Proceedings of the ASME 2024
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC-CIE2024
August 25-28, 2024, Washington, DC

DETC2024-146284

EXPLORING DEEP LEARNING MODELS FOR PATHOLOGICAL TREMORS PREDICTION USING EMG AND KINEMATIC MEASUREMENTS

Subham Samal[1], **Oumar Barry**[1],*

[1]Department of Mechanical Engineering,
Virginia Polytechnic Institute and State University,
Blacksburg, Virginia 24061

ABSTRACT

Pathological tremor is a common neuromuscular disorder that significantly affects the quality of life for patients worldwide. With recent developments in robotics, rehabilitation exoskeletons serve as one of the solutions to alleviate these tremors. Accurate predictive modeling of tremor signals can be used to provide alleviation from these tremors via various currently available solutions like adaptive deep brain stimulation, electrical stimulation and rehabilitation orthoses, motivating us to explore better modeling of tremors for long-term predictions and analysis. This study is a preliminary step towards the prediction of tremors using artificial neural networks using EMG signals, leveraging the 20-100 ms of Electromechanical Delay. The kinematics and EMG data of a publicly available Parkinsonian tremor dataset is first analyzed, which confirms that the underlying EMGs have similar frequency composition as the actual tremor. 2 hybrid CNN-LSTM based deep learning architectures are then proposed to predict the tremor kinematics ahead of time using EMG signals and tremor kinematics history, and the results are compared with baseline models. The motivation behind hybrid CNN-LSTM models is to exploit both the temporal and spatial dependencies using CNN and LSTM respectively. This is then further extended by adding constraints-based losses in an attempt to further improve the predictions.

1. INTRODUCTION

Pathological tremors are involuntary rhythmic movements of varying frequencies and amplitudes that can affect one or more body parts, hindering the patients from performing activities of daily living (ADLs) like writing, eating, and object manipulation. Parkinson's Disease (PT) [1] and Essential Tremors (ET) [2] are the two most common disorders that cause tremors and affect millions of people around the world.

Existing tremor alleviation solutions include adaptive deep brain stimulation, electrical stimulation and rehabilitation orthoses, all of which would benefit from accurate modeling and prediction of tremor signals. Time delay and harmonic/frequency models serve as the foundation for many of the tremor prediction algorithms currently in use, such as the Band-limited Multifrequency Fourier Linear Combiner (BMFLC)[3], Weighted Frequency Fourier Linear Combiner (WFLC) [4], and Autoregressive model (AR)[5]. Previous research has shown that these models are too general and simplistic to accurately predict tremors in the long term [6]. Earlier studies have theorized presence of highly complex and nonlinear neuromusculoskeletal dynamics behind pathological tremors, which involve motor cortex [7], feedback/reflex loops (e.g., Golgi Tendon Organs, Renshaw Cells, Spindle Organs) [8], and time delays [9]. There are a few studies that have conducted modeling of the neuromusculoskeletal (NMM) systems, particularly for tremor studies [8][10]. However, the adopted models are simplified or limited to linear models. These unsolved problems have motivated us to explore a better model for accurate long-term predictions.

Deep learning techniques have developed rapidly in recent years. In the domain of tremor signals forecasting, various previous studies have used neural networks to predict tremors using historical kinematic signals. Ibrahim et al. [11] proposed a hybrid convolutional multilayer perceptron architecture and Wang et al [6] proposed an LSTM architecture. EMG activity from skeletal muscles precedes mechanical tension by 20–100 ms [12][13]. This electromechanical delay (EMD) motivated us to use EMG data to develop a prediction model. Previous studies have leveraged this EMD to predict joint kinematics using EMG signals and neural networks [14][15][16]. In this study, we aim to extend this approach to predict pathological tremor kinematics from EMG. In this work, we propose to evaluate and compare different neural network architectures, using MLPs, LSTM, and combined

*Corresponding author: obarry@vt.edu

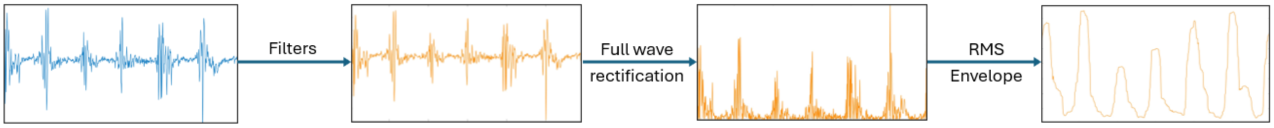

FIGURE 1: EMG signal processing steps

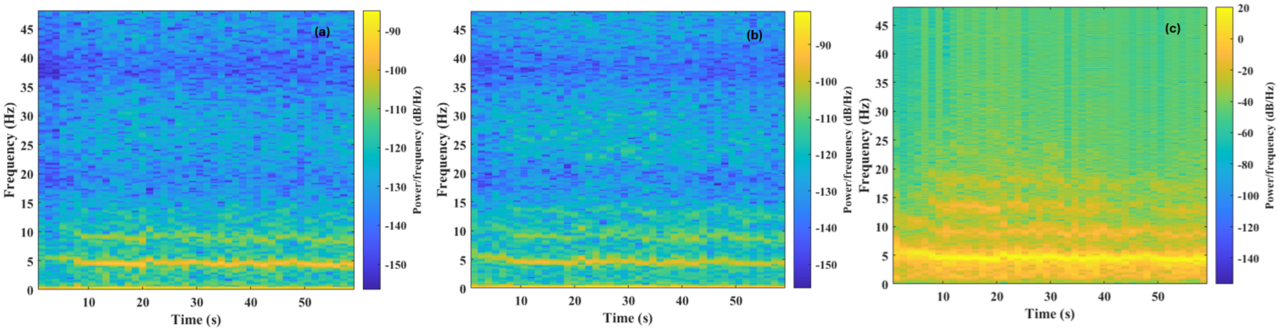

FIGURE 2: Spectrograms of Extensor EMG (a), Flexor EMG (b) and wrist flexion/deviation (c)

models to predict wrist tremors using sEMG signals. An openly available dataset is used to test the efficacy of our models. Our primary target is to be able to successfully predict about 100 ms the tremor pattern (about one-half cycle of tremor) so that we can use it later for the rehabilitation exoskeleton being developed in our lab [17][18].

The rest of the paper is arranged as follows: In Section 2, the dataset used for this study and the signal processing steps are discussed. Section 3 describes the proposed deep learning architectures, followed by Section 4, in which the results of the proposed models are discussed and comparisons are done. Section 5 explores the addition of constraint-based loss functions to improve performance. Finally, Section 6 summarizes the findings and proposes future work.

2. TREMOR DATA AND OBSERVATIONS

2.1 Datasets

For this study, a publicly available dataset was obtained from a recent study by Pinheiro et al. [19]. The study consisted of 5 volunteers, 4 of whom were diagnosed with PT, and 1 in ET. Two sEMG electrodes were placed on the ECRL and FCU muscles of the most affected arm and an Inertial Measurement Unit (IMU) (Trigno Wireless System, Delsys, Inc.) was then positioned at the hand's back, and the angular velocity signal was recorded. The acquisitions were performed at 1925.93 Hz and 148.5 Hz sampling frequencies for sEMG and IMU respectively. The volunteers were asked to extend their arms forward perpendicular to the torso, with palms down as steady as possible. About one minute of sensors' data was recorded in each trial. Since sEMG data were collected for ECRL and FCU muscles, tremors in the flexion-extension displacement angle were selected for this study, which is the IMU's measured pitch.

2.2 Preprocessing

The raw EMG signals may encompass noise attributable to ambient environmental factors, poor electrode contact, or the inherent noise from the equipment itself. This noise can adversely impact the performance of neural networks. Thus, it is necessary to mitigate this noise to the greatest extent possible and to render the signal smoother. For the proposed, tremor prediction method here, several preprocessing steps on the raw kinematic data and EMG signals were carried out basis recommendations in literature[20], and outlined in Figure 1. The EMG signals were first passed through a high pass filter to remove the DC component of the voltage, and then through a low pass filter (500 Hz) to remove unusable high-frequency noise[21]. Zero-phase 5th-order infinite impulse response (IIR) filters were used to ensure that there is no delay in the filtered signal. A notch filter with 60 Hz was also used to reduce power line noise. The filtered signals are then rectified, and the envelope is obtained by computing the root mean square (RMS) value of the signal within a sliding window of 50 ms. The envelopes are a representation of the muscle activation level[22].

For the wrist flexion-extension data, a band-pass filter (1 Hz - 20 Hz) was used to remove DC offsets and noise. Although voluntary movement components here are limited due to postural task execution, but when present, tremor can be separated from most voluntary movements through frequency-based filtering techniques [23]. All the preprocessed signals are then subsampled to 1000 Hz.

Figure 2 shows the spectrograms of the preprocessed joint angles and EMGs generated by short-time Fourier transformations. In general, we can observe that the frequencies of the dominant harmonic components have similar patterns for both EMGs and kinematics data. We also observe that the frequencies and amplitudes of the harmonic components are changing over time, indicating that pathological tremor is a nonlinear dynamics

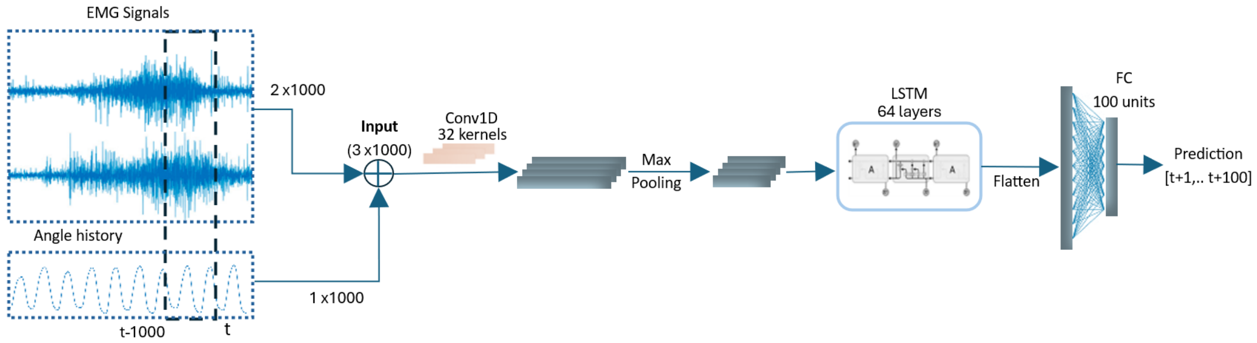

FIGURE 3: Architecture of the Sequential CNN-LSTM 1 Network as given in Table 1

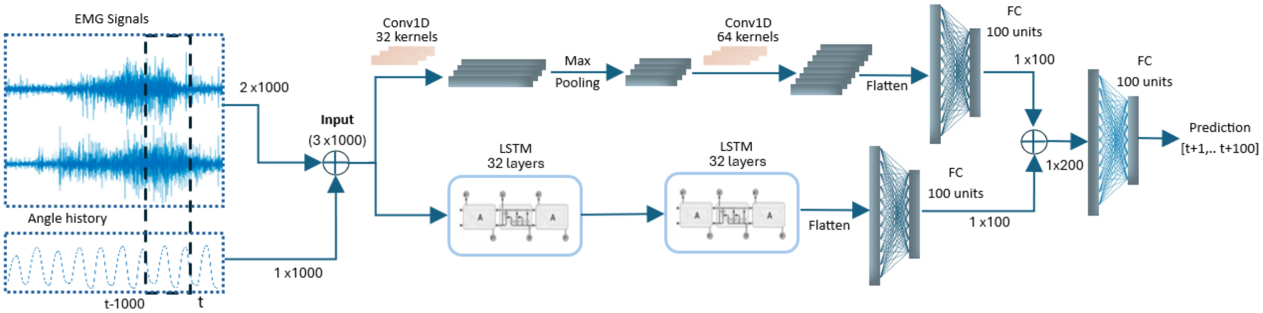

FIGURE 4: Architecture of the Parallel Y CNN-LSTM 2 Network as given in Table 1

problem, supporting findings in previous studies.[6][24][25] The preprocessed data are then normalized to a 0-1 scale with the min-max feature scaler[26]; normalization has been found to aid in improvement of the performance and training stability of deep learning models. The normalized data is then segmented into windows to be fed into the neural network models. Each window is 1.1 seconds in length, of which the first 1 second is the input sequence and the next 0.1s (100 ms) is the prediction horizon. The first 80% of the dataset is used as training data, and the rest is test data. While training, 80% of the training data was used for training, and the rest for validation.

3. TREMOR MODELING WITH NEURAL NETWORKS

This section discusses the approaches taken for modeling of pathological tremors using various neural network architectures.

In general, the musculoskeletal system can be represented as:

$$\tau = f_\eta(Z_q, Z_\tau, Z_\eta) \tag{1}$$

where τ represents the muscle force, Z_q, Z_τ, Z_η represent the delayed time series of kinematics states q, muscle load τ and neuromuscular states η[27]. f_η is a non-linear function which maps τ to Z_q, Z_τ and Z_η . Now, using forward dynamics, we can write the kinematic states as:

$$\ddot{q} = f_q(q, \dot{q}, \tau, u) \tag{2}$$

which is a general multibody model[28][29] of a forearm skeletal system, with f_q being a non-linear function and u is the external inputs (gravity, external forces). Using the above equations 1 and 2, we can infer that the current kinematic state is a function of time series of historical neuromuscular states and kinematic states, among others. In this study, using neural networks, we aim to obtain the best approximation of this unknown function for pathological tremors.

3.1 Deep Learning Architectures

Different neural network architectures, using MLPs, LSTM and CNNs were created for this study. CNN is a type of feed-forward neural network that incorporates convolutional computation and is capable of learning spatial properties. The one-dimensional convolution of 2 functions f and g for sequence data is shown as follows:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\, d\tau \tag{3}$$

LSTM is a temporal recurrent neural network, and thus is strong in extrapolating the temporal characteristics, making it suitable for processing and forecasting events with time intervals and delays. To leverage the benefits of both these structures to extract both spatial and temporal features, 2 hybrid CNN-LSTM models are also proposed. The first model is the parallel Y-CNN-LSTM network, as outlined in Figure 4. In this, the preprocessed EMG signals and the kinematics are fed into 2 parallel blocks, a Convolution block and an LSTM Block. The Convolution and LSTM block consists of one or more hidden CNN and LSTM layers;

the resultant tensors are then flattened and connected to a fully connected (FC) layer of 100 neurons. The outputs of the 2 blocks are concatenated, and connected to a FC layer of 100 units. In the Sequential CNN-LSTM network, the inputs are fed into the CNN layers, and the feature vectors obtained are then passed through LSTM layers. The resultant tensor is then flattened and connected to a FC layer. A feedforward MLP with 3 hidden layers was created as a baseline model. The architectures used for evaluation in Table 1 are summarized as:

1. MLP: 2 fully connected hidden layers with 1000, 500 units
2. CNN 1: 1 hidden CNN layer of 32 filters
3. LSTM 1: 1 hidden LSTM layer of 32 units,
4. Seq CNN-LSTM 1: 1 CNN layer of 32 filters connected to a LSTM layer of 32 units
5. Y CNN-LSTM 1: Parallel CNN and LSTM layers (32 units each)
6. CNN 2: 2 CNN layers of 32 and 64 filters
7. Seq CNN-LSTM 2: 2 CNN layer of 32 and 64 filters connected to first LSTM layer of 32 units, which is connected to 2nd LSTM layer with 32 units
8. Y CNN-LSTM 2: Parallel CNN (32 and 64 units) and LSTM (32 and 32 units) layers

The outer layer of each model is a dense layer of 100 units, to predict 100 timesteps (100 ms) of data. All the CNN layers are Conv1D layers, connected to a batch normalization layer[30] and MaxPooling layer [31] to reduce internal covariance shift and add regularization respectively. All the models were created in Tensorflow V2.15 environment.

3.2 Evaluation Criteria

The prediction output from the neural network is collected and to verify the effectiveness and quantify the performance of the proposed framework, root mean square error (RMSE) is first used. RMSE is often used as a metric in time series forecasting tasks and is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{T}\Sigma_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{4}$$

where y_t and $\hat{y}_t$ indicate the true value and the corresponding predicted value respectively. The closer the RMSE term is to 0, the better the fit.

Pearson's correlation coefficient (PC) is also employed as another metric, and can be calculated by:

$$PC = \frac{\sum_{t=1}^{T}(y_t - \bar{y}_t)(\hat{y}_t - \bar{\hat{y}}_t)}{\sqrt{\sum_{t=1}^{T}(y_t - \bar{y}_t))^2}\sqrt{\sum_{t=1}^{T}(\hat{y}_t - \bar{\hat{y}}_t)^2}} \tag{5}$$

where $\bar{y}_t$ and $\bar{\hat{y}}_t$ are the mean of the ground truth and the predicted value, respectively. The Pearson correlation measures the strength of the linear relationship between two variables, with the value varying between -1 to 1. The closer the value to 1, more the positive correlation; while closer to -1 indicates a negative correlation, and 0 being no correlation.

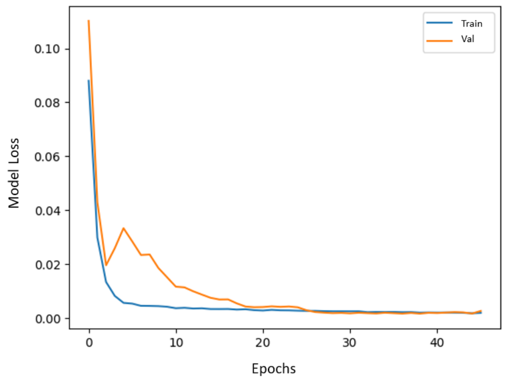

FIGURE 5: Illustration of the MSE loss over time/epochs

Finally, R-squared metric is also employed and is given as:

$$R^2 = 1 - \frac{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{T}(y_t - \bar{y}_t)^2} \tag{6}$$

R-Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.

4. RESULTS

In this section, the performance of the different frameworks are verified on wrist F/E angle predictions on the dataset. First, the training process of the proposed framework is illustrated. Overall comparisons of all the models are then performed to demonstrate the predicted results of the proposed frameworks, including representations of the predicted F/E joint angles. Effect in performance across different input sequences and prediction horizon are also studied. Finally, the effects of changing input and prediction windows, and hyperparameters on performance are investigated.

4.1 Training Process

For training, the MSE loss function was considered, which is defined as:

$$Loss_{mse} = \frac{1}{T}\Sigma_{t=1}^{T}(y_t - \hat{y}_t)^2 \tag{7}$$

Here, y_t and $\hat{y}_t$ are the truth and predictions of the normalized F/E angle. The training was set to 100 epochs with a batch size of 64. An Early Stopping functionality was also included, which stops the training if there is no decrease in the validation loss after 7 epochs. This is implemented to stop the model from overfitting unto the training data [32]. The training of the models are carried out using Tensorflow on a workstation with 12th Gen Intel(R) Core(TM) i7-12650H (16 CPUs), ~2.3GHz processor, GeForce RTX 3060 GPU, and 16G RAM. Figure 5 demonstrates the convergence of the training process, as we see both training loss and validation loss initially decrease and finally reach a minima after certain epochs.

4.2 Comparison across Network Architectures

To quantitatively evaluate the performance of the different architectures, detailed comparisons are first performed across 4

TABLE 1: Performance of different architectures across subjects for the Pinheiro tremor dataset; with the 3 best models highlighted for subjects 1, 4 and 5

Subject	Model	RMSE	R-Squared	PC	Subject	Model	RMSE	R-Squared	PC
S1	MLP	0.090	0.772	0.823	S3	MLP	0.11	0.376	0.486
	CNN 1	**0.057**	**0.907**	**0.888**		CNN 1	0.127	0.356	0.473
	LSTM 1	**0.060**	**0.918**	**0.934**		LSTM 1	0.098	0.43	0.62
	Seq CNN-LSTM 1	**0.056**	**0.911**	**0.875**		Seq CNN-LSTM 1	0.168	0.356	0.473
	Y CNN-LSTM 1	0.083	0.804	0.813		Y CNN-LSTM 1	0.107	0.554	0.570
	CNN 2	0.062	0.898	0.855		CNN 2	0.126	0.310	0.499
	Seq CNN-LSTM 2	0.080	0.804	0.813		Seq CNN-LSTM 2	0.132	0.377	0.481
	Y CNN-LSTM 2	0.079	0.828	0.832		Y CNN-LSTM 2	0.128	0.356	0.445
S4	MLP	0.077	0.932	0.964	S5	MLP	0.067	0.832	0.865
	CNN 1	0.045	0.978	0.983		CNN 1	0.036	0.942	0.941
	LSTM 1	0.042	0.978	0.989		LSTM 1	0.037	0.944	0.951
	Seq CNN-LSTM 1	**0.035**	**0.986**	**0.994**		Seq CNN-LSTM 1	**0.029**	**0.959**	**0.962**
	Y CNN-LSTM 1	0.042	0.980	0.985		Y CNN-LSTM 1	0.044	0.881	0.931
	CNN 2	0.044	0.975	0.983		CNN 2	0.052	0.880	0.891
	Seq CNN-LSTM 2	**0.042**	**0.980**	**0.992**		Seq CNN-LSTM 2	**0.007**	**0.997**	**0.988**
	Y CNN-LSTM 2	**0.038**	**0.983**	**0.989**		Y CNN-LSTM 2	**0.011**	**0.995**	**0.980**

Parkinsons subjects' datasets, as summarized in Tables 1. The metrics presented are evaluated on the test data after training was completed. Figure 6 shows the predictions using the proposed hybrid CNN-LSTM models. As expected, all the deep learning models performed significantly better than the baseline MLP architecture, which corroborates the fact that deep learning-based methods can automatically extract high-level features. Amongst the deep learning architectures, Sequential CNN-LSTM models performed better for most cases (except subject 3, for which none of the models performed well), with the RMSE well below 0.05 in most cases. Since our data was normalized to a 0 to 1 scale, this effectively means less than 5% error. The Y CNN-LSTM, in overall performed almost similarly or slightly better than the standard LSTM model, despite having more parameters to train. It can also be observed that increasing the depth of the networks did not necessarily improve the performances. With the trained Sequential CNN-LSTM 2 model, the prediction for 100 ms of data took only about 0.3 ms on the same workstation, and thus could be used for real-time operations. The sequential CNN-LSTM 1 model was used to study the effects of input and prediction horizon, and hyperparameters on performance in sections 4.3 and 4.4.

4.3 Effects of Input and Prediction horizon

We also evaluate the proposed model by varying the length of the input and prediction horizon, as detailed in Table 2. It was observed that the performance of the model remained quite good up to 100 ms of prediction horizon (<5% error). However, upon further increase, the performance reduced drastically, and even increasing the input sequence length did not improve the performance.

TABLE 2: Comparison with different Input and Prediction Horizon

Input	Prediction	RMSE	R-Squared	PC
1000	40	0.025	0.985	0.941
1000	80	0.032	0.948	0.934
1000	100	0.040	0.952	0.944
1000	150	0.072	0.803	0.867
1500	150	0,072	0.809	0.881
1000	200	0.077	0.782	0.864
1000	400	0.096	0.664	0.824
2000	400	0.122	0.510	0.729

TABLE 3: Comparison with different Activation Functions

Activation Function	RMSE	R-Squared	PC
Tanh	0.040	0.952	0.944
ReLu	0.041	0.946	0.944
Leaky-ReLu	0.038	0.957	0.932
Sigmoid	0.084	0.868	0.922

4.4 Effects of Hyperparameters

In this section, we investigate the effects of hyperparameters on performance, i.e. learning rate and types of activation functions on the proposed Sequential CNN-LSTM 1 framework. The detailed results are shown in Table 4 and Table 3. The metrics shown are the average values obtained over S1, S4, and S5. Specifically, 3 learning rates are considered, i.e., 0.01, 0.001, and 0.0001. We can observe from table 4 that better performance was achieved with smaller learning rates (0.001 or 0.0001). 4 activation functions were tested, tanh, ReLU, Leaky ReLU, and Sigmoid, all of which are non-linear activation functions. Ob-

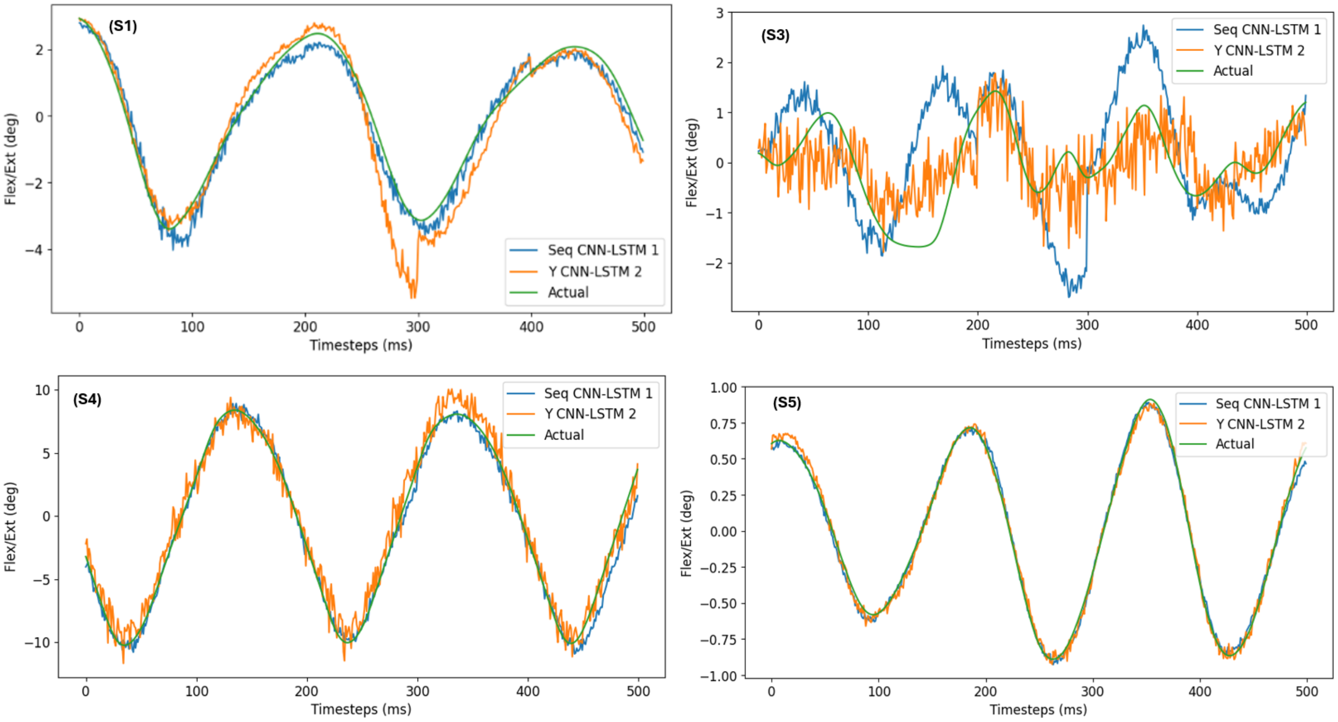

FIGURE 6: Prediction results for the Seq CNN-LSTM 1 and Y CNN-LSTM 2 models across the 4 subjects

TABLE 4: Comparison with different Learning Rate

Learning Rate	RMSE	R-Squared	PC
0.01	0.110	0.23	0.3876
0.001	0.042	0.942	0.914
0.0001	0.040	0.952	0.944

served from Table 3, it can be seen that tanh, ReLU, and Leaky ReLU activation functions achieved similar performance and better than the sigmoid activation function.

5. ADDING CONSTRAINS TO NEURAL NETWORK MODELS

As observed from Figure 6, while the predictions are close to the actual values most of the time, the predictions are still quite noisy. While methods like exponential moving averages or Kalman Filters can be implemented in real-time, we try to explore if we can modify our model in such a way that it can learn to predict smoother outputs. Inspired by Physics-informed neural networks, where the loss functions are modified to accommodate physics-based constraints and losses, we devise loss functions based on the first-order and second-order derivatives of the actual F/E angular data. The rationale being that the 1st and 2nd derivatives give info on the continuity and smoothness, and adding these losses would help to embed the same during training.

The proposed constraint loss functions can be given as:

$$Loss_1 = \frac{1}{T}\Sigma_{t=1}^{T}((y_{t+1} - y_t) - (\hat{y}_{t+1} - \hat{y}_t))^2 \tag{8}$$

TABLE 5: Comparison with adding constraint loss terms

Loss	γ_1	γ_2	RMSE	R^2	PC
MSE	0	0	0.040	0.952	0.944
MSE + $Loss_1$	0.01	0	0.041	0.929	0.945
	0.1	0	0.040	0.940	0.967
	1	0	0.046	0.920	0.946
Total Loss	0.1	0.01	0.042	0.925	0.963
	0.1	0.1	0.038	0.936	0.947

$$Loss_2 = \frac{1}{T}\Sigma_{t=1}^{T}((y_{t+2} + y_t - 2y_{t+1}) - (\hat{y}_{t+2} + \hat{y}_t - 2\hat{y}_t))^2 \tag{9}$$

The total combined loss function is then given as:

$$\text{Total Loss} = Loss_{mse} + \gamma_1 Loss_1 + \gamma_2 Loss_2 \tag{10}$$

From Table 5, it can be observed that although some in terms of metrics, there is very slight improvement, especially in terms of correlation coefficient (PC). However, increasing γ_1 too high affects the performance negatively. Further, from 7, it can be observed that when the model is trained with Total Loss, the predictions are much smoother. In future studies, we aim to add more actual physics-based loss terms based on musculoskeletal dynamics. This will involve using empirical models of muscle activation and contraction dynamics, muscle-tendon models and joint kinematic models[33].

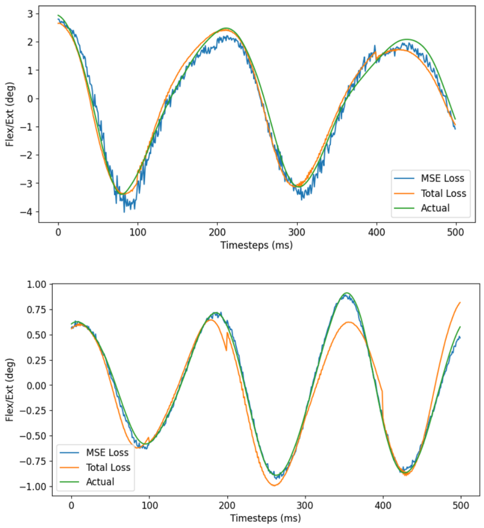

FIGURE 7: Prediction results for Subject 1 (top) and Subject 5 (bottom); after Seq CNN-LSTM 1 is trained using MSE Loss and Total Loss

6. CONCLUSION AND FUTURE WORK

This study presented a preliminary investigation towards deep learning based long-term pathological tremors prediction using EMG signals and historical motion data. An open-source dataset of sEMG and motion signals was collected and preprocessed. Various deep learning architectures were proposed, including hybrid CNN-LSTM models to predict pathological tremor signals from EMG data with fairly good results. However, the study is limited in many aspects. First, only one direction (F/E) of the tremors are studied, but in practical cases, all the 3 wrist movements are affected. Second, while neural networks perform appreciable predictions, the physical meanings of the models can't really be interpreted, giving us no information on the tremor dynamics. Finally, the application of the methods in real-time needs to be experimented.

Currently, we are performing experiments to gather a comprehensive collection of measurement data from studies, such as kinematics, EEG, and EMG, and use this data to build more precise models of pathological tremors. The collected datasets would consist of multi-directional movement measurements. For tremor modeling and prediction, we will explore Physics-Constrained Data-driven Modeling, which would employ a combination of model-based (MBR) and model-free (MFR) regressions to obtain neuromusculoskeletal models that can be used for both analytical studies and real-time applications.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation (NSF) Grant CBET #2306984: A Neuromechanical Robot Approach to Control Pathological Tremor in Upper Limbs

REFERENCES

[1] Kalia, Lorraine V and Lang, Anthony E. "Parkinson's disease." *The Lancet* Vol. 386 No. 9996 (2015): pp. 896–912.

[2] Louis, Elan D. "Essential tremor." *The Lancet Neurology* Vol. 4 No. 2 (2005): pp. 100–110.

[3] Veluvolu, Kalyana C and Ang, Wei Tech. "Estimation of physiological tremor from accelerometers for real-time applications." *Sensors* Vol. 11 No. 3 (2011): pp. 3020–3036.

[4] Riviere, Cameron N, Reich, Stephen G and Thakor, Nitish V. "Adaptive Fourier modeling for quantification of tremor." *Journal of neuroscience methods* Vol. 74 No. 1 (1997): pp. 77–87.

[5] Tatinati, Sivanagaraja, Veluvolu, Kalyana C, Hong, Sun-Mog, Latt, Win Tun and Ang, Wei Tech. "Physiological tremor estimation with autoregressive (AR) model and Kalman filter for robotics applications." *IEEE Sensors Journal* Vol. 13 No. 12 (2013): pp. 4977–4985.

[6] Wang, Jiamin and Barry, Oumar R. "Exploring data-driven modeling and analysis of nonlinear pathological tremors." *Mechanical Systems and Signal Processing* Vol. 156 (2021): p. 107659.

[7] Lindenbach, David and Bishop, Christopher. "Critical involvement of the motor cortex in the pathophysiology and treatment of Parkinson's disease." *Neuroscience & Biobehavioral Reviews* Vol. 37 No. 10 (2013): pp. 2737–2750.

[8] Zhang, Dingguo, Poignet, Philippe, Bo, Antonio PL and Ang, Wei Tech. "Exploring peripheral mechanism of tremor on neuromusculoskeletal model: A general simulation study." *IEEE transactions on biomedical engineering* Vol. 56 No. 10 (2009): pp. 2359–2369.

[9] Hajdu, David, Milton, John and Insperger, Tamas. "Extension of stability radius to neuromechanical systems with structured real perturbations." *IEEE transactions on neural systems and rehabilitation engineering* Vol. 24 No. 11 (2016): pp. 1235–1242.

[10] Lakie, Martin, Vernooij, Carlijn A, Osborne, Timothy M and Reynolds, Raymond F. "The resonant component of human physiological hand tremor is altered by slow voluntary movements." *The Journal of physiology* Vol. 590 No. 10 (2012): pp. 2471–2483.

[11] Ibrahim, Anas, Zhou, Yue, Jenkins, Mary E, Trejos, Ana Luisa and Naish, Michael D. "The design of a parkinson's tremor predictor and estimator using a hybrid convolutional-multilayer perceptron neural network." *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*: pp. 5996–6000. 2020. IEEE.

[12] Widjaja, Ferdinan, Shee, Cheng Yap, Au, Wing Lok, Poignet, Philippe and Ang, Wei Tech. "Using electromechanical delay for real-time anti-phase tremor attenuation system using functional electrical stimulation." *2011 IEEE International Conference on Robotics and Automation*: pp. 3694–3699. 2011. IEEE.

[13] Corcos, Daniel M, Gottlieb, Gerald L, Latash, Mark L, Almeida, Gil L and Agarwal, Gyan C. "Electromechanical delay: An experimental artifact." *Journal of Electromyography and Kinesiology* Vol. 2 No. 2 (1992): pp. 59–68.

[14] Yi, Chunzhi, Jiang, Feng, Zhang, Shengping, Guo, Hao, Yang, Chifu, Ding, Zhen, Wei, Baichun, Lan, Xiangyuan

and Zhou, Huiyu. "Continuous prediction of lower-limb kinematics from multi-modal biomedical signals." *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 32 No. 5 (2021): pp. 2592–2602.

[15] Krausz, Nili E, Lamotte, Denys, Batzianoulis, Iason, Hargrove, Levi J, Micera, Silvestro and Billard, Aude. "Intent prediction based on biomechanical coordination of EMG and vision-filtered gaze for end-point control of an arm prosthesis." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 28 No. 6 (2020): pp. 1471–1480.

[16] Feleke, Aberham Genetu, Bi, Luzheng and Fei, Weijie. "EMG-based 3D hand motor intention prediction for information transfer from human to robot." *Sensors* Vol. 21 No. 4 (2021): p. 1316.

[17] Wang, Jiamin. "Design and Control of an Ergonomic Wearable Full-Wrist Exoskeleton for Pathological Tremor Alleviation." Ph.D. Thesis, Virginia Tech. 2023.

[18] Samal, Subham and Barry, Oumar. "Model Predictive Control for Tremor Suppressing Exoskeleton." *IFAC-PapersOnLine* Vol. 56 No. 3 (2023): pp. 337–342.

[19] Pinheiro, Wellington C, Ferraz, Henrique B, Castro, Maria Claudia F and Menegaldo, Luciano L. "An OpenSim-based closed-loop biomechanical wrist model for subject-specific pathological tremor simulation." *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society* .

[20] Manal, Kurt and Buchanan, Thomas S. "A one-parameter neural activation to muscle activation model: estimating isometric joint moments from electromyograms." *Journal of biomechanics* Vol. 36 No. 8 (2003): pp. 1197–1202.

[21] Kieliba, Paulina, Tropea, Peppino, Pirondini, Elvira, Coscia, Martina, Micera, Silvestro and Artoni, Fiorenzo. "How are muscle synergies affected by electromyography preprocessing?" *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 26 No. 4 (2018): pp. 882–893.

[22] D'Alessio, T and Conforto, S. "Extraction of the envelope from surface EMG signals." *IEEE Engineering in Medicine and Biology Magazine* Vol. 20 No. 6 (2001): pp. 55–61.

[23] Vaz, Christopher A and Thakor, Nitish V. "Adaptive Fourier estimation of time-varying evoked potentials." *IEEE Transactions on Biomedical Engineering* Vol. 36 No. 4 (1989): pp. 448–455.

[24] Timmer, Jens, Häußler, Siegfried, Lauk, Michael and Lücking, C-H. "Pathological tremors: Deterministic chaos or nonlinear stochastic oscillators?" *Chaos: An Interdisciplinary Journal of Nonlinear Science* Vol. 10 No. 1 (2000): pp. 278–288.

[25] Darbin, Olivier, Adams, Elizabeth and Dees, Daniel. "Nonlinear dynamics in parkinsonism." *Frontiers in Neurology* Vol. 4 (2013): p. 61229.

[26] Nawi, Nazri Mohd, Atomi, Walid Hasen and Rehman, Mohammad Zubair. "The effect of data pre-processing on optimized training of artificial neural networks." *Procedia Technology* Vol. 11 (2013): pp. 32–39.

[27] Wang, Jiamin, Gupta, Sunit K and Barry, Oumar. "Towards data-driven modeling of pathological tremors." *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 83914: p. V002T02A030. 2020. American Society of Mechanical Engineers.

[28] Guo, Jianqiao, Chen, Junpeng, Wang, Jing, Ren, Gexue, Tian, Qiang and Guo, Chuanbin. "EMG-assisted forward dynamics simulation of subject-specific mandible musculoskeletal system." *Journal of Biomechanics* Vol. 139 (2022): p. 111143.

[29] Wang, Xinyue, Guo, Jianqiao and Tian, Qiang. "A forward-inverse dynamics modeling framework for human musculoskeletal multibody system." *Acta Mechanica Sinica* Vol. 38 No. 11 (2022): p. 522140.

[30] Ioffe, Sergey and Szegedy, Christian. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International conference on machine learning*: pp. 448–456. 2015. pmlr.

[31] Gholamalinezhad, Hossein and Khosravi, Hossein. "Pooling methods in deep neural networks, a review." *arXiv preprint arXiv:2009.07485* (2020).

[32] Prechelt, Lutz. "Early stopping-but when?" *Neural Networks: Tricks of the trade*. Springer (2002): pp. 55–69.

[33] Zhao, Yihui, Zhang, Zhiqiang, Li, Zhenhong, Yang, Zhixin, Dehghani-Sanij, Abbas A and Xie, Shengquan. "An EMG-driven musculoskeletal model for estimating continuous wrist motion." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 28 No. 12 (2020): pp. 3113–3120.