetic and the real-world datasets. The horizontal line indicates the mean of absolute errors and extreme vertical points of the dotted diamond show the standard deviation of the errors.

TABLE I

MEASURED METRICS FOR VALIDATION ON SYNTHETIC DATASET

		Metric Synthetic		Metric Real-world	
		MAE	Inversion	MAE	Inversion
Method	NeuroErgo	$\mathbf{5.8 \pm 0.8}$	**31.8 %**	$\mathbf{4.0 \pm 1.9}$	**2.8 %**
	dREBA	11.8 ± 1.5	32.6 %	4.6 ± 2.2	23.0 %

VI. CONCLUSION AND FUTURE WORKS

This paper presented a deep neural network approximation method called NeuroErgo for modeling tabular ergonomic assessment methods. To show its applicability and performance, we instantiated the method for approximating the REBA score, which is mostly used for ergonomic evaluation. The proposed model was compared to another widely used existing literature method, and it showed a significant improvement in approximating the REBA values. The proposed model is simply differentiable with the well-known back-propagation method to be applicable in human-robot collaboration for ergonomic optimization.

The performance of the instantiated model for the REBA approximation can be enhanced by increasing the computation power and number of training data. Moreover, applying other topologies of DNNs for training the tables can be promising. Consequently, future work is the model's performance enhancement.

ACKNOWLEDGEMENT

This work was supported by Flanders Make ErgoEyeHand, EU SOPHIA (871237) and by the Flemish Government under the program "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen".

REFERENCES

[1] V. Villani, F. Pini, F. Leali, and et al., "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[2] A. Ajoudani, P. Albrecht, M. Bianchi, and et al., "Smart collaborative systems for enabling flexible and ergonomic work practices [industry activities]," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 169–176, 2020.

[3] L. Punnett and D. H. Wegman, "Work-related musculoskeletal disorders: the epidemiologic evidence and the debate," *Journal of electromyography and kinesiology*, vol. 14, no. 1, pp. 13–23, 2004.

[4] N. M. Nur, S. Z. Dawal, and M. Dahari, "The prevalence of work related musculoskeletal disorders among workers performing industrial repetitive tasks in the automotive manufacturing companies," in *Proc. of the 2014 Inter. Conf. on Industrial Engineering and Operations Management Bali, Indonesia*, 2014, pp. 1–8.

[5] K. K. Lett and S. M. McGill, "Pushing and pulling: personal mechanics influence spine loads," *Ergonomics*, vol. 49, no. 9, pp. 895–908, 2006.

[6] J. P. Jansen, M. J. Hoozemans, A. J. van der Beek, and M. H. Frings-Dresen, "Evaluation of ergonomic adjustments of catering carts to reduce external pushing forces," *Applied ergonomics*, vol. 33, no. 2, pp. 117–127, 2002.

[7] S. A. Elprama, C. I. Jewell, A. Jacobs, and et al., "Attitudes of factory workers towards industrial and collaborative robots," in *Proc. of the Companion of the 2017 ACM/IEEE Inter. Conf. on Human-Robot Interaction*, ser. HRI '17. ACM, 2017, p. 113–114.

[8] I. El Makrini, S. A. Elprama, J. Van den Bergh, and et al., "Working with walt: How a cobot was developed and inserted on an auto assembly line," *IEEE Robotics Automation Magazine*, vol. 25, no. 2, pp. 51–58, 2018.

[9] B. Busch, G. Maeda, Y. Mollard, and et al., "Postural optimization for an ergonomic human-robot interaction," in *2017 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2778–2785.

[10] W. Kim, L. Peternel, M. Lorenzini, J. Babič, and A. Ajoudani, "A human-robot collaboration framework for improving ergonomics during dexterous operation of power tools," *Robotics and Computer-Integrated Manufacturing*, vol. 68, 4 2021.

[11] L. Peternel, C. Fang, N. Tsagarakis, and et al., "A selective muscle fatigue management approach to ergonomic human-robot comanipulation," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 69–79, 2019.

[12] W. Gomes, P. Maurice, E. Dalin, and et al., "Multi-objective trajectory optimization to improve ergonomics in human motion," *IEEE Robotics and Automation Letters*, 2021.

[13] S. Hignett and L. McAtamney, "Rapid entire body assessment (reba)," *Applied ergonomics*, vol. 31, no. 2, pp. 201–205, 2000.

[14] L. McAtamney and E. N. Corlett, "Rula: a survey method for the investigation of work-related upper limb disorders," *Applied ergonomics*, vol. 24, no. 2, pp. 91–99, 1993.

[15] B. Busch, M. Toussaint, and M. Lopes, "Planning ergonomic sequences of actions in human-robot interaction," in *2018 IEEE Inter. Conf. on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1916–1923.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[17] T. Nguyen, M. Raghu, and S. Kornblith, "Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth," *arXiv preprint arXiv:2010.15327*, 2020.

[18] W. Kim, J. Lee, L. Peternel, and et al., "Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, pp. 68–75, 1 2018.

[19] A. Merikh-Nejadasl, I. El Makrini, G. Van De Perre, and et al., "A generic algorithm for computing optimal ergonomic postures during working in an industrial environment," *Inter. Journal of Industrial Ergonomics*, vol. 84, p. 103145, 2021.

[20] A. Abobakr, D. Nahavandi, J. Iskander, and et al., "A kinect-based workplace postural analysis system using deep residual networks," *2017 IEEE International Systems Engineering Symposium (ISSE)*, pp. 1–6, 10 2017.

[21] M. MassirisFernández, J. Álvaro Fernández, J. M. Bajo, and et al., "Ergonomic risk assessment based on computer vision and machine learning," *Computers and Industrial Engineering*, vol. 149, 11 2020.

[22] A. Abobakr, D. Nahavandi, J. Iskander, and et al., "Rgb-d human posture analysis for ergonomie studies using deep convolutional neural network," *2017 IEEE Inter. Conf. on Systems, Man, and Cybernetics (SMC)*, pp. 2885–2890, 10 2017.

[23] G. Villarrubia, J. F. De Paz, P. Chamoso, and et al., "Artificial neural networks used in optimization problems," *Neurocomputing*, vol. 272, pp. 10–16, 2018.

[24] A. Yazdani, R. S. Novin, A. Merryweather, and et al., "Ergonomically intelligent physical human-robot interaction: Postural estimation, assessment, and optimization," *arXiv preprint arXiv:2108.05971*, 2021.

[25] "Generally accepted values for normal range of motion (rom) in joints," https://www.verywellhealth.com/what-is-normal-range-of-motion-in-a-joint-3120361, accessed: 2010-09-30.

[26] J. Y. Yam and T. W. Chow, "A weight initialization method for improving training speed in feedforward neural network," *Neurocomputing*, vol. 30, no. 1-4, pp. 219–232, 2000.

[27] G. Pandey and A. Dukkipati, "To go deep or wide in learning?" in *Artificial Intelligence and Statistics*. PMLR, 2014, pp. 724–732.

[28] Y. Shin, "Effects of depth, width, and initialization: A convergence analysis of layer-wise training for deep linear neural networks," *arXiv preprint arXiv:1910.05874*, 2019.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[30] A. Franco, A. Magnani, and D. Maio, "Joint orientations from skeleton data for human activity recognition," in *Inter. Conf. on Image Analysis and Processing*. Springer, 2017, pp. 152–162.

[31] T. Benoist, B. Estellon, F. Gardi, and et al., "Localsolver 1. x: a black-box local-search solver for 0-1 programming," *4or*, vol. 9, no. 3, pp. 299–316, 2011.

[32] R. Smits, "KDL: Kinematics and Dynamics Library," http://www.orocos.org/kdl.

[33] D. Coleman, I. Sucan, S. Chitta, and et al., "Reducing the barrier to entry of complex robotic software: a moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.

[34] H. Mannila, "Measures of presortedness and optimal sorting algorithms," in *Inter. Colloquium on Automata, Languages, and Programming*. Springer, 1984, pp. 324–336.

[35] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.