Real-Time Voluntary Motion Prediction and Parkinson's Tremor Reduction Using Deep Neural Networks

Anas Ibrahim, *Graduate Student Member, IEEE*, Yue Zhou, *Member, IEEE*, Mary E. Jenkins, Ana Luisa Trejos, *Senior Member, IEEE*, and Michael D. Naish, *Member, IEEE*

***Abstract*— Wearable tremor suppression devices (WTSD) have been considered as a viable solution to manage parkinsonian tremor. WTSDs showed their ability to improve the quality of life of individuals suffering from parkinsonian tremor, by helping them to perform activities of daily living (ADL). Since parkinsonian tremor has been shown to be nonstationary, nonlinear, and stochastic in nature, the performance of the tremor models used by WTSDs is affected by their inability to adapt to the nonlinear behaviour of tremor. Another drawback that the models have is their limitation to estimate or predict one step ahead, which introduces delay when used in real time with WTSDs, which compromises performance. To address these issues, this work proposes a deep neural network model that learns the correlations and nonlinearities of tremor and voluntary motion, and is capable of multi-step prediction with minimal delay. A generalized model that is task and user-independent is presented. The model achieved an average estimation percentage accuracy of 99.2%. The average future voluntary motion prediction percentage accuracy with 10, 20, 50, and 100 steps ahead was 97.0%, 94.0%, 91.6%, and 89.9%, respectively, with prediction time as low as 1.5 ms for 100 steps ahead. The proposed model also achieved an average of 93.8% ± 1.5% in tremor reduction when it was tested in an experimental setup in real time. The tremor reduction showed an improvement of 25% over the Weighted Fourier Linear Combiner (WFLC), an estimator commonly used with WTSDs.**

***Index Terms*— Parkinson's disease, pathological tremor, deep neural networks, tremor prediction, voluntary motion prediction, tremor estimation.**

I. Introduction

PARKINSON'S Disease (PD) is a disorder that affects the nerve cells in the deep parts of the brain called the basal ganglia and the substantia nigra [1]. More than ten million people worldwide are living with PD [2], [3]. PD signs and symptoms include tremor that is time varying, nonlinear, stochastic, and nonstationary in nature [4]. Tremor significantly affects and impacts people's mobility and their ability to perform activities of daily living (ADLs) [5]–[8].

The current approach for the treatment of PD tremor is medication and deep brain stimulation (DBS) [9], [10]; however, medication is connected with a series of adverse effects and DBS carries significant risks [10] and could cost up to $100,000 per patient [3].

Recently, assistive technologies, such as wearable assistive devices, have been considered and are being developed as an alternative approach to manage tremor [11]–[21]. The mechanical suppression of tremor by these devices serves to assist individuals with tremor perform ADLs. However, the performance of the assistive devices is directly impacted by the tremor models that they use for control [8]. The models use adaptive filtering techniques to estimate tremor. Initially, the Fourier Linear Combiner (FLC) [22] was developed and used. The FLC operates under the assumption that tremor can be represented as a periodic signal that can be modelled by a Fourier series; however, Timmer *et al.* showed that tremor is not periodic but rather time varying, nonlinear, and stochastic. The FLC works using a predefined frequency, and it is not able to adaptively track tremor frequencies. The Weighted-Frequency Fourier Linear Combiner (WFLC) was proposed to overcome the drawbacks of the FLC. The WFLC is a commonly used method in tremor estimation [23], which adapts the frequency and amplitude of the model to the tremor signal. Similar to the FLC, the WFLC assumes that tremor is sinusoidal in nature, and it is unable to extract periodic signals

Manuscript received November 4, 2020; revised May 10, 2021; accepted July 9, 2021. Date of publication July 13, 2021; date of current version July 26, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Canadian Institutes of Health Research (CIHR) through a Collaborative Health Research Projects (CHRP) under Grant 396234, in part by the Canadian Foundation for Innovation (CFI) through the Ontario Research Fund (ORF), in part by the Ontario Ministry of Economic Development, Trade and Employment through the Ontario Ministry of Research and Innovation through the Early Researcher Award, and in part by the Peter C. Maurice Research Fellowship in Biomedical Engineering. *(Corresponding author: Michael D. Naish.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Health Science Research Ethics Board of Western University under Approval No. 106172.

Anas Ibrahim is with the Department of Electrical and Computer Engineering (ECE), Western University, London, ON N6A 5B9, Canada (e-mail: aibrah43@uwo.ca).

Yue Zhou is with the School of Biomedical Engineering (BME), Western University, London, ON N6A 5B9, Canada (e-mail: yzhou426@uwo.ca).

Mary E. Jenkins is with the Movement Disorders Program, Clinical Neurological Sciences, Western University, London, ON N6A 5B9, Canada (e-mail: mary.jenkins@lhsc.on.ca).

Ana Luisa Trejos is with the Department of Electrical and Computer Engineering (ECE), Western University, London, ON N6A 5B9, Canada, and also with the School of Biomedical Engineering (BME), Western University, London, ON N6A 5B9, Canada (e-mail: atrejos@uwo.ca).

Michael D. Naish is with the Department of Mechanical and Materials Engineering, Western University, London, ON N6A 5B9, Canada, also with the Department of Electrical and Computer Engineering (ECE), Western University, London, ON N6A 5B9, Canada, and also with the School of Biomedical Engineering (BME), Western University, London, ON N6A 5B9, Canada (e-mail: mnaish@uwo.ca).

Digital Object Identifier 10.1109/TNSRE.2021.3097007

containing more than one dominant frequency. As a result, it cannot adapt to the frequency of the higher harmonics of tremor, since the model uses Fourier series, and the frequency parameters of the higher harmonics are set proportional to the estimated frequency of the dominant harmonic. Therefore, it cannot be used to accurately estimate tremor.

The Band-limited Multiple Fourier Linear Combiner (BMFLC) is an algorithm that is based on the FLC. The BMFLC was developed to address the shortcomings of the WFLC. The limitation of the BMFLC is that it works with a predetermined set of frequencies and is unable to accurately track and adapt its frequency to tremor frequencies. Studies have shown that the calculated errors tend to be high [24], because the BMFLC is not robust to noise. Thus, if there is noise within the frequency band, the BMFLC will add the noise to the estimation as well. As a result, the Extended-BMFLC (EBMFLC) [25] was developed to overcome these limitations. It showed promising results in both the accuracy and consistency of extracting tremor signals; however, the tremor is still considered as a series of sine and cosine signals, thus, the adaptive filtering methods face challenges in estimating and extracting tremor accurately.

One common limitation of these estimators is the inherent time delay that they introduce for tremor control in wearable assistive devices [8], [26], [27]. Studies have shown that a delay as small as 20 ms degrades the performance of wearable devices [26], [27].

In an attempt to overcome the drawbacks of the FLC and its derivatives and to increase tremor suppression, Taheri *et al.* [20] proposed an algorithm that uses torque to suppress tremor, showing promising results. Taheri *et al.*, extended their work in [21] to estimate the muscle torque that produced the tremor instead of estimating tremor, and then applied an equal and opposite torque to suppress tremor. Other research has focused on predicting future steps of tremor in an attempt to improve tremor suppression; however, Veluvolu *et al.* [27] demonstrated that integrating autoregressive models, the WFLC, and the BMFLC with a Kalman Filter (KF) to perform one-step prediction of tremor, still resulted in poor performance. The accuracies achieved were as low as 40% for up to 20 ms of future prediction. Recently, the WAKE framework [28] was proposed, based on wavelet decomposition and adaptive Kalman filtering; however, it is limited to estimation and one-step prediction. Another framework, PHTNet has been recently proposed by Shahtalebi *et al.* [29]. It is limited to the prediction of only one step ahead, and it introduces a time delay of 100 and 150 ms before it can properly estimate the signal. This delay is detrimental when applied to real-time control, as it will negatively affect the performance of the wearable assistive devices that manage tremor.

Due to the challenges and limitations presented in the previous work, deep neural networks (DNN) [30] were considered. DNNs have shown their robustness in learning from data, and high prediction accuracies in different areas including robotics and assistive devices that use motion data as their input. As such, in order to effectively suppress tremor with wearable assistive devices and overcome the time delay problem, this work implements a novel approach based on a one-dimensional Convolutional-Multilayer Perceptron model (1D-CNN-MLP) for multistep prediction. The robustness of the 1D-CNN-MLP model for voluntary and tremor time series prediction and estimation was shown in a previous study by the authors [31], where it was demonstrated that the 1D-CNN-MLP model was able to learn correlations between past and present events, and future events. Given a time series, such as tremor signals and voluntary motion, the 1D-CNN-MLP model reads a string of numbers that represent tremor or voluntary motion (e.g., acceleration, velocity, or position) and predicts the number that is most likely to occur next.

In this paper, the performance of the proposed 1D-CNNMLP model was evaluated and compared to four known DNN architectures that are based on Recurrent Neural Networks (RNN) to find the best architecture that could be used for WTSDs. Even though the networks have never been used for WTSDs, nor for multi-step prediction, the RNN-based architectures have been recognized and shown to be robust when used with time-series data [32]–[36]. The RNN-based architectures are the Long–Short Term Memory (LSTM) [37], the Gated Recurrent Unit (GRU) [38], the Bidirectional LSTM (BLSTM), and the Bidirectional GRU (BGRU) [39]. While the 1D-CNN-MLP has been shown to be robust to time-series data, it also requires less computational power and performs automatic feature extraction, as will be discussed in Section II. As such, the 1D-CNN-MLP was compared to the RNN-based architectures to find the best model to use with WTSDs. The performance of the 1D-CNN-MLP was then evaluated and compared to the Weighted Fourier Linear Combiner (WFLC), which is commonly used with WTSDs.

Therefore, the contribution of this work is a model that can learn directly from data, can adapt to the various tremor frequencies and amplitudes, and can learn the nonlinear relationships between voluntary motion and tremor in order to increase tremor suppression when used with WTSDs. Another contribution of this work is a DNN model that can predict 10, 20, 50, and 100 steps ahead with high accuracy, and low prediction time.

II. One Dimensional Convolutional-Multilayer Perceptron and Recurrent Neural Networks

A Convolutional Neural Network (CNN) is one type of Neural Network (NN) architecture that takes advantage of the concept of convolutions to learn higher-order features [31], [40], [41]. CNNs are mainly used for image classification and object recognition. They are also used for complex tasks such as sound and text recognition [42]–[44]. CNNs are known for their automatic feature extraction process, where they learn an internal representation of n-dimensional data. The same process can be exploited for one-dimensional sequences of data, such as time-series data of hand motion that are collected using Inertial Measurement Units (IMU). In this work, the angular velocity data from the three axes (x, y, and z) were used in the training and testing sets; however, the prediction was performed on a single axis. CNNs learn to map the internal features extracted from sequences of observations to follow a motion sequence. A sequence of convolutions are performed in the first convolutional layer. The sum of convolutions is passed through the activation function *f*, followed by a sub-sampling operation before passing the one-dimensional sequences of data to the following layers to learn and extract the features. Herein, the output of the last CNN layer was used in the prediction task that is performed by the Multi-Layer Perceptron (MLP) layer [31] that estimates and predicts voluntary motion. Due to the nonlinear, nonstationary, and stochastic nature of PD tremor, a hybrid of a one-dimensional CNN (1D-CNN) and a MLP network were considered based on the previous work proposed by the authors [31], as shown in Fig. 1.

The LSTM architecture was introduced by Hochreiter and Schmidhuber [37] to solve the vanishing and exploding gradients problem that RNNs face. The concept of LSTMs relies on memory cells and gated units that regulate the flow of information, and enables them to learn the important data from the input sequence to make a prediction. The LSTM cell consists of a forget gate, an update gate, an output gate, and a memory cell [37]. The memory cell carries relevant information throughout the processing of the input sequence, and information is added to and removed from the memory cell through the different gates. The forget gate decides which information should be kept and which information should be removed from the previous states. The update gate decides which information should be updated, and which information is important to keep from the current step. The output gate decides what the prediction is and what the hidden state should be.

The GRU architecture was recently introduced by Cho *et al.* [38]. Similar to LSTMs, GRUs consist of gate units but do not contain memory cells. The GRU architecture is simpler than that of the LSTM, as it only consists of a reset gate and an update gate. The reset gate allows the GRU cell to drop any information that is considered to be irrelevant in the future, thus, allowing for a more compact representation of data. The update gate, on the other hand, controls how the information from previous hidden states carries over to the current hidden state. The update gate acts similarly to the memory cell in the LSTM network. It allows the network to remember long-term information, and decides whether the hidden state is to be updated with a new hidden state.

The BLSTM and BGRU work by training the network simultaneously in the positive and negative time direction. In other words, the network will run from the past to the future, and from the future to the past [39].

LSTMs and GRUs are able to preserve information from both the past and the future at any point in time due to their hidden states, which makes them robust for time-series data analysis. The RNN-based networks have complex architectures that require more computation and memory to hold the information learned, while the 1D-CNN-MLP requires less computational power and memory storage than the LSTMs and GRUs, which leads to a faster training process.

Regardless of the limitations that the RNN-based architectures have, which are their complexity and their slower training process, they have been shown to be robust with time-series data. Moreover, the 1D-CNN-MLP model is a new approach that has been shown to be robust with time-series data [31]. As a result, the five architectures were considered and compared to find the best architecture to be used with WTSDs.

III. Methods

To develop a voluntary motion estimator and a multi-step predictor, a two-step method was followed. The first step presents a comparison between the five different neural network (NN) architectures. The best architecture was then used in a simulation study that shows the model's performance when predicting voluntary motion. In the second step, the model with the best performance in simulation is used in a real-life experimental setup, where one degree-of-freedom (DOF) motion data were reproduced using a DC brushless motor. A second motor was used to track and predict the voluntary motion using the best model from the first step. The goal of this study is to find a method that is based on neural networks, that can learn directly from data to estimate and predict voluntary motion when used with WTSDs to increase tremor suppression. The method was evaluated to show the performance of the predictor on voluntary motion up to 100 steps ahead.

The following subsections present the data collection and preparation, the implementation and accuracy measures, and the comparison of the five DNNs.

A. Data Collection

A total of 18 subjects with PD participated in this study. The data were recorded at the Wearable Biomechatronics Laboratory at Western University [5]. The participants were recruited by a movement disorders neurologist, and the study was approved by the Health Sciences Research Ethics Board (#106172). The PD group included eleven males and seven females with varying tremor severity. Their ages ranged from 60 to 84 (mean ± standard deviation is 69 ± 7). 16 out of the 18 participants were on medication (Levodopa, Pramipexole, and Amantadine), but did not stop taking their medicine before they arrived at the lab for data recording. Furthermore, the tremor scores from the MDS-UPDRS Motor Exam Part 3 were recorded, and they are as follows: for Item 15, the mean score was 1.56 ± 0.92. For Item 16, the mean score was 0.83 ± 0.71. For Item 17, the mean score was 2.72 ± 0.67. For Item 18, the mean score was 3.61 ± 0.92. Data were collected from the hand with the dominant tremor for each of the PD participants.

IMUs were used to record the data, and they were placed as shown in Fig. 2. IMU 1 was affixed to the wrist support (above the ulna and radius bones) at the distal end of the forearm, IMUs 2, 3, 4 and 5 were affixed to the thumb metacarpal bone, the index and middle fingers metacarpal bones (dorsal side), and the metacarpophalangeal (MCP) joints of the thumb and the index finger, respectively. The motion of each target joint was measured differentially with the two nearby IMUs. The sensors communicated with an NXP LP1768 microcontroller through a serial peripheral interface (SPI) at a sampling rate of 100 Hz.

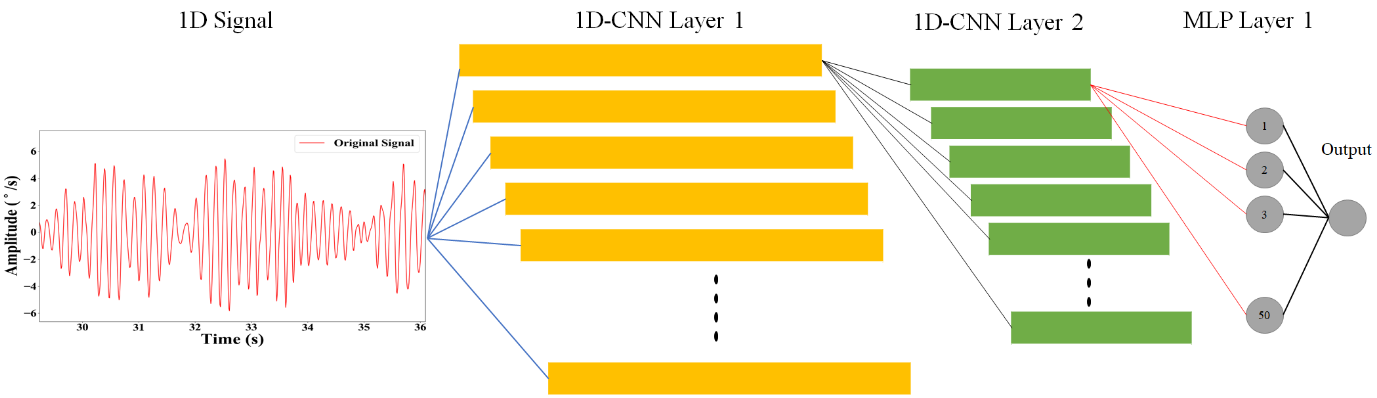

Fig. 1. A sample of the network architecture configuration with 2 CNN and 1 MLP layers.

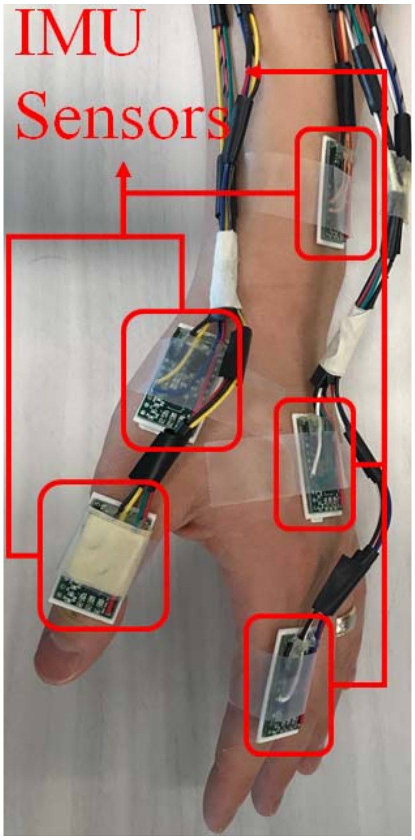

Fig. 2. Visualization of the placement of the IMU sensors on the wrist support, the thumb support, the hand support, and the (MCP) joints of the thumb and the index finger.

PD tremor can be classified into three types of tremor, resting, postural, and action tremor. Resting tremor occurs when a limb is in a resting position, postural tremor can be observed when an individual with PD tremor maintains a hand position against gravity, and action tremor is observed when an individual with PD tremor performs a voluntary movement. To assess the three types of PD tremor, the participants performed six different tasks, as follows:

1) The participant's hand was in a resting position with the palm facing down.
2) The participant's hand was in a resting position with the palm facing up.
3) The participant's hand was held in a postural position.
4) The participant was asked to extend and flex their wrist, and pinch a pencil when extending the wrist joint.
5) The participant was asked to move a pencil with the thumb and index finger when extending and flexing the wrist joint.
6) The participant was asked to draw a spiral.

The data for the first three tasks were recorded for sixty seconds. Both tasks four and five were repeated five times, and the sixth task was performed once.

B. Data Preparation

Recent studies have shown that the frequencies of parkinsonian tremor range from 3 to 17 Hz [5], [8], and voluntary motion frequencies range from 0 to 2 Hz [5], [8]. For the network to be able to learn and distinguish the difference between tremor and voluntary motion, it was trained on both tremor and voluntary motion data. To extract the tremor signal, the IMUs were aligned to the user's joint in the direction of flexion–extension, and the angular velocity data of the three axes were used to train the neural network. A 4th order Butterworth band-pass filter with cutoff frequency between 3 and 17 Hz was applied to the original motion from each axis. To extract and differentiate the ground truth voluntary motion from the action tremor, a low-pass filter with cutoff frequency of 2 Hz was applied to the original motion data of tasks four, five and six. After filtering the data, zero-phase shifting with forward–backward filtering was applied offline to remove the resulting phase shift before training the neural network models.

In order to develop a generalized model and to assess its performance on new unseen data, the data were divided into a training set, a validation set, and a testing set. The training set consisted of data from twelve PD participants, the validation dataset consisted of data from a different set of three PD participants, and the testing dataset consisted of a final set with the remaining three PD participants. The test set was only used to evaluate the performance of the neural networks after they were trained using the training and validation sets. The training process consisted of five-fold cross validation that was repeated 50 times using the data from the training and validation datasets combined to make sure that the model was not overfitted. It was then tested on new unseen data with the data from the testing set. The training, validation,

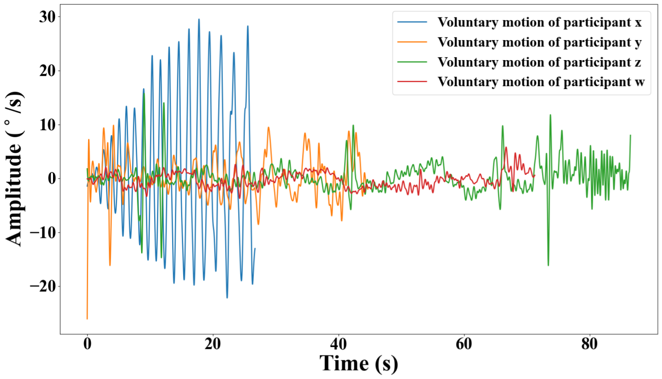

Fig. 3. Visualization of randomly selected data from different PD participants to show the difference in the data between participants.

and testing sets included all the repetitions of all the tasks of the participants' data that were in each set. Furthermore, after analyzing the recorded data from the PD participants, it was found that there is large variability in the data on which the model can be evaluated, as can be seen in Fig. 3. These data represent the unseen data from the testing dataset. The voluntary motion frequencies differ from one participant to another, and the motion is affected by the tremor that alters the patterns in each participants' data. Given this variability, the data provides a realistic basis from which to assess the applicability of the models to real-life scenarios.

C. Implementation and Accuracy Assessment

Data processing and the implementation of the five DNN architectures were performed using Python 3.7 and TensorFlow 2.0. The five networks were trained on an Intel® Core™ i7-9700 CPU @ 4.7 GHz PC, using an NVIDIA GeForce RTX 2060 to speed up the training process.

The root mean squared error (RMSE) metric was used to assess the performance of the models, and it is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}, \tag{1}$$

where $\hat{y}_i$ is the predicted value, y_i is the ground truth value, and N is the length of the input sequence.

To show the performance of the models when tracking voluntary motion, the prediction percentage accuracy (PPA) for multi-step future prediction, and the estimation percentage accuracy (EPA) were computed. Estimation is the process of approximation of past seen observations, while prediction is the process of forecasting future unseen observations. The PPA is calculated as follows:

$$\left(1 - \frac{\mid \text{True Value} - \text{Predicted Value} \mid}{\mid \text{True Value} \mid}\right) \times 100. \tag{2}$$

The EPA was calculated using the PPA equation and replacing the "Predicted Value" by "Estimated Value". The network with the lowest RMSE was then used in the simulation analysis and in the experimental analysis.

D. Statistical Analysis

A statistical analysis was conducted to compare the performance of the five networks with five different numbers of layers. The data were tested for normality, and it was found that they are nonparametric. As a result, a Kruskal-Wallis test was performed to identify differences between each predictive model. The analysis was performed using the IBM Statistical Package for the Social Sciences (SPSS version 25) statistics software.

E. Network Comparison

To find the network architecture that performed the best, the five networks were trained with tremor and voluntary motion data separately, in order for the networks to learn the difference between them. They were compared using the 5-fold cross validation repeated 50 times and with different numbers of hidden layers, as presented in Fig. 4. The RMSE accuracy measure was used to compare the models, as the problem is a regression problem, and it gives a clear indication of which one performed better. In the four RNN based architectures, 100 nodes were used in each layer. In the CNN-MLP network, each of the CNN layers consists of 64 filters, which are the learnable weights of the network, a kernel of size two, and a 1D-MaxPooling layer of size two; the MLP network consisted of one layer that consisted of 50 nodes, and the output layer.

The activation function that was used in all of the hidden layers is the Rectified Linear Unit (ReLU) [45]. The networks were trained using the *Adam* optimizer [46] and a learning rate of 0.0001 with a decay of 1e-6.

The results of the five DNNs showed that they performed very well using hand motion data, where the average RMSE of each of the five networks with different layers was below 0.5°/s. The 1D-CNN-MLP network had the lowest RMSE (RMSE $<$ 0.003°/s). One of the reasons why the 1D CNNs performed well is because they were designed with a similar concept to 2D CNNs. Vanilla CNN or 2D CNN are designed to operate exclusively on 2D data such as images and videos, and learn higher-order features from the data. To work with one-dimensional data, a modified version of the 2D CNNs has been developed. Recent studies [31] have shown that 1D CNNs are advantageous when dealing with 1D signals, such as time-series data. The forward propagation and backward propagation of the 1D CNNs require simple array operations, which result in a lower computational complexity compared to 2D CNNs and to RNN-based architectures. The first 1D CNN layer defines a set of filters to learn multiple features, which will be the input to the following 1D CNN layer. The second 1D CNN layer will also define a set of layers that will learn more features, and so on. As a result, the 1D CNN is able to automatically learn features about the data, which makes it more robust compared to the RNN-based architectures, which retain information using the memory cells, but do not perform automatic feature extraction.

The statistical analysis showed that there is a significant difference between the 1D-CNN-MLP with different layers and with the other networks with different layers. The statistical analysis showed that within each of the first four

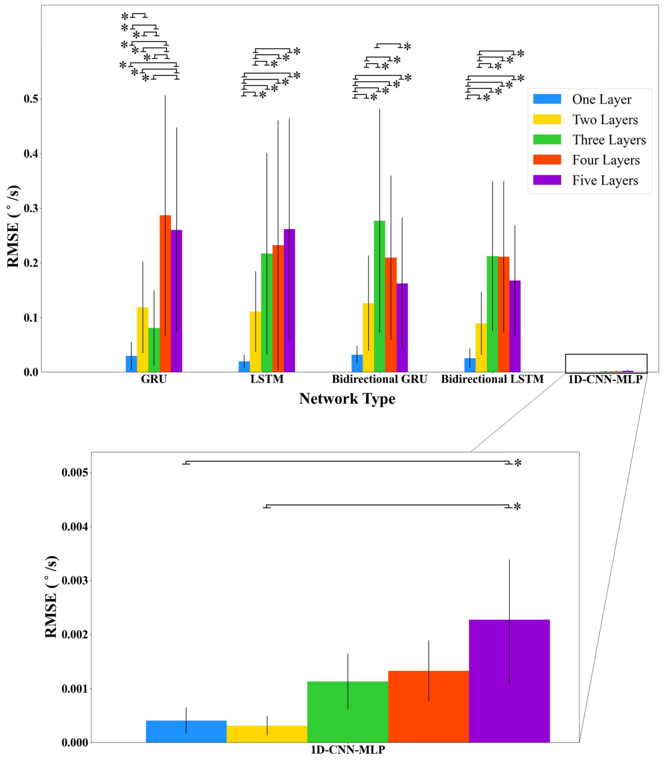

Fig. 4. Comparison of the five neural network architectures with five different layers. The black bars represent the standard deviation (±). A significant difference ($p < 0.05$) between the number of layers in each network is indicated by a "*".

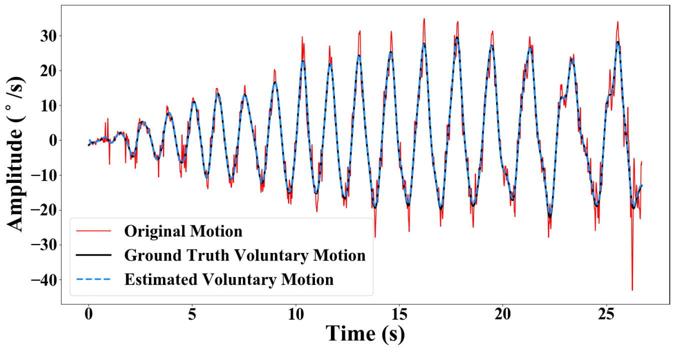

Fig. 5. Visualization of a sample output of the 1D-CNN-MLP model when the data of Task 6 for Participant 16 were used as an input. Task 6 represents the data when the PD participants drew a spiral.

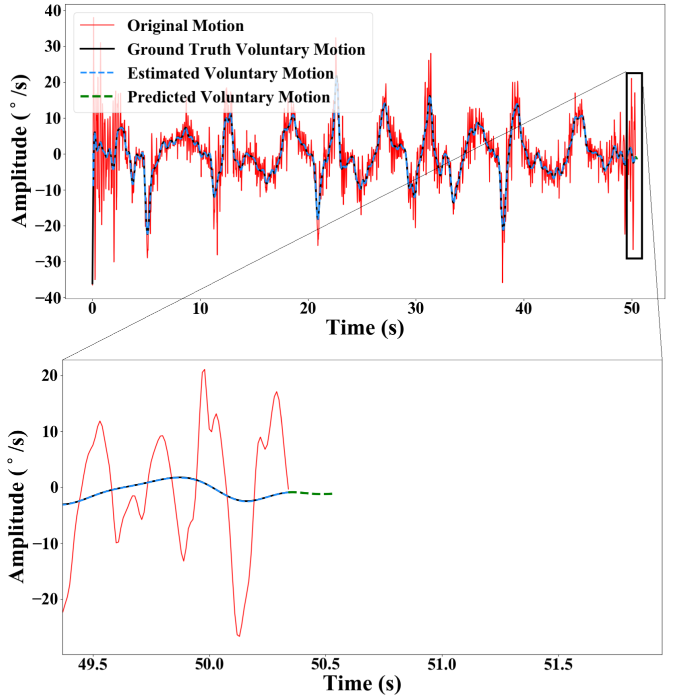

Fig. 6. Visualization of a sample output of the 1D-CNN-MLP model when the data of Task 5 for Participant 17 were used as an input. Task 5 represents the data when the PD participants were asked to move a pencil with their thumb and index finger while extending and flexing their wrist joint.

networks (GRU, LSTM, BGRU, BLSTM), there are significant differences between the different layers.

With regard to the 1D-CNN-MLP networks, the results showed that there is a significant difference between 1D-CNN-MLP with one layer and five layers, and between 1D-CNN-MLP with two layers and five layers. There were no significant differences between networks with any other combination of layer numbers. Given what has been presented, the 1D-CNN-MLP network with two 1D CNN layers was chosen arbitrarily since the 1D-CNN-MLP with one 1D CNN layer and the 1D-CNN-MLP with two 1D CNN layers had the same performance and there was no advantage of choosing one over the other. In addition to performance, and based on how the data were structured and prepared for training, there was no noticeable difference computation and training wise when choosing one over the other, where both networks took the same amount of time to train. As such, the 1D-CNN-MLP network with two 1D CNN layers and one MLP layer was used in simulation as presented in the following section.

IV. Simulation Results

In this section, the performance of the 1D-CNN-MLP model for estimation and for multi-step future prediction of voluntary motion is evaluated in different scenarios. Figs. 5, 6, and 7 show the performance of the model for different tasks that were chosen randomly from the testing dataset. Fig. 5 shows data from the original motion performed by Participant 16, the ground truth voluntary motion (black) that was extracted as discussed in Section III-B, and the estimated voluntary motion (blue). Figs. 6 and 7 show the performance of the model on new unseen participant data. They show the original voluntary motion (black), the estimated voluntary motion (blue), and the motion predicted (green) 20 steps ahead, where the prediction is highlighted in the second inset in both figures. Table I shows the performance of the model when estimating voluntary motion, and Table II shows the performance of the model for different multistep predictions of voluntary motion. The time it took the model for multistep prediction was also measured, and the results are shown in Table III.

The 1D-CNN-MLP model achieved an average EPA of 99.2% for voluntary motion when tested using the nine tasks of new data with an RMSE of 0.000313°/s. On the other hand, the 1D-CNN-MLP model achieved an average PPA of 97.0%, 94.0%, 91.7%, and 90.6% for 10, 20, 50, and 100 steps ahead of voluntary motion, respectively. The 1D-CNN-MLP model

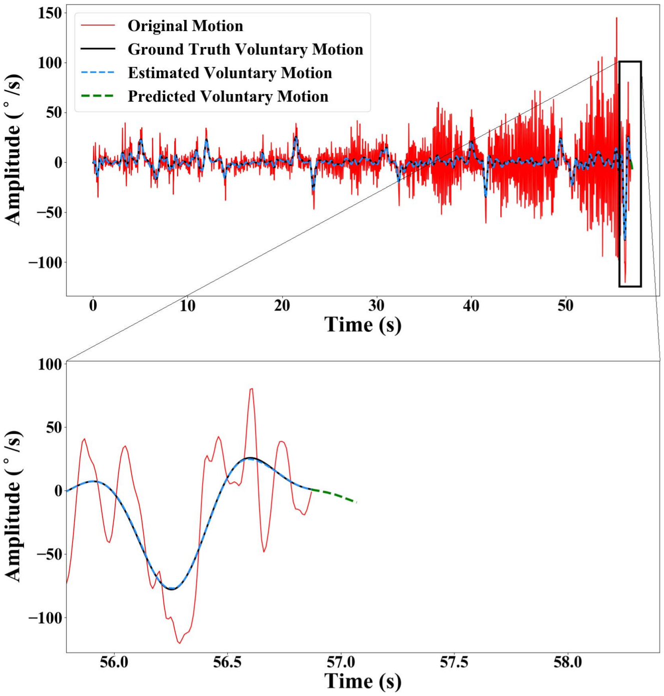

Fig. 7. Visualization of a sample output of the 1D-CNN-MLP model when the data of Task 4 for Participant 18 were used as an input. Task 4 represents the data when the PD participants were asked to extend and flex their wrist and tap their thumb and index finger together when their wrist joint is extended.

TABLE I
VOLUNTARY MOTION ESTIMATION PERFORMANCE OF THE 1D-CNN-MLP MODEL DURING TASKS 4, 5, AND 6

Subject	Voluntary Motion - EPA	SD ±
16	99.5%	0.2%
17	99.0%	0.23%
18	99.1%	0.15%
Average	99.2%	0.19%

TABLE II
VOLUNTARY MOTION PREDICTION PERFORMANCE OF THE 1D-CNN-MLP MODEL FOR DIFFERENT STEPS INTO THE FUTURE

Subject	10 steps	20 steps	50 steps	100 steps
16	97.0%	94.8%	90.2%	89.3%
17	96.2%	93.6%	92.6%	91.4%
18	98.4%	95.8%	91.7%	90.2%
Average - PPA	97.2%	94.7%	91.5%	90.3%
SD ±	0.8%	1.2%	0.8%	1%

TABLE III
MULTISTEP PREDICTION TIME FOR THE 1D-CNN-MLP MODEL

Number of steps ahead	Time (ms)
10	0.9
20	0.96
50	1
100	1.5

achieved an RMSE of 0.001, 0.0018, 0.007, and 0.04°/s for 10, 20, 50, and 100 steps ahead, respectively.

The results presented in this section show the robustness of the proposed model to track and predict voluntary motions. The results also show that the proposed model outperforms the ones proposed in the literature [8], [22]–[24], [27]–[29] for both estimation and multistep future prediction. This is important and integral to the development and enhancement of wearable tremor suppression gloves (WTSGs) and wearable assistive devices, in order to improve the effectiveness of tremor management delivered by these devices.

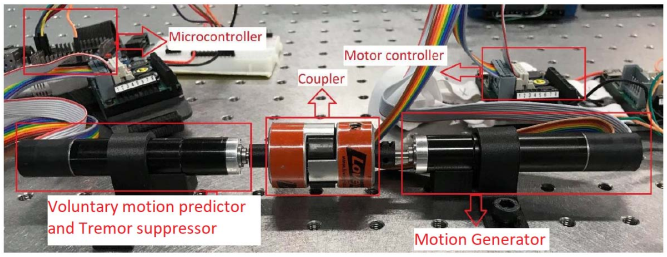

Fig. 8. Visualization of the experimental setup—showing the two motors connected by a coupler, one motor was used to simulate hand motion, and the second was controlled by either of the models to track voluntary motion and suppress tremor.

The following sections present the design of an experimental setup for motion simulation. The proposed 1D-CNN-MLP model will be compared to one of the most common estimators used in robotic devices that manage tremor—the Weighted Fourier Linear Combiner (WFLC) [23]. Both models will be used and evaluated based on their ability to help a tremor suppressor track voluntary motion while suppressing tremor.

V. MOTION REPRODUCTION: EXPERIMENTAL SETUP

This section presents the experimental setup used to compare the WFLC and the proposed 1D-CNN-MLP model. The objective of this experiment was to evaluate the performance of the two models in tracking voluntary motion while suppressing tremor. This is important since the end goal is to effectively predict voluntary motion and help in tremor suppression when integrated with WTSDs for individuals living with parkinsonian tremor.

A. Experimental Setup and Evaluation

The design of the bench-top setup is shown in Figs. 8 and 9. It consists of two brushless DC motors, each with a planetary gearhead (EC-max 16, reduction ratio 29:1, Maxon Motors, with EPOS2 24/2 controllers). The first motor (motion generator) was used as a simulator to reproduce the recorded motions of the index finger of the PD participants. The second motor (voluntary motion predictor and tremor suppressor) was controlled by the 1D-CNN-MLP model and the WFLC to compare their performances. The data from the 18 PD participants were used in the bench top experimental setup. The recorded participants' motion data were sent directly to the motion generator's motor controller using LabView Software (Version 2014, NI). The voluntary motion predictor was connected to the same PC described in Section III.C. The two motors were connected by a coupler, and an IMU sensor was placed on the right side of the coupler to read the motion reproduced by the simulator. The data were then sent to the PC to be processed. The processed data were used as the input to the WFLC and the 1D-CNN-MLP models. The output of the models—which is the estimated voluntary motion—was then used as the command motion for the motor. The performances

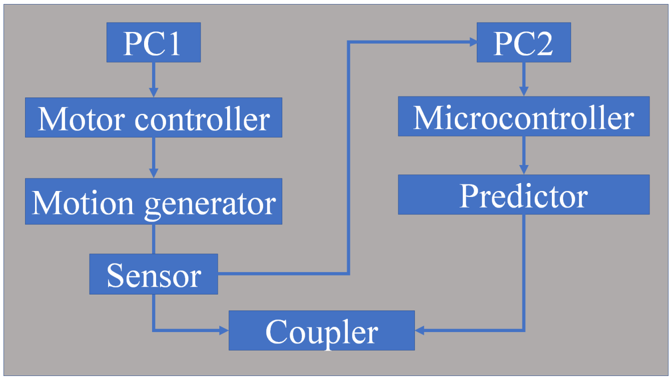

Fig. 9. A block diagram of the experimental setup that specifies the input and the output signals.

of the WFLC and the 1D-CNN-MLP models were evaluated by calculating the percentage reduction of tremor using the power spectral density (PSD) of the outputs of the WFLC and the 1D-CNN-MLP model. The PSD estimates the power distribution of the input signal over a specific frequency range. This frequency domain feature captures the overall frequency content of the specific signal and is expressed as the Fourier transform of the autocorrelation of the obtained tremor signal. The PSD of both the original signal and the output signal of the model were then calculated as discussed in [47] as follows:

$$\phi(f) = \frac{\text{FFT}(x)}{N}, \tag{3}$$

$$\text{PSD} = \sum_{i=f_1}^{f_2} \frac{\phi(i)}{T} = \frac{1}{Nt_s} \sum_{i=f_1}^{f_2} \phi(i), \tag{4}$$

where x is the signal in the time domain, N is the length of the signal, t_s is the sampling period, and f_1 and f_2 are the lower and upper limits of the range of interest. The tremor PSD reduction was then calculated by subtracting the PSD of the measured tremor motion after the tremor suppressor was activated, from the PSD of the original tremor motion, then dividing by the PSD of the original tremor motion, and then multiplying by 100 as follows:

$$\text{PSD}_{\text{reduction}} = \frac{\text{PSD}_{\text{original}} - \text{PSD}_{\text{suppressed}}}{\text{PSD}_{\text{original}}} \times 100. \tag{5}$$

The RMSE was also used to calculate how well both models performed in tracking voluntary motion.

B. Statistical Analysis

A statistical analysis was conducted to compare the performance of the 1D-CNN-MLP model to the WFLC during all six tasks. The data have been tested for normality, and the results showed that they are not normally distributed for the 1D-CNN-MLP with Tasks 1 and 3, and for the WFLC with Task 4. As such, the Kruskal-Wallis test was performed to identify differences between the two models across these tasks, and a two-by-six repeated measures ANOVA with a Bonferroni correction post hoc test was performed to identify differences between the two models and the six tasks for the normally distributed data. The α was set to 0.05. The analysis was performed using SPSS.

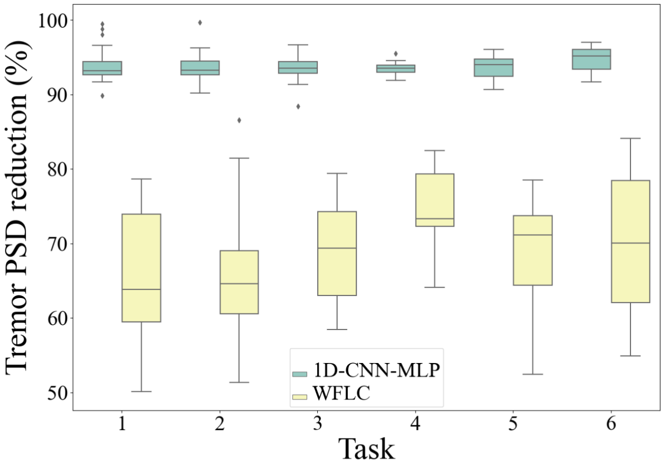

Fig. 10. Comparison between the performance of the 1D-CNN-MLP model and the WFLC in suppressing tremor. The figure shows the tremor PSD percentage reduction by the WFLC, and the tremor PSD percentage reduction by the 1D-CNN-MLP model. The black line indicates the median, the box indicates the range between the 25% and the 75% quartiles. The x axis indicates the task number, and the y axis indicates the tremor PSD reduction.

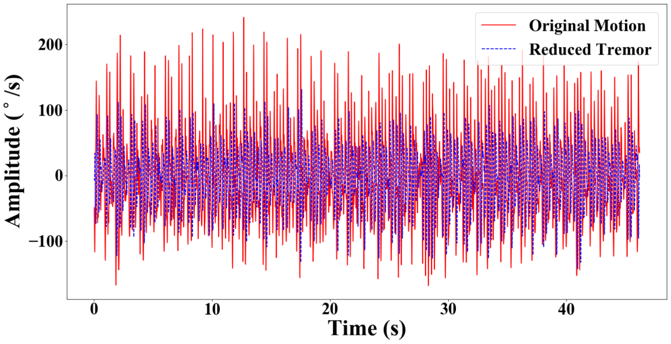

Fig. 11. Visualization of a sample output of the WFLC when data of Task 2 were used as an input. Task 2 represents the data when the hand was in a resting position with the palm facing up.

C. Results and Discussion

In this section, the performance of the WFLC and the 1D-CNN-MLP models for tracking voluntary motion and suppressing tremor is presented. The statistical analysis showed that there is a significant difference ($p < 0.05$) between the 1D-CNN-MLP model and the WFLC across the six tasks, and that there is interaction between the two main factors. Fig. 10 shows a comparison between the performance of the WFLC and 1D-CNN-MLP model in reducing tremor across the six tasks. It is important to note that the analysis showed that there was no significant difference between the six tasks.

The results show that the WFLC achieved an average of 68.8% ± 7.5% in tremor reduction, while the 1D-CNN-MLP model achieved an average of 93.8% ± 1.5% in tremor reduction, which is an improvement of 25%.

Figs. 11 and 13 show sample data with resting tremor and the output of the WFLC and 1D-CNN-MLP in suppressing tremor. Figs. 12 and 14 show the PSD of the original tremor and the suppressed tremor by the WFLC and the 1D-CNN-MLP respectively.

Fig. 15 shows a comparison of the performance of the WFLC and the 1D-CNN-MLP model when tracking voluntary

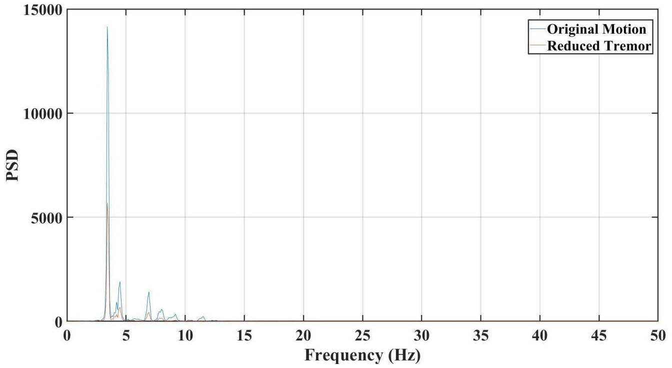

Fig. 12. Visualization of PSD of the original and suppressed tremor for the WFLC model.

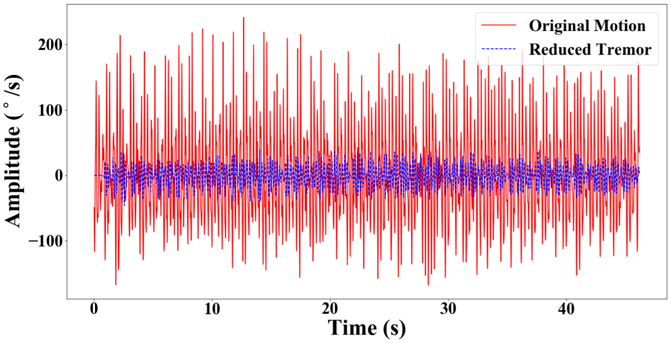

Fig. 13. Visualization of a sample output of the 1D-CNN-MLP model when data of Task 2 were used as an input. Task 2 represents the data when the hand was in a resting position with the palm facing up.

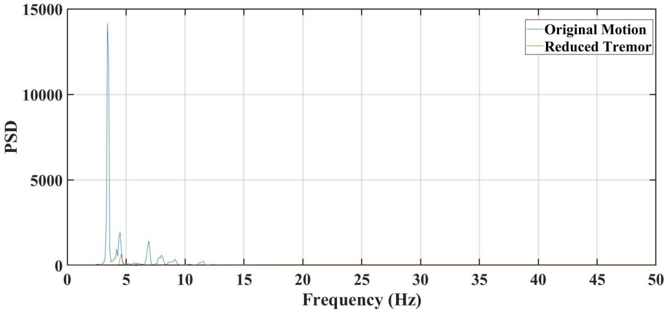

Fig. 14. Visualization of PSD of the original and suppressed tremor for the 1D-CNN-MLP model.

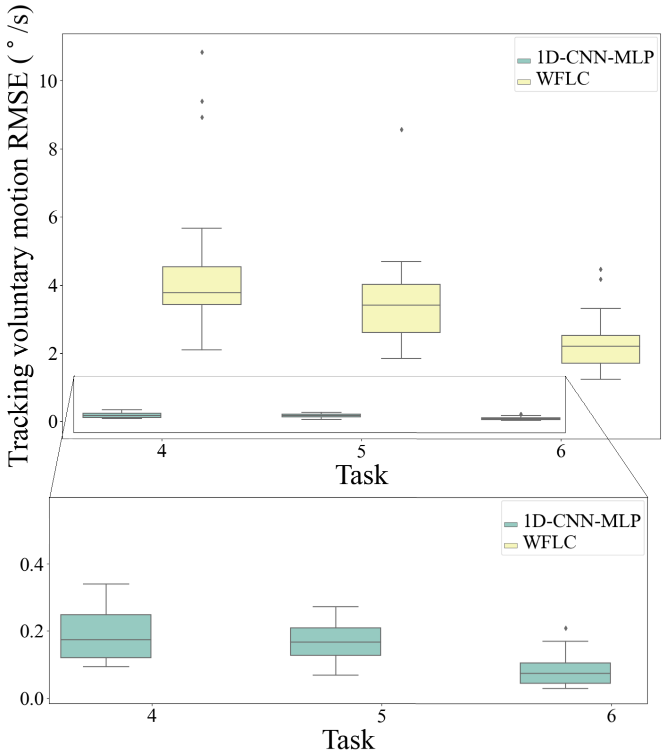

Fig. 15. Comparison between the performance of the 1D-CNN-MLP model and the WFLC when tracking voluntary motion. The figure shows the RMSE of the WFLC, and the RMSE of the 1D-CNN-MLP model. The black line indicates the median and the box indicates the range between the 25% and the 75% quartiles. The *x* axis indicates the task number, and the *y* axis indicates the tremor PSD reduction. Note the different scales in the *y* axis.

motion of Tasks 4, 5 and 6, where the analysis has also shown that there was no significant difference between these tasks.

Figs. 16 and 17 show sample data of Task 4 that contains action tremor and voluntary motion. Fig. 16 shows the performance of the WFLC in estimating voluntary motion, while Fig. 17 shows how the 1D-CNN-MLP performed in real time when predicting voluntary motion. The input to both models was fed in real time using the data from the IMU, which measures the motion reproduced by the simulator. The 1D-CNN-MLP model was configured to predict 50 ms ahead to compensate for the time delay introduced by the system (data processing and motor response times). This is important to move the motor to the predicted position to follow voluntary motion and suppress tremor in real time. When estimating voluntary motion, the RMSE achieved by the WFLC in this scenario was 3.7°/s, while the 1D-CNN-MLP model achieved an RMSE of 0.07°/s. The results of the WFLC output can be seen in Fig. 16, where it cannot accurately track voluntary motion. This is likely due to the fact that the WFLC cannot track higher tremor harmonics adaptively and separately. Furthermore, the WFLC estimates a signal, which results in a delayed response from the motors, while the proposed model can learn directly from the data to predict the future steps.

The prediction accuracies of the 1D-CNN-MLP model decreased slightly in the experimental setup compared to the simulation results presented in Section IV, where the average RMSE increased from 0.012°/s to 0.12°/s, however, it is still considered relatively small. The increase in the RMSE could be the result of different factors. One of these factors is that the use of the model in the experimental setup is closer to the real-life behaviour of motion data under study, and the simulation results represent the behaviour of the model in an ideal situation. Other factors to be considered are the data transfer or missing data during data transfer, and motor(s) response time. One main limitation of using the DNN model proposed is the time it takes to find the optimal parameters for the network, and the training time of the network.

Despite the slight increase in the RMSE when predicting voluntary motion, the results of the 1D-CNN-MLP model still showed its robustness to learn and differentiate between voluntary motion and tremor from the data directly, and to predict future voluntary motion with high accuracies. It was able to increase the tremor reduction to 93.8% on average compared to the reduction achieved by the WFLC.

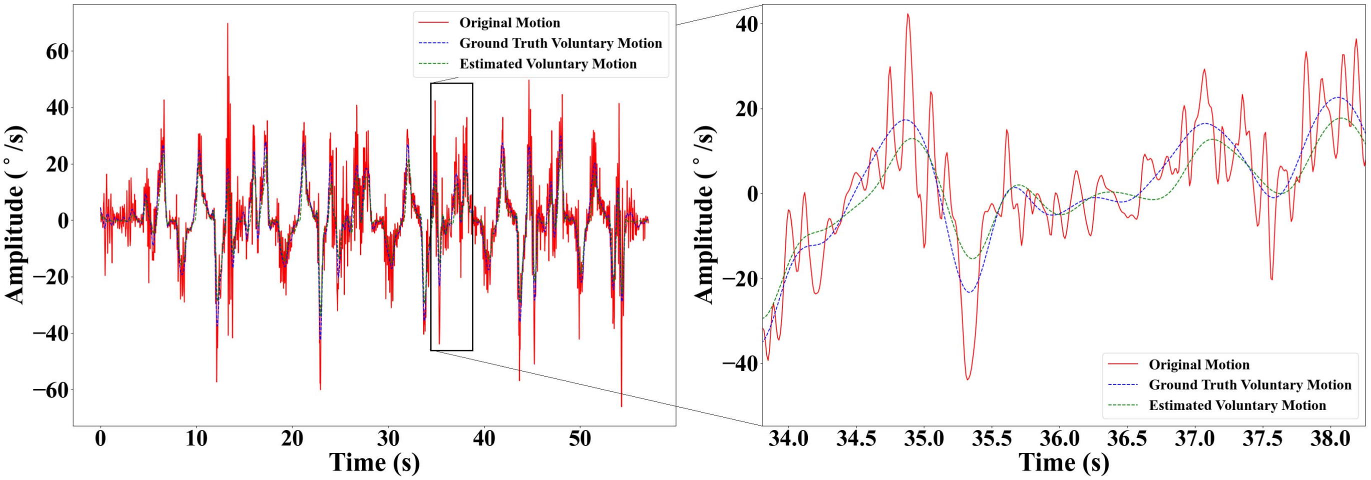

Fig. 16. Visualization of a sample output of the WFLC when data of Task 4 were used as an input. Task 4 represents the data when the participants were asked to extend and flex their wrist, and tap their thumb and index finger together when their wrist joint is extended.

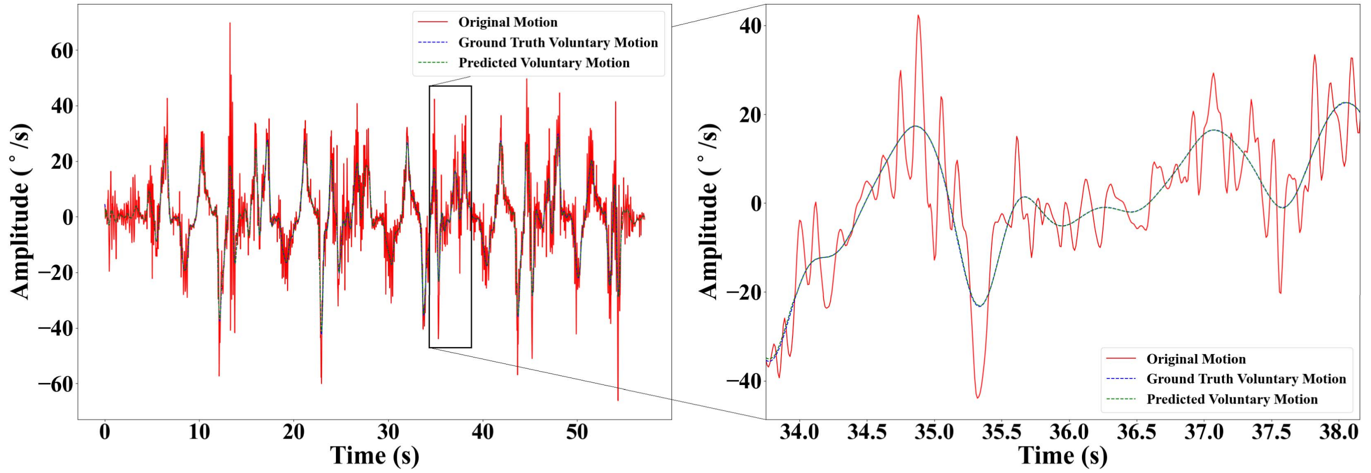

Fig. 17. Visualization of a sample output of the 1D-CNN-MLP when data of Task 4 were used as an input. Task 4 represents the data when the participants were asked to extend and flex their wrist, and tap their thumb and index finger together when their wrist joint is extended.

Lastly, although finding the optimal parameters and training the networks consumes time, once the network is trained it can be used without the need to retrain and find the parameters again. The computational time of the proposed model to predict future voluntary motion was as low as 1.5 ms for 100 steps ahead.

VI. Conclusion

This work proposes the design and implementation of a novel user and task-independent voluntary motion predictor that is based on DNN. The results of the proposed work show its potential when tracking voluntary motion, and predicting up to 100 steps ahead with high accuracy. In both simulation and an experimental assessment, the proposed 1D-CNN-MLP model showed that it overcomes the drawbacks of the estimators and predictors proposed in the literature, especially the time delay for real-time tremor management required by wearable assistive devices.

Future work will focus on investigating the factors that contributed to the increase of RMSE when predicting voluntary motion for the bench-top experimental setup, and to minimize their effect as much as possible so that the model has better performance in a real-life situation. Future work will also investigate the integration of the developed model with the WTSD being developed in the Wearable Biomechatronics Laboratory at Western University. It will also be important to investigate whether there are any advantages to using the cloud to train the network online and on the go; however, this will require further investigation into encrypting patients' data to ensure that their privacy is not compromised. An alternative is to investigate the use of a more powerful computing unit (e.g., Nvidia Jetson Nano, Raspberry Pi4, or Google Coral) to perform overnight training to increase the model's performance. However, the drawbacks of using a more powerful computing unit are the increase in the price, size, and power consumption of the device, along with the likely need for a cooling mechanism.

Acknowledgment

The authors would like to acknowledge José Guillermo Collí Alfaro who helped support this work, as well as all of the people who participated in the trials.

References

[1] C. A. Davie, "A review of Parkinson's disease," *Brit. Med. Bull.*, vol. 86, no. 1, pp. 109–127, 2008.

[2] C. Marras *et al.*, "Prevalence of Parkinson's disease across North America," *NPJ Parkinson's Disease*, vol. 4, no. 1, pp. 1–7, 2018.

[3] P. N. Today. (2020). *Parkinson's Disease Statistics*. [Online]. Available: https://parkinsonsnewstoday.com/parkinsons-disease-statistics

[4] J. Timmer, S. Häussler, M. Lauk, and C.-H. Lücking, "Pathological tremors: Deterministic chaos or nonlinear stochastic oscillators?" *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 10, no. 1, pp. 278–288, Mar. 2000.

[5] Y. Zhou, M. E. Jenkins, M. D. Naish, and A. L. Trejos, "The measurement and analysis of parkinsonian hand tremor," in *Proc. IEEE-EMBS Int. Conf. Biomed. Health Informat. (BHI)*, Las Vegas, NV, USA, Feb. 2016, pp. 414–417.

[6] G. Deuschl, P. B. Ma, and M. Brin, "Consensus statement of the movement disorder society on tremor," *Movement Disorders*, vol. 13, no. S3, pp. 2–23, 1998.

[7] E. R. de Lima, A. O. Andrade, J. L. Pons, P. Kyberd, and S. J. Nasuto, "Empirical mode decomposition: A novel technique for the study of tremor time series," *Med. Biol. Eng. Comput.*, vol. 44, no. 7, pp. 569–582, Jul. 2006.

[8] Y. Zhou, M. E. Jenkins, M. D. Naish, and A. L. Trejos, "Characterization of parkinsonian hand tremor and validation of a high-order tremor estimator," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 9, pp. 1823–1834, Sep. 2018.

[9] K. E. Lyons and R. Pahwa, "Deep brain stimulation and tremor," *Neurotherapeutics*, vol. 5, no. 2, pp. 331–338, 2008.

[10] W. J. Elias and B. B. Shah, "Tremor," *JAMA*, vol. 311, no. 9, pp. 948–954, 2014.

[11] A. As'arry, M. Z. M. Zain, M. Mailah, and M. Hussein, "Hybrid learning control for improving suppression of hand tremor," *Proc. Inst. Mech. Eng. H, J. Eng. Med.*, vol. 227, no. 11, pp. 1171–1180, 2013.

[12] B. Taheri, D. Case, and E. Richer, "Theoretical development and experimental validation of an adaptive controller for tremor suppression at musculoskeletal level," in *Proc. ASME Dyn. Syst. Control Conf.*, Palo Alto, CA, USA, Oct. 2013, Art. no. V002T22A005.

[13] Y. Zhou, M. E. Jenkins, M. D. Naish, and A. L. Trejos, "Development of a wearable tremor suppression glove," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomech. (Biorob)*, Enschede, The Netherlands, Aug. 2018, pp. 640–645.

[14] J. Á. Gallego, E. Rocon, J. M. Belda-Lois, and J. L. Pons, "A neuroprosthesis for tremor management through the control of muscle co-contraction," *J. NeuroEng. Rehabil.*, vol. 10, no. 1, p. 36, 2013.

[15] B. Taheri, "Real-time pathological tremor identification and suppression in human arm via active orthotic devices," Ph.D. dissertation, Dept. Mech. Eng., Southern Methodist Univ., Dallas, TX, USA, 2013.

[16] D. Case, B. Taheri, and E. Richer, "Design and characterization of a small-scale magnetorheological damper for tremor suppression," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 1, pp. 96–103, Feb. 2013.

[17] Y. Zhou, M. E. Jenkins, M. D. Naish, and A. L. Trejos, "Design and validation of a novel mechatronic transmission system for a wearable tremor suppression device," *Robot. Auton. Syst.*, vol. 91, pp. 38–48, May 2017.

[18] S. Kazi, A. As'arry, M. Z. M. Zain, M. Mailah, and M. Hussein, "Experimental implementation of smart glove incorporating piezoelectric actuator for hand tremor control," *WSEAS Trans. Syst. Control*, vol. 5, no. 6, pp. 443–453, 2010.

[19] K. C. Veluvolu and W. T. Ang, "Estimation of physiological tremor from accelerometers for real-time applications," *Sensors*, vol. 11, no. 3, pp. 3020–3036, 2011.

[20] B. Taheri, D. Case, and E. Richer, "Robust controller for tremor suppression at musculoskeletal level in human wrist," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 2, pp. 379–388, Mar. 2014.

[21] B. Taheri, D. Case, and E. Richer, "Adaptive suppression of severe pathological tremor by torque estimation method," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 717–727, Apr. 2015.

[22] C. A. Vaz and N. V. Thakor, "Adaptive Fourier estimation of time-varying evoked potentials," *IEEE Trans. Biomed. Eng.*, vol. 36, no. 4, pp. 448–455, Apr. 1989.

[23] C. N. Riviere, R. S. Rader, and N. V. Thakor, "Adaptive cancelling of physiological tremor for improved precision in microsurgery," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 7, pp. 839–846, Jul. 1998.

[24] S. Wang, Y. Gao, J. Zhao, and H. Cai, "Adaptive sliding bandlimited multiple Fourier linear combiner for estimation of pathological tremor," *J. Biomed. Signal Process. Control*, vol. 10, pp. 260–274, Mar. 2014.

[25] S. F. Atashzar, M. Shahbazi, O. Samotus, M. Tavakoli, M. S. Jog, and R. V. Patel, "Characterization of upper-limb pathological tremors: Application to design of an augmented haptic rehabilitation system," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 5, pp. 888–903, Aug. 2016.

[26] K. C. Veluvolu, S. Tatinati, S. M. Hong, and W. T. Ang, "Multi-step prediction of physiological tremor for robotics applications," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Osaka, Japan, Jul. 2013, pp. 5075–5078.

[27] K. C. Veluvolu, S. Tatinati, S.-M. Hong, and W. T. Ang, "Multistep prediction of physiological tremor for surgical robotics applications," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 11, pp. 3074–3082, Nov. 2013.

[28] S. Shahtalebi, S. F. Atashzar, R. V. Patel, and A. Mohammadi, "WAKE: Wavelet decomposition coupled with adaptive Kalman filtering for pathological tremor extraction," *Biomed. Signal Process. Control*, vol. 48, pp. 179–188, Feb. 2019.

[29] S. Shahtalebi, S. F. Atashzar, O. Samotus, R. V. Patel, M. S. Jog, and A. Mohammadi, "PHTNet: Characterization and deep mining of involuntary pathological hand tremor using recurrent neural network models," *Sci. Rep.*, vol. 10, no. 1, pp. 1–19, Dec. 2020.

[30] M. van Gerven and S. Bohte, "Artificial neural networks as models of neural information processing," *Frontiers Comput. Neurosci.*, vol. 11, p. 114, Dec. 2017.

[31] A. Ibrahim, Y. Zhou, M. E. Jenkins, A. L. Trejos, and M. D. Naish, "The design of a Parkinson's tremor predictor and estimator using a hybrid convolutional-multilayer perceptron neural network," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Montreal, QC, Canada, Jul. 2020, pp. 1–4.

[32] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, Bruges, Belgium, Apr. 2015, pp. 89–94.

[33] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018.

[34] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, Dec. 2018.

[35] X. Zhang, F. Shen, J. Zhao, and G. Yang, "Time series forecasting using GRU neural network with multi-lag after decomposition," in *Neural Information Processing* (Lecture Notes in Computer Science), vol. 10638, D. Liu, S. Xie, Y. Li, D. Zhao, and E. S. El-Alfy, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-70139-4_53.

[36] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Olomouc, Czech Republic, Dec. 2013, pp. 273–278.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[39] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[41] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[42] Y. LeCun, C. Cortes, and C. Burges. *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist

[43] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2017, pp. 933–941.

[44] S. Shon, A. Ali, and J. Glass, "Convolutional neural networks and language embeddings for end-to-end dialect recognition," 2018, *arXiv:1803.04567*. [Online]. Available: http://arxiv.org/abs/1803.04567

[45] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 807–814.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[47] E. Rocon, J. M. Belda-Lois, A. F. Ruiz, M. Manto, J. C. Moreno, and J. L. Pons, "Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 3, pp. 367–378, Sep. 2007.