A Neural Network-based Design of an on-off Adaptive Control for Deep Brain Stimulation in Movement Disorders

Pitamber Shukla[1], Ishita Basu[1], Daniel Graupe[1], Daniela Tuninetti[1] and Konstantin V. Slavin[2]

Abstract— The current Food and Drug Administration approved system for the treatment of tremor disorders through Deep Brain Stimulation (DBS) of the area of the brain that controls movement, operates open-loop. It does not automatically adapt to the instantaneous patient's needs or to the progression of the disease. This paper demonstrates an adaptive closed-loop controlled DBS that, after switching off stimulation, tracks few physiological signals to predict the reappearance of tremor before the patient experiences discomfort, at which point it instructs the DBS controller to switch on stimulation again. The core of the proposed approach is a Neural Network (NN) which effectively extracts tremor predictive information from non-invasively recorded surface-electromyogram(sEMG) and accelerometer signals measured at the symptomatic extremities. A simple feed-forward back-propagation NN architecture is shown to successfully predict tremor in 31 out of 33 trials in two Parkinson's Disease patients with an overall accuracy of 75.8% and sensitivity of 92.3%. This work therefore shows that closed-loop DBS control is feasible in the near future and that it can be achieved without modifications of the electrodes implanted in the brain, i.e., is backward compatible with approved DBS systems.

Index Terms— Surface EMG, closed-loop deep brain stimulation, feed-forward back-propagation neural network, Levenberg-Marquardt learning algorithm, tremor onset prediction, movement disorders.

I. INTRODUCTION

Parkinson's Disease (PD) is a progressive chronic neurological disorder of the central nervous system for which no cure is available at present [1]. Early stage PD's most common symptoms are tremor, rigidity, imbalance, and slowness in movement, which can be treated with drug therapy. In advanced stage PD, drugs become ineffective or start producing side-effects. Surgical procedures, such as Deep Brain Stimulation (DBS), relieve patients from most of their debilitating motor symptoms. Currently approved DBS (from Medtronic) uses battery-operated surgically-implanted electrodes to deliver high frequency electrical stimulation to the neurons in the brain that control movement. DBS is particularly effective at suppressing involuntary rhythmic tremor in the $4 - 12$ Hz frequency band [2].

Open-loop DBS. Medtronic's DBS system operates *open-loop* [1], that is, the physician sets the DBS stimulation parameters, such as frequency, pulse amplitude, and pulse duration, with visual feedback from the patient. DBS stimulation is provided continuously and its parameters remain constant until the physician changes them. This implies that existing DBS technology is neither adaptive to the patient's needs nor able to follow the patient's disease progression over time.

Closed-loop DBS. To design a closed-loop DBS system we need to adaptively control the DBS stimulation parameters as well as the time intervals when DBS stimulation is administered. This requires the identification of suitable physiological signals that contain information related to the real-time response of the brain to DBS stimulation. In particular we are interested in a simple (and therefore robust) on-off control of DBS such as when stimulation is switched off and the patient does not experience tremor, the system tracks few physiological signals and predicts when tremor is about to reappear. When tremor reoccurrence is predicted, stimulation is switched on for a fixed amount of time. A subset of the authors showed in [3] that it is possible to switch off stimulation for often more than 50% of the time, which gives a considerable saving in battery life and reduces the amount of current injected in the brain.

Past Work. The neuronal brain activity measured at the site of the DBS implant would offer the necessary predictive information about tremor reappearance. However, measuring from the stimulating electrodes would require changes in the currently approved DBS system whose testing and approval could take years and considerably delay closed-loop DBS commercialization. To overcome this limitation the authors of [3], [4] proposed a DBS closed-loop design where the controller inputs are parameters estimated from non-invasively recorded sEMG signals that correlate with the tremor symptoms. In [4], a reliable predictor of tremor onset for Essential Tremor was obtained by using entropy measures, in particular approximate and wavelet entropy. Although the predictor in [4] successfully predicts tremor in all trials, the results were limited to 8 trials for one patient; moreover, the algorithm had manual training and setting of several parameter thresholds. Since this is unfeasible when it comes to massive commercialization of closed-loop DBS, and since parameter thresholds can change over time, in this work we propose a novel tremor predictor based on a self adaptive artificial neural network (NN).

Main Contribution. NN-based predictors, with physiological signals as inputs, have been widely used in clinical applications such as for predicting, for example, epileptic seizures [6] and sleep apnea [7]. In this work we propose to use a feed-forward back-propagation NN to predict tremor reappearance in PD patients. We show that the proposed NN successfully predicts tremor in 31 out of 33 trials with an

[1]The starting trials for closed-loop brain implants were announced by Medtronics in September 2009.

overall accuracy of 75.8% and sensitivity of 92.3%. This shows the possibility of NN-based closed-loop DBS system design without making any modification to the existing electrodes implanted in the brain.

Paper Organization. The paper is organized as follows. Section II describes the recording procedure of the physiological signals, the methodology used for parameter extraction and the chosen NN architecture. Section III presents the results of testing the proposed NN-based tremor predictor on actual PD data. Conclusions and directions for future work are highlighted in Section IV.

II. METHODOLOGY

A. Data Set

Two PD patients were recruited for the study. Informed consent approved by the IRB was obtained from both patients. Both patients had bilateral DBS electrodes implanted in the subthalamic nuclei. Both patients had dominant tremor in one or both arms and their symptoms were well controlled by the combination of DBS stimulation and medications. Both patients had one recording session each in the Neural Control of Movement Laboratory (NCML) at UIC. On the testing day, the patients were on their usual medication and a series of sEMG recordings were done from the the extensor digitorum communis (upper forearm). The recording setup was as in [8]. The sEMG signal was amplified (gain set to 1,000) and bandpass filtered between 20Hz and 450Hz (Delsys Inc., Boston, MA). Along with sEMG, acceleration data was recorded from the finger tip (with a calibrated Coulbourn type V 94-41 miniature solid-state piezoresistive accelerometer with resolution was 0.01g). Both sEMG and accelerometer data were sampled at f_s =1000Hz.

In the beginning of the experiment, the patient was comfortably seated in an upright position. The arm of the chair served as the supportive surface for the patient's forearm. DBS stimulation was switched off for some time before recording started. Then a trial started with the stimulation on for 20s to 50s followed immediately by an interval with stimulation off. The total duration of each trial, T, was between 50s to 100s. In each trial, the patient was in one of the following three states: R) at Rest with his/her hand completely relaxed and hanging from the chair's arm rest; P) holding a Posture, with his/her testing wrist and hand in a neutral extended position; and A) performing a voluntary Action such as reaching for his/her opposite shoulder or flexion of the wrist. In states P and A, the movement/posture was initiated either before or after switching the stimulation off. There were 15 to 20 such trials recorded for each patient. After stimulation was switched off at time t_{off}, the first instant t_{tr} when tremor visibly reappeared was noted. Reappearance of tremor was also verified by thresholding the acceleration data at 0.2mm/s^2 for states R and P.

B. Parameter Extraction

The parameters extracted from the sEMG and acceleration signals and used as inputs to the NN are:

Spectral Measures. We use two spectral measures calculated for the acceleration and sEMG signals. The sEMG is first smoothed by calculating its power over windows of 50ms (=50 samples) duration that slid over every sample.

1) Let $X_n(t)$ be the smoothed sEMG signal in the n-th frequency band, $n \in \{1, ..., N\}$ at time t, $t \in \{1, ..., L\}$, obtained by a Daubechies4 wavelet decomposition. Here $N = 10$, $L = Tf_s$ where f_s=1KHz and T are the sampling rate and the total duration of the signal, respectively. The *mean power* in the n-th frequency band, defined as

$$\overline{P}_n := \frac{1}{L} \sum_{t=1}^{L} |X_n(t)|^2, \tag{1}$$

captures the average signal content in the n-th frequency band over an interval of duration T.

2) Let P_i be the power of the smoothed sEMG signal or of the acceleration signal in the frequency band centered around f_i, $i \in \{1, ..., M\}$, $M = 512$, obtained by using a Fourier transform. Then we define:

$$P_{\max} = \frac{\max_{j\in\{4,...,19\}} P_j}{\sum_{j\in\{20,...,41\}} P_j}; \; f_{\max} = \arg \max_{j\in\{4,...,19\}} P_j. \tag{2}$$

where $f_4 = 3\text{Hz}$, $f_{19} = 18\text{Hz}$, $f_{41} = 40\text{Hz}$. The meaning of these quantities is as follows. The smoothed sEMG signal has most of its power concentrated in the 0-40Hz range. We omit the 0-3Hz band to account for the DC value and very low frequency movement artifacts. $f_{\max}$ is the frequency in the 3-18 Hz range (typical tremor frequency range) with the largest power, while $P_{\max}$ is the power at $f_{\max}$. Note that $P_{\max}$ is normalized by the power in the 18-40Hz band (outside the typical tremor frequency range) because the power at $f_{\max}$ can be compared over different trials which might have significantly different power outside the range of interest.

Entropy Measure. The *Sample Entropy* allows investigation of the dynamics of a time series [9]. For a given time series U (here the smoothed sEMG signal) of length L, the sample entropy, denoted as $\text{SpEn}(U, m, r)$, involves two input parameters m and r, which are the pattern length and the similarity criterion, respectively, and is defined as

$$\text{SpEn}(U, m, r) := \lim_{L\to\infty} -\log \frac{B_{m+1}(r)}{B_m(r)}, \tag{3}$$

where $B_{m+1}(r)/B_m(r)$ has the meaning of the conditional probability that two sub-sequences of U that are similar for m points remain similar within a tolerance r at the next point; a lower $\text{SpEn}(U, m, r)$ value reflects a high degree of regularity [9]. Here, $m = 2, r = 0.14\sigma$, where σ is the standard deviation of the sEMG signal.

C. Neural Network (NN)

A NN consists of many neurons/nodes arranged in different layers, referred to as input, hidden and output layers. A NN is a connectionist architecture based on functional and/or structural aspects of the human brain. It consists of a massive set of interconnected neurons in the form of weights, working in unison to solve a specific problem. A NN is an adaptive network that changes its organization based on the information it receives that is defined by: (i) the interconnection between different neurons (weights), (ii) the learning process for updating the weights of the

interconnection between the neurons and the bias at each neuron and (iii) the activation function that converts a neuron weighted input to its output activation, which we introduce next.

To predict the onset of the tremor for the problem at hand, we propose to use a *feed-forward back-propagation NN* [10].

Attributes of the NN. Our NN architecture consists of an input layer with six neurons, two to three hidden layers each with 20 neurons and an output layer with two neurons. Each layer has its own weight matrix, bias vector, input vector and output vector, whose dimensions are given by the number of neurons/nodes in the layer. For the j-th node in a layer, let $\mathbf{w}_j = [w_{j,1}, ..., w_{j,n}]$ be its weight vector, $\mathbf{x}_j = [x_1, ..., x_n]$ be its input vector, b_j be its bias and o_j be its output (here n indicates the number of neurons in the preceding layer). The initialization of the weight/bias is performed by Nguyen-Widrow Randomization technique [11]. This algorithm chooses the initial values of weights and biases in order to distribute the active region of each neuron approximately evenly across the layer's input space.

The *net input* at j-th node is defined by $o_j := \sum_{\ell=1}^{n} w_{j,\ell} x_\ell + b_j$. The net output at j-th neuron is a function $f(o_j)$ of the net signal o_j, where f is the *activation function*. For nodes in the input layer we choose the activation function $f(o_j) = \tanh(o_j)$, while for nodes in the hidden or output layer $f(o_j) = o_j$. The net output values of the nodes in one layer became the input values of the nodes in the next layer (here for notation convenience we omitted the layer index).

Weights and bias are chosen so as to minimize the mean square error (MSE). The *net error* over all P input-output pairs at the N-node output layer is

$$E_{\mathrm{net}}(\mathbf{w}, \mathbf{b}) := \sum_{p=1}^{P} \frac{1}{N} \sum_{k=1}^{N} e_{p,k}^2, \tag{4}$$

where $e_{p,k} := d_{p,k} - y_{p,k}$ is the training error at k-th output layer for the p-th input-output pair (defined as the difference between the desired/ground truth output $d_{p,k}$ and the NN predicted output $y_{p,k} = f(o_{p,k})$, and where $\mathbf{w}$ is the vector with all the weights and $\mathbf{b}$ the vector with all the biases. Here $N = 20$, and $P = 7$ and 8 for PD1 and PD2, respectively.

In our NN design we use the derivative-based Levenberg-Marquardt (LM) algorithm for the learning phase to optimize $(\mathbf{w}, \mathbf{b})$. In the LM learning algorithm the weight and bias update is based on the second-order derivative of the total error function in (4) [12], [13]. It is formulated in term of the Hessian matrix $\mathbf{H}$ of (4) with respect to $(\mathbf{w}, \mathbf{b})$ which is approximated in term of Jacobian matrix $\mathbf{J}$ as $\mathbf{H} \approx \mathbf{J}^T \mathbf{J} + \mu \mathbf{I}$, with $\mathbf{I}$ is the identity matrix. Finally, the weights or bias at iteration k, indicated here as z_k, is updated as

$$z_{k+1} = z_k - \left(\mathbf{J}_k{}^T \mathbf{J}_k + \mu \mathbf{I} \right)^{-1} \mathbf{J}_k \mathbf{e}_k \tag{5}$$

where μ is the non-negative combination coefficient ($\mu \to 0$: Gauss-Newton algorithm; $\mu \to \infty$: steepest decent algorithm) and $\mathbf{e}$ is the error vector that contains all the $e_{p,k}$ in the definition error in (4).

Training of the NN. The NN is trained with the following parameters extracted from a subset of the recorded sEMG and accelerometer signals: Mean power in the frequency band 8-16Hz of the sEMG signal defined in (1); Frequency at maximum power of the sEMG and of the accelerometer signal defined in (2); Peak frequency of the sEMG and of the accelerometer signal defined in (2); and Sample entropy of the sEMG signal defined in (3).

The time series of all extracted parameters from the sEMG and accelerometer signals are divided into two regions: tremor region and no-tremor region, visually separated by the bold vertical line in Fig. 1. In order to define an input-output vector pair, a particular time step (shown in Fig. 1 by thin vertical line) is chosen from the given time series. The time series points (indicated by the arrows in Fig. 1) on the thin vertical line forms the input vector which consists of six elements corresponding to each extracted parameter at the particular time step. If the time step lies in no tremor region the output vector/ground truth is [0,1] otherwise it is [1,0]. For example, in Fig. 1 the chosen time step lies in the no-tremor region and therefore the output vector is [0,1].

Training of the NN is performed by using 7/16 and 8/17 sEMG/acc data sets for PD1 and 2 respectively. The NN is first trained with the original data set in order to adjust the $(\mathbf{w}, \mathbf{b})$ parameters that were initialized with the Nguyen-Widrow Randomization technique. The second phase of training is performed by using the same training set, but this time the different trials are uniformly permuted at random. This second phase serves to further tune the weights and biases. Finally, the so trained NN is fed with the parameters extracted from the sEMG and accelerometer signals as inputs and is used to predict the onset of tremor after the stimulation turned off in those trials not used for training.

III. RESULTS

Classification of Prediction Outcomes. To analyze the prediction performance, each considered trial is classified based on the prediction outcome as follows: let T be the total duration of a trial, t_{on} and t_{off} be the times when stimulation was switched on and off, respectively, and t_{tr} and t_{pr} be the times when tremor was detected and predicted using the NN, during the stimulation off period, respectively. Trials where tremor was detected over the recorded interval after stimulation was off, i.e $t_{\mathrm{tr}} < T$, are denoted as TD (Tremor Detected), while those where tremor was not detected, i.e $t_{\mathrm{tr}} > T$, are denoted as NTD (No-Tremor Detected). In particular we classify as follows. For TD:

• If $[(t_{\mathrm{tr}} > t_{\mathrm{pr}})$ and $(t_{\mathrm{tr}} - t_{\mathrm{pr}}) < \max(5s, 0.5(t_{\mathrm{pr}} - t_{\mathrm{off}}))]$ or $[(t_{\mathrm{tr}} < t_{\mathrm{pr}})$ and $(t_{\mathrm{pr}} - t_{\mathrm{tr}}) < 1s]$, then the algorithm successfully predicts tremor and this outcome is classified as a *true positive (TP)*. This is a bit different from the classical TP definition, in that we require that the prediction be at most 50% of the tremor free off period or 5s (whichever is greater) before actual tremor reappears. This allows penalizing too early prediction outcomes.

• If $(t_{\mathrm{tr}} > t_{\mathrm{pr}})$ and $(t_{\mathrm{tr}} - t_{\mathrm{pr}}) > \max(5s, 0.5(t_{\mathrm{pr}} - t_{\mathrm{off}}))$, then the prediction is too early and the outcome is classified as *false positive (FP)*.

• If $(t_{\mathrm{tr}} < t_{\mathrm{pr}})$ and $(t_{\mathrm{pr}} - t_{\mathrm{tr}}) > 1s$, then the prediction is too late and the outcome is classified as *false negative (FN)*. For NTD:
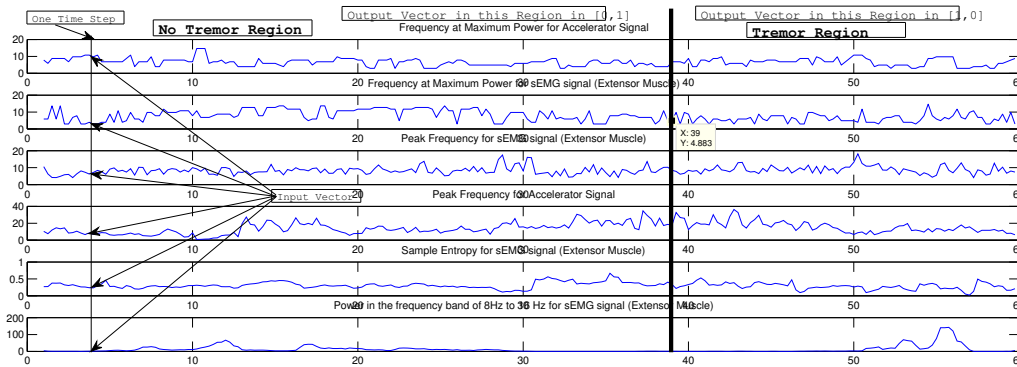
Fig. 1: Input and Output Vectors in Training Data Set. The bold vertical line divides the time series into tremor and no-tremor regions. A time step (shown via a thin vertical line) consists of six input vector elements which are indicated by the arrows. The ground truth output vector is either [0 1] in a no-tremor region or [1 0] in a tremor region.

- If the algorithm does not predict any tremor over the entire interval $T - t_{\text{off}}$, then its classified as *true negative (TN)*.
- If the algorithm predicts tremor over the entire interval $T - t_{\text{off}}$, then its classified as *false positive (FP)*.

Performance Evaluation. For the algorithm to perform well, the total number of TP and TN must be maximized while minimizing FP and eliminating FN. This would achieve the maximum "tremor-free" interval when the stimulation is off. In order to quantify this, the following performance metrics are defined

$$\text{accuracy} = \frac{\#TP + \#TN}{\#TP + \#FP + \#TN + \#FN} \quad (6)$$

$$\text{sensitivity} = \frac{\#TP}{\#TP + \#FN}. \quad (7)$$

For the purpose of this application, sensitivity should be very high (over 90%) because we want to avoid missing any tremor event. At the same time, we also want to have high accuracy, which indicates the ratio of successfully predicted trials to the total number of trials, not only ensures that there is no miss in detecting a tremor event but also the prediction is not too early.

Additionally, the following ratio is defined as $R_{\text{pd}} := \sum(t_{\text{pr}} - t_{\text{off}})/\sum(t_{\text{tr}} - t_{\text{off}})$, where both the summations are over all the trials and R_{pd} is calculated for all trials with a TP/TN/FP, where for the NTD trials we use $t_{\text{tr}} = T - t_{\text{off}}$, $t_{\text{pr}} = \min(T, t_{\text{pr}}) - t_{\text{off}}$. Since R_{pd} is the ratio between the predicted delay to the actual delay in tremor, it provides a measure of how good the prediction is, i.e., a higher value indicates that the predicted delay is closer to the actual delay which is desirable. Table I shows some typical trials to illustrate the outcome classification. In Table II the accuracy, sensitivity and R_{pd} for each PD patient is shown. The overall value is calculated by considering all trials from both patients. Thus the NN predictor achieves a high sensitivity which is highly desirable since we do not want to miss any tremor event. It also achieves an accuracy of at least 75% and an $R_{\text{pd}} > 64$, which means that the predictor loses less than 36% of the actual delay period due to early prediction.

IV. CONCLUSIONS

In this paper, we showed that a NN can serve as a simple yet accurate model for the prediction of tremor

TABLE I: Some typical trials and their classification for PD1 and PD2. Legend: R(rest), P(posture), A(action) while 1 and 2 refers to PD1 and PD2, respectively.

Trial#	$t_{\text{off}} - t_{\text{on}}$	$t_{\text{tr}} - t_{\text{off}}$	$t_{\text{pr}} - t_{\text{off}}$	$t_{\text{pr}} - t_{\text{tr}}$	N	o/p type
R1	31	21	21.75	-0.75	80	TP
P1	41	18	7.25	10.75	70	FP
A1	31	11.5	8.5	3	70	TP
R2	47	16	15	1	80	TP
P2	52.75	5.25	3.75	1.5	80	TP
A2	52.5	11.5	3.25	8.25	70	FP

TABLE II: Prediction performance with ratios in %.

Patient#	N	TP,TN,FP,FN	accuracy	sensitivity	R_{pd}
PD1	16	12,0,3,1	75	92.3	64.9
PD2	17	13,0,3,1	76.5	92.9	69.7
overall	33	25,0,6,0	75.8	92.5	67.3

in Parkinson's patients. We tested the feed-forward back-propagation NN model with inputs from non-invasively measured sEMG/accelerometer signals. The accuracy of the tremor prediction after the stimulation turned off is about 75.8% with 92.3% sensitivity. For future work we will investigate whether the performance can be improved by other more advanced neural network models such as the LAMSTAR network.

REFERENCES

[1] Jankovic J., Tolosa E. Parkinson's disease and movement disorders. Lippincott Williams and Wilkins Philadelphia, 2007.
[2] Volkmann J, Herzog J, Kopper F, Deuschl G. Introduction to the programming of deep brain stimulators. Mov Disord. 2002;17 Suppl 3:S181-7.
[3] Graupe D, Basu I, Tuninetti D, Vannemreddy P, Slavin KV. Adaptively controlling deep brain stimulation in essential tremor patient via surface electromyography. Neurol Res. 2010;32(9):899-904.
[4] Basu I, Tuninetti D, Graupe D, Slavin KV. Adaptive control of Deep Brain Stimulator for Essential Tremor: entropy-based tremor prediction using surface-EMG. Conf Proc IEEE Eng Med Biol Soc. 2011;2011:7711-4.
[5] Pullman SL, Goodin DS, Marquinez AI, Tabbal S, Rubin M. Clinical utility of surface EMG: report of the therapeutics and technology assessment subcommittee of the American Academy of Neurology. Neurology. 2000;55(2):171-7.
[6] Srinivasan V, Eswaran C, Sriraam N. Approximate entropy-based epileptic EEG detection using artificial neural networks. IEEE Trans Inf Technol Biomed. 2007;11(3):288-95
[7] Waxman JA, Graupe D, Carley DW. Automated prediction of apnea and hypopnea,using a LAMSTAR artificial neural network. Am J Respir Crit Care Med. 2010;181(7):727-33.
[8] Sturman MM, Vaillancourt DE, Metman LV, Bakay RA, Corcos DM. Effects of subthalamic nucleus stimulation and medication on resting and postural tremor in Parkinson's disease. Brain. 2004;127(Pt 9):2131-43.
[9] Richman JS, Moorman JR. Physiological time-series analysis using approximate entropy and sample entropy. Am J Physiol Heart Circ Physiol. 2000;278(6):H2039-49.
[10] Graupe D., Principles of Artificial Neural Networks, 2nd Edition, World Scientific, 2007.
[11] Nguyen D. and Widrow B. Improving the learning speed of 2-layer neural neural network by choosing initial values of the adaptive weights. In Proceedings of IEEE International Joint Conference on Neural Networks, 5(4):595-603, 1992.
[12] Levenberg K. A method for the solution of certain problems in least squares. Quarterly of Applied Mathematics, Vol.5, page 164-168, 1944.
[13] Marquardt D. An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics, Vol.11(2), page 431-441, 1963.