

ARTUR SZYMAŃSKI  
ANNA KUBIS  
ANDRZEJ W. PRZYBYSZEWSKI

## DATA MINING AND NEURAL NETWORK SIMULATIONS CAN HELP TO IMPROVE DEEP BRAIN STIMULATION EFFECTS IN PARKINSON'S DISEASE

### Abstract

*Parkinson's Disease (PD) is primary related to substantia nigra degeneration and, thus, dopamine insufficiency. L-DOPA as a precursor of dopamine is the standard medication in PD. However, disease progression causes L-DOPA therapy efficiency decay (on-off symptom fluctuation), and neurologists often decide to classify patients for DBS (Deep Brain Stimulation) surgery. DBS treatment is based on stimulating the specific subthalamic structure: subthalamic nucleus (STN) in our case. As STN consists of parts with different physiological functions, finding the appropriate placement of the DBS electrode contacts is challenging. In order to predict the neurological effects related to different electrode-contact stimulations, we have tracked connections between the stimulated part of STN and the cortex with the help of diffusion tensor imaging (DTI). By changing a contacts number and amplitude of stimulus (proportional in size to stimulated area), we have determined connections to cortical areas and related neurological effects. We have applied data mining methods to predict which contact (and at what amplitude) should be stimulated in order to improve a particular symptom. We have compared different data mining methods: Weka Random Forest classifier and Rough Set Exploration System (RSES). We have demonstrated that the Weka classifier was more accurate when predicting the effects of stimulations on general neurological improvements, while RSES was more accurate when using specific neurological symptoms. We have simulated other effects of stimulation related to the interruption of pathological oscillation in the basal ganglia found in PD. Our model represents possible STN neural population with inhibitory and excitatory connections that have pathologically synchronized oscillations. High-frequency electrical stimulation has interrupted synchronization. something that is also observed in PD patients.*

### Keywords

Parkinson's Disease, Deep Brain Stimulation, neural computation, data mining

### Citation

Computer Science 16 (2) 2015: 199–215

## 1. Introduction

DBS is a well-established method of PD treatment, and it is most-often applied when pharmacological treatment is no longer effective for a particular patient. Despite this, DBS has been widely used for more than two decades. There are still a number of factors (e.g., exact electrode-contact localization that may significantly affect the effectiveness of this method).

The target of the stimulation is localized in the midbrain and is not visible in MRI for most patients. Therefore, an approximate estimation of STN coordination is, in most cases, based on nearby visible MRI structures (e.g., red nuclei). A more-precise STN position was found by intraoperative recordings of changes in neural activity associated with the depth of the microelectrode [4]. In addition, the STN is divided into three regions that are related to motor, associative, and limbic functions. The main target of our procedure is the motor STN region.

The purpose of the study was to predict optimal parameters of stimulation in MRIs for each individual PD patient. In our research, we apply data-mining techniques to compare the neurological symptoms measured by doctors with the anatomical data found by MRI/DTI measurements. Previously, data-mining techniques have already been applied in DBS procedures [4]. Also, DTI methods were used to analyze DBS effects in different disorders [5, 6], among others in PD [17]. We have also performed a simulation of neuronal population similar to that in the STN. Our model was created with the help of the Numerical Python package, and it is an extension of a simple model of spiking neurons (proposed by Izhikevich [9]).

## 2. Methods

In this research, we have analysed the DTI, MRI, and CT data of ten patients measured before and after DBS surgery. All patients had their data separated into two sessions; one taken before DBS surgery and one afterwards. Together, this makes 20 objects to be analyzed in later stages using the data-mining approach. As this study is preliminary, we expect better results with an increase in the number of patients.

As the first step, we performed brain registration and tractography seeding using 3DSlicer (Harvard Medical School, Boston, MA). In this step, we selected appropriate subsets of pre- and post-operative images. This step helped us to assess the position of electrodes visible only in postoperative data. In the next step, we registered pre-operative DTIs with post-operative MRIs.

### 2.1. MRI registration

In order to perform tractography seeding for the given contacts, we need to first register postoperative to preoperative images acquired via MRI or CT. These scans originate from standard procedures run before and after DBS surgery. The first session is taken before the procedure to help assess the position of STN by the surgical team, and the second to confirm proper placement of the electrode and its contacts.

Following this stage, we obtained three relevant sets of images for each patient: preoperative MRI, postoperative MRI or CT, and preoperative DTI. Each set of images contains multiple subsets that were taken in the same session with different parameters. From all subsets, we chose those taken with three-dimension acquire mode and having the smallest slice spacing and width. Also, an important factor in the selection of the final images was the visibility of brain structures.

Based on these criteria, we selected one preoperative MRI, one preoperative DTI, and one postoperative MRI. In the next step, we had to perform image data registration of these data sets, as information regarding the position of electrodes is contained only on the postoperative images, and DTI data is available only in the preoperative data sets. This task was carried out using the ‘BRAINFit’ module of 3DSlicer. We have performed MRI registration of the same patient but with the appropriate corrections, as imaging was taken in different conditions and MRI sessions were held on different days (pre- and post-OP) (see Table 1).

**Table 1**

Table shows major non-default parameters for 3D Slicer BRAINFit module used to register postoperative to preoperative MRI images of patients.

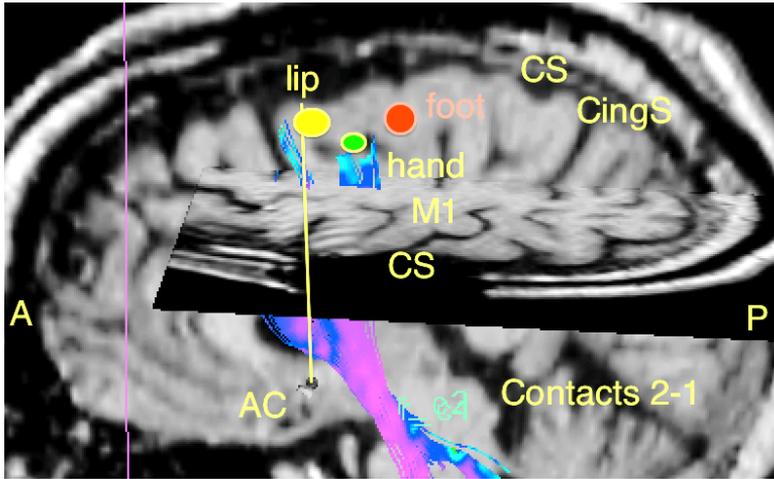
Parameter name	Value
Initialize transform mode	Use center of head align Registration phases Rigid
Number of samples	100 000
Max iterations	1500

After multiple-layer registration, 3DSlicer was used to assess transformation accuracy through the visible brain structure coverage. The level of structure coverage assessed the accuracy of the transformation manually.

## 2.2. Tractography analysis

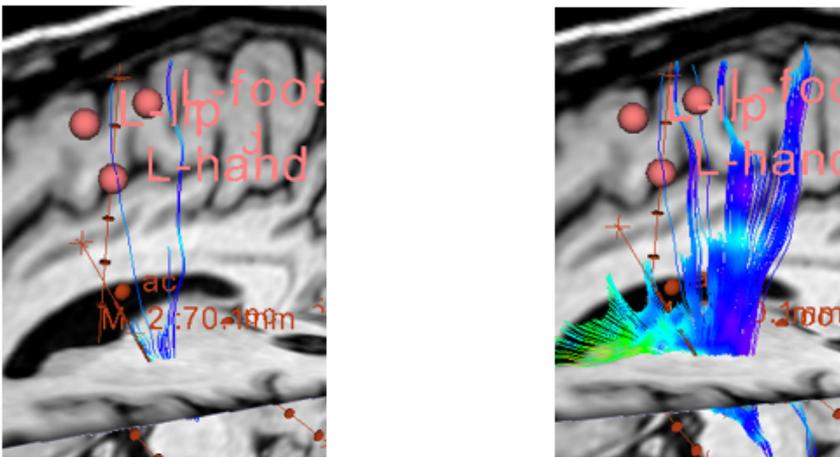
After postoperative to preoperative images were registered, we transformed preoperative DTI images into data in Slicer to generate tracts for given points of interest (POI). In our case, POI were electrode contacts. For this purpose, we applied DWI to the DTI estimation module using the weighted-least-squares algorithm. After obtaining a proper DTI dataset, we needed to locate electrode contacts in order to generate relevant tracts to our regions of interest (ROI). In order to do this, we needed to assess their position using electrode markings visible in the postoperative data. To assess contact positions, we used the ruler tool with the appropriate dimension parameters for the given electrode type. After setting up the POI relevant for different contacts, we were ready to generate stimulated tracts for the selected contact. We used Tractography Interactive Seeding module for this task.

Before the next step we needed to localize our ROI. In this work, we focused mainly on regions of primary and supplementary motor corteces, in particular, regions with lip, hand, and foot somatotopic representations [3, 15].



**Figure 1.** In the above figure, we can see a patient's brain images annotated with different brain structures and markings that helped us find ROIs. Three ROIs that were most interesting for us are marked by three round shapes: yellow – lip, red – foot, green – hand.

Manually localizing different structures in the brain assessed these regions. Based on their position, we were able to localize target ROIs (see Fig. 1).



**Figure 2.** This figure presents two sets of tracts gathered for each patient. The image on the left shows a minimal set consisting of only few tracts going into ROI, and was generated with region size 1mm and stopping value 0.25. The right image shows a maximum set of tracts, generated with radius size 6mm and stopping value 0.25. The generation parameters change between subjects, depending on brain physiology.

In the next step, we harvested two sets of data for a given patient. The first set was performed with parameters that generated a small number of tracts into primary and supplementary motor cortices. In the second set, the number of tracts was accounted for more than 30. For each given patient, parameters for both sessions were stored and used in later stages of this experiment. The parameters that were used during the generation of proper tractography were seed spacing and stopping value. The first parameter (seed spacing) was used as a radius from POI for generating tractography. The second parameter (stopping value) was used as a parameter for ceasing generation of the given tract (see Fig. 2).

### 2.3. Atlas registration

In order to extract the exact position of STN in relation to an implanted electrode, we need to assess the stimulated structure position in the patients MRIs. In order to do this, we will need to perform series of registrations between patient MRI and two atlases.

In most cases, atlases are based on one or more specimens and are correlated into one single data structure. In the case of the SPL-PNL atlas, we will register its MRI representation to a patients pre-OP MRI scan. The primary difference between this one and previous registrations is the fact that we are trying to register different brains that have structures of different sizes and locations. Attempts to perform a full registration of such images would result in largely deformed unusable transform. But, as far as our analysis is concerned, we only need to determine the accurate position of the few structures in our area of interest. To increase the accuracy of our registration we performed it only for the given parts of the MRIs. Since we expected a different proportion of registered structures, it was better to perform two different types of registration: linear [10] and bspline [13]. Linear registration has good accuracy in the registration of similar shapes with different locations, while bspline gives better results when co-registering images with different parameters. After registration, we manually assessed the precision of the registration transform by examining the overlaying structures.

After having a patients pre-OP MRI scan registered to the general brain atlas included in SPL-PNL, we needed to locate STN that was not included in the original SPL-PNL atlas. To achieve this, we needed to register the SPL-PNL atlas to the Krauth et al. thalamus model [11] that includes the exact position of the STN. This model was created based on multiple specimens and provides more-accurate structure localization. In this case, we used surface-model registration [14]. As a target for our registration, we selected structures that could be defragmented from each model and existed in both atlases. Specifically, we registered each side of the brain by the surfaces of the red nucleus and thalamus. Since we were again registering two different brains, we applied a few different algorithms for registration: Rigidbody, Similarity, and Affine [14]. After completion of the procedure, we estimated the level of coverage between structures from both models and picked a registration transform with the largest part that overlays both models.

After performing the registration of a patients pre MRI scan with the SPL-PNL atlas and the SPL-PNL atlas to Krauths thalamus model (and correlating this data), we are able to position STN on the patients brain images.

## 2.4. Data mining

After the appropriate data from the 3DSlicer was harvested, we correlated it with the neurological data acquired by the doctors assessment of the patients conditions. This data included classification with the Unified Parkinson's Disease Scale (UPDRS) and logical information about whether the given scale rating was obtained for a patient with the best medical therapy (BMT) and/or DBS enabled. Following this schema, many patients had several sessions of neurological data correlated with the previously described attributes from the 3DSlicer.

After collecting all of the data, we used Weka 3 [7] to analyze it and predict the given attributes of the stimulation. In order to do this, we needed to first prepare a decision table containing all of the attributes. Part of this table can be seen in Table 2.

**Table 2**

In the table below, we can see part of the decision table containing the following attributes: DBS – indicates if a given session was acquired with DBS on/off; BMT – pharmacological treatment on/off; UPDRS III – total motoric classification of patient' condition; UPDRS 30 – UPDRS specific classifier for postural stability, 0 means no disability, 4 severe disability; Slicer MAX L/R region size – radius for generating tractography for the selected contact for a given patient with a MAX count of tracts; Slicer MAX L/R lip tracts – count of tracts going to a given ROI in patient brain; L/R selected contact – selected contact for a given hemisphere; L/R selected amplitude – selected amplitude for a given hemisphere.

Patient number	#10	#10	#25
DBS	0	1	0
BMT	1	1	0
UPDRS III	13	–	16
UPDRS 30 – postural stability	3	–	1
Slicer MAX L – region size	6	6	8
Slicer MAX L – lip tracts	20	20	22
Slicer MAX R – region size	5	5	8.5
Slicer MAX R – lip tracts	15	15	35
L – selected contact	1	1	1.5
L – selected amplitude	2	2	1.5
R – selected contact	1	1	1.5
R – selected amplitude	3.2	3.2	1.5

The full table consists of 50 attributes and 20 objects from ten patients. In this research, we performed a number of experiments using parts of this table to predict decision attributes. Those attributes were as follows: the selected electrode contact

and the amplitude of the stimulation. In our experiments, we used the Random Forest classifier and a 4-fold classification method.

## 2.5. Neuronal population model

Our model simulates the pathological activity of neurons in the case of PD. It is known that, in PD patients, neurons showed synchronized bursts in contrast to healthy patients who have unsynchronized spiking activity [2, 12].

We based our simulation on the simple model of spiking neurons by Izhikevich [9]. In our version, we converted the original Matlab application to Python with the use of the Numpy package, which provides an environment similar to Matlab using Python interpreter.

This model was selected due to its relatively low complexity as compared to other models of neuronal cells (like the Morris–Lecar or Hodgkin–Huxley models). As already demonstrated, the Izhikevichs model has similar properties as models with higher complexity [12].

The network consists of two populations. The first takes the role of the stimulated part of STN, and the other simulates a brain cortex. Each population consists of 1,000 neurons, which are split into inhibitory (200) and exhibitor (800) neurons. This is exactly the split that Izhikevich proposed in his model that consisted of only one population of neurons. Both populations were organized in a vector shape.

With the simple spiking model, we were able to simulate various kinds of neuronal activity, including regular spiking, fast spiking, thalamo-cortical spiking, and many more. Based on the simple spiking neuron simulation, we selected a heterogeneous model [8, 9] for driving our simulation.

The neuron signal is given by the following equation:

$$v' = 0.04v^2 + 5v + 140 - u - I \quad (1)$$

With initial variables being reset after spike which is indicated by variable  $v$  being greater than 30. Reset behaviour is given by following equation:

$$v > 30mV, \text{ then } \{ v \leftarrow cu \leftarrow u + d \quad (2)$$

In these equations, we introduced two dimensionless variables ( $v$  and  $u$ ) and four parameters ( $a$ ,  $b$ ,  $c$ , and  $d$ ). Parameter  $v$  represents membrane potential and parameter  $u$  membrane recovery variable for a given neuron. Variable  $a$  represents a time scale of recovery variable  $u$ ,  $b$  stands for sensitivity of  $u$ ,  $c$  represents after spike reset value of  $v$ , and  $d$  represent the same property as  $c$  for  $u$ . By using the selected parameters, we can mimic different kinds of spiking. In the case of the heterogeneous model, we used the following base parameters (which are randomly modified in the scope of the defined borders as described in the original model). All parameters were set using guidelines provided in the simple model of neuron spiking by Izhikevich [9].

Connections inside our simulation were modeled like Izhikevich proposed in the simple model of spiking neurons as a connection with random weights between each

neuron inside one population. Additionally, since we have two populations, we defined a connection between them as one to one, making each neuron from one population connected to exactly one from the other network. These were also weighted using random distribution.

Our simulation starts with setting up the necessary variables as previously defined. The example below shows a definition of only the specific variables for a given population as well as the structures that are relevant for both.

```

Ne = 800 # exhibitor neuron count
Ni = 200 # inhibitor neuron count
re = np.random.rand(Ne, 1) # uniform random distribution for exhibitory neurons
ri = np.random.rand(Ni, 1) # uniform random distribution for exhibitory neurons

stimStart = 400 # number of neuron from which we start DBS
stimStop = 450 # number of neuron from which we stop DBS
stimDelta = stimStop - stimStart # count of stimulated neurons

stimTime = 1000 # time of simulation
dbsStart = math.floor(stimTime/2) # start of DBS

# neuron model parameters set
a = np.append(0.02 * np.ones(Ne, int), 0.02 + 0.08*ri)
b = np.append(0.2 * np.ones(Ne, int), 0.25 - 0.05*ri)
c = np.append(-65 + 15 * np.power(re, 2), -65*np.ones(Ni, int))
d = np.append(8 - 6 * np.power(re, 2), 2*np.ones(Ni, int))
# setting population 1 related variables
# connection inside population
S1 = np.concatenate((0.5*np.random.rand(Ne+Ni, Ne), -np.random.rand(Ne+Ni, Ni))
v1 = -65*np.ones(Ne+Ni, int) # Initial values of v
u1 = b*v1 # Initial values of u
firings1 = [] # spike timings
sn1 = [v1] # neuron signal recording

...

# connection between two populations
S12 = np.append(np.random.rand(Ne), -1*np.random.rand(Ni), 1)

dbs = np.ones(Ne+Ni)*0 # initial values of stimulation vector
Stemp = np.ones(Ne+Ni)*0 # initial values of stimulation between population

synch1 = [] # synchronization measure of population 1

```

As seen in the code above, we define both the earlier-mentioned parameters as well as the initial variables for storing information regarding the current state of simulation ( $v$  and  $u$ ) as well as the structures for storing results for analysis of results, as for example `sn1` or `firings`.

In the next step, we start our simulation based on the parameters set in the first section. The code below presents population 1 workflow during one step of simulation.

```

for t in range(1, stimTime):
    # first population
    I1 = np.append(5*np.random.randn(Ne, 1), 2*np.random.randn(Ni, 1)) # thalamic
    fired1 = np.nonzero(v1>=30)[0] # catching spikes firings into vector
    firings1 += zip([t]*len(fired1), fired1) # storing firings with time

    runVTeta = 0 # variable holding sum of e**(i*teta) in time t
    runTeta = 0 # variable holding sum of teta of all neurons in time t
    for x in range(stimStart, stimStop):
        teta = np.arctan(v1[x] / t)
        runVTeta += ( np.cos(teta) + 1j * np.sin(teta) )
        runTeta += teta

    # calculation of measure for given t
    synchMeasure = ( ( runVTeta / (stimDelta) )
/ ( ( np.cos(runTeta / (stimDelta) ) + 1j*np.sin(runTeta / (stimDelta))) ) )
    # storing synch measure for given t
    synch1 += [(t, synchMeasure)]

    # calculating driving effect between populations, 1 -> 2
    Stemp = np.ones(Ne+Ni)*0
    if t != 1:
        Stemp[fired1] = S12[fired1]
        Stemp = np.abs(Stemp*v1)

    # calculation based on Izhikevich equation
    v1[fired1] = c[fired1]
    u1[fired1] = u1[fired1] + d[fired1]
    I1 = I1 + np.sum(S1[:,fired1], 1) # calculation of thalamic input + potenti
    # calculates DBS stimulation value for selected period
    if t > dbsStart:
        dbs = np.concatenate( (np.ones(stimStart)*0,
                                np.random.randn(stimDelta)*50,
                                np.ones((Ne+Ni) - stimStart - (stimDelta))*0) ,
        # calculation based on Izhikevich equation with DBS, step 0.5 ms for numeri
        v1 = v1 + 0.5*(0.04*np.power(v1, 2) + 5*v1 + 140 - u1 + I1 + dbs)
        v1 = v1 + 0.5*(0.04*np.power(v1, 2) + 5*v1 + 140 - u1 + I1 + dbs)
        u1 = u1 + a*(b*v1 - u1)
    sn1 += [v1] # collection of v potential

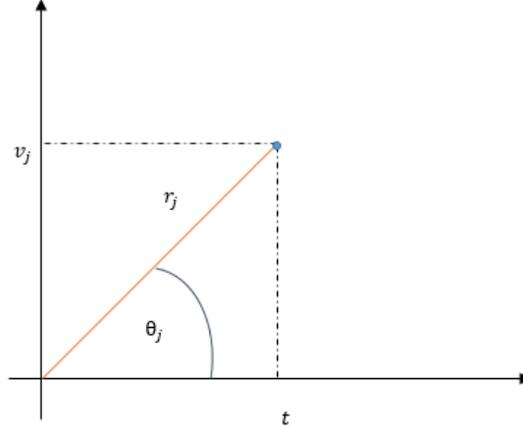
```

We start with the loop over the previously configured timeframe, first initializing the vector for thalamic input I1 and then catching all firings in a given step, writing them with time to the appropriate structure. Thalamic input is generated with Gaussian random distribution.

Next, we calculate and store synchronization measure as well as  $v$  and  $u$  variables basing on the initial equation for resetting spike firings.

The measure of synchronization is constructed at given time  $t$  for a set of  $N$  neurons with respective potentials  $v_1(t), v_2(t), \dots, v_N(t)$  at given time  $t$ .

We can describe the potential of the neuron at given time  $t$  by its coordinates in a polar coordinate system, using radius  $r_j(t)$  and angle  $\theta_j(t)$  (see Fig. 3).



**Figure 3.** The figure shows the potential in the polar coordinate system.

Then:

$$v_j(t) = r_j(t) \cdot e^{i \cdot \theta_j(t)}. \quad (3)$$

If we sum up the potentials of all the neurons at given time  $t$ , we get

$$\sum_{j=1}^N v_j(t) = \sum_{j=1}^N r_j(t) \cdot e^{i \cdot \theta_j(t)}. \quad (4)$$

Therefore, the average potential of the single vector at given time  $t$  would be

$$v_{avg}(t) = \frac{1}{N} \sum_{j=1}^N r_j(t) \cdot e^{i \cdot \theta_j(t)}. \quad (5)$$

To be able to effectively use this formula to measure the synchronization of the neurons, we need to normalize it. In order to normalize the measure, we will scale it by the value of the potential in the center of mass of all of the neurons potentials. This potential would be described by some radius  $r(t)$  and some angle  $\theta(t)$ :

$$v(t) = r(t) \cdot e^{i \cdot \theta(t)} \quad (6)$$

The measure of the synchronization will be then

$$R(t) = \frac{v_{avg}(t)}{v(t)} = \frac{\frac{1}{N} \sum_{j=1}^N r_j(t) \cdot e^{i \cdot \theta_j(t)}}{r(t) \cdot e^{i \cdot \theta(t)}} \quad (7)$$

As  $r_j(t) = \frac{t}{\cos\theta_j(t)}$ , the final formula will take the form of

$$R(t) = \frac{v_{avg}(t)}{v(t)} = \frac{\frac{1}{N} \sum_{(j=1)^N} \frac{t}{\cos\theta_j(t)} \cdot e^{i\theta_j(t)}}{\frac{t}{\cos\theta(t) \cdot e^{i\theta(t)}}} = \frac{\frac{1}{N} \sum_{(j=1)^N} (t + i \tan\theta_j(t))}{t + i \cdot \tan\theta(t)} \quad (8)$$

In the case of perfect synchronization, all of the neurons have the same potential equal to the potential at the center of mass of potentials. Therefore, in this case  $R(t) = 1$ .

The more desynchronized the neurons are, the bigger the difference between nominator and denominator of the formula of  $R(t)$  is. As a result,  $R(t)$  becomes closer to 0 with a growth of the synchronization of neurons.

In order to calculate measure  $R(t)$ , we need to know the value of angle  $\theta_j(t)$ . Given the value of potential  $v_j(t)$ , we can calculate it in the following way:

We know that  $v_j(t) = \frac{r_j(t)}{\sin\theta_j(t)}$  and  $r_j(t) = \frac{t}{\cos\theta_j(t)}$  and  $r_j(t) = \frac{t}{\cos\theta_j(t)}$ .

Then  $v_j(t) = t \cdot \tan\theta_j(t)$ . The measure of synchronization is a modified version of the work of Latteri et al. [12].

Afterwards, if we reach a previously defined time for stimulation, we calculate the vector for the stimulation current with a given mask to mimic the stimulation of selected parts of STN. The stimulation value is calculated with Gaussian distribution being adequately amplified to deliver the expected network desynchronization.

Finally, we calculate the  $v$  variable potential and store the whole vector in the sn1 structure. The variable is calculated using the Izhikevich equation, with the addition of the previously calculated stimulation vector  $db_s$ .

The second population simulation step looks similar to the first. The only difference is that DBS does not take place there. Moreover, firings from the first population are added to the thalamic input of the second population. An example of the calculation of thalamic input for the second population:

```
# calculation of thalamic input + potential of other neuron firings from both I
I2 = I2 + np.sum(S2[:,fired2], 1) + Stemp
```

## 3. Results

### 3.1. Data mining

Preliminary results of this research have demonstrated that, depending on selected subsets of attributes from the full decision table, we were able to obtain accuracy levels reaching 70%–80%. We used a few different subsets of attributes from the decision table in order to see which were the most efficient in predicting decision attributes. The following subsets with the highest accuracy were used:

- Prediction of the selected contact amplitude/number using UPDRS III and attributes from Slicer. Dataset consists of 16 attributes and 20 objects.

- Prediction of the selected contact amplitude/number using Slicer attributes from left hemisphere and opposite side UPDRS, For example, analyzing the left side lip ROI count of tracts and specific UPDRS attributes from the right side of the body, like right-hand rigidity.

A more-efficient subset appeared to be the prediction of selected contact amplitude by using UPDRS III, which provided 85% accuracy (as compared to 70% accuracy for the predicting contact number). Classifications using specific UPDRS values were much less accurate for predicting number and amplitude, at 35% and 15% respectively. We have compared these results with our previous research using RSES [1], and in comparison, using UPDRS III subset results for predicting contact amplitude are 10% more accurate using Wekas Random Forest algorithm. The exact results for predicting specific attributes in both systems can be seen in Table 3.

**Table 3**

In the table below, we have shown a comparison of RSES results to WEKAs. In rows regarding specific UPDRS attributes, we always performed tests on opposite-side Slicer parameters to UPDRS data.

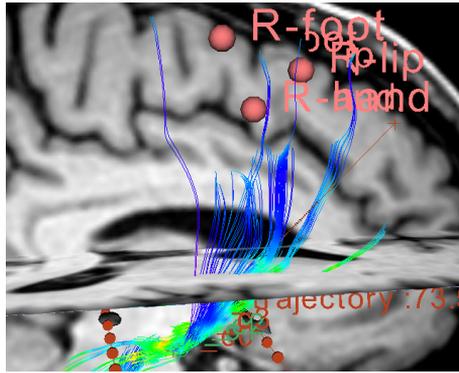
Contact property	Attributes sets	RSES	Weka
Amplitude	UPDRS III + Slicer	75%	85%
Number	UPDRS III + Slicer	65%	70%
Amplitude	Specific UPDRS + Slicer	80%	15%
Number	Specific UPDRS + Slicer	62.5%	35%

As we can see, RSES results were more accurate when predicting stimulated contact parameters based on specific UPDRS values, whereas Weka was more accurate in predicting based on UPDRS III.

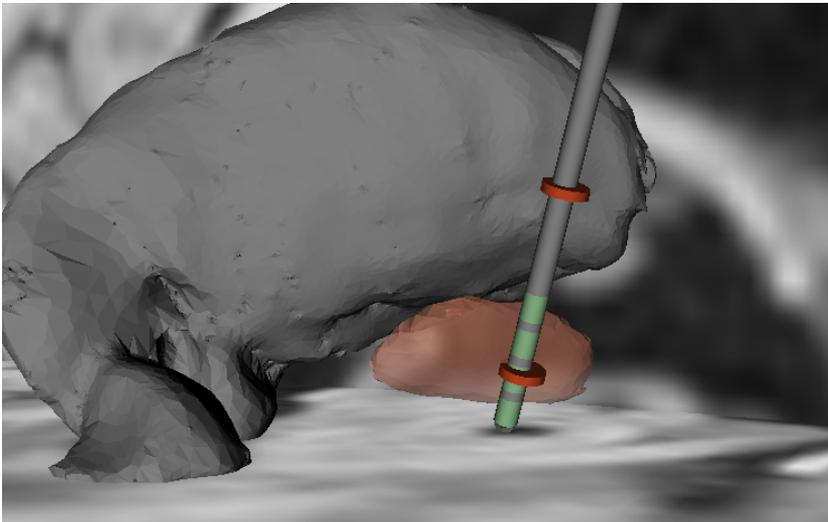
### 3.2. DBS simulation using neuronal population (NP)

We have applied the NP model defined before in the case of one of the patients who was included in our data-mining experiment. The entry point for the analysis was the number and location of tracts outgoing from the stimulated contact and their somatotopic connections. The goal of the simulation was the creation of such an NP configuration that would help in assessing the stimulation effects. In this study, we present the results of sample simulation based on one patient that would use our model to show alleviation of patient symptoms by stimulation of selected parts of the network.

First, we need to analyze the DTI projections of Patient #9. In Figure 4, we can see a projection of tracts going into motor and supplementary motor cortices, reaching regions responsible for both lower limbs (foot) and upper limbs (hand). The location of the stimulation electrode in relation to patient STN can be seen in Figure 5. The selected stimulation contact, in this case, is number 1 with 2.3 V amplitude.



**Figure 4.** The picture above shows DTI projections from a contact selected by the neurologist that leads into the supplementary motor cortex.



**Figure 5.** The picture above shows the position of the electrode in regards to STN for patient #9. In this case, the neurologist selected contact #1, which we can see is adjacent to the mid and lower parts of the structure.

From the neurological data collected during a physical examination, we know that the stimulation of this contact caused improvement in left-hand fingertip movement as well as in gait and tremor issues.

On the basis of this information, we used our NP model to simulate previously-mentioned synchro and desynchronization of neurons that are responsible for given motor cortex functions. The first step was the mapping of the previously-defined neuron populations into a somatotopic organization of STN and cortex.

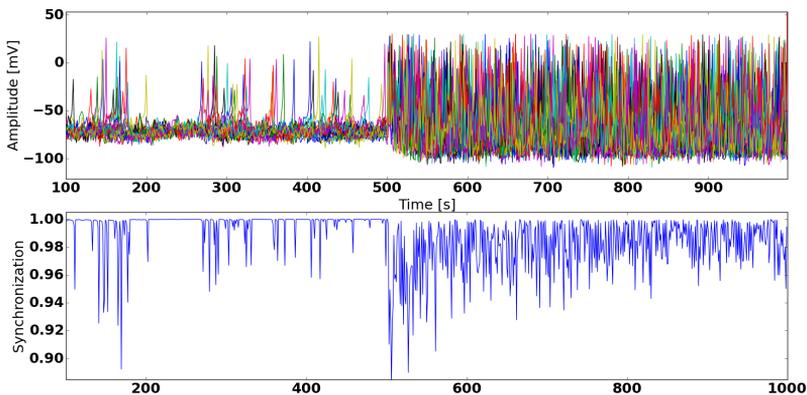
As for the mapping of population 1 (which in our simulation was responsible for STN functions), we used the fact that it is organized in a vector. This allows us to map the length of the electrode that was located inside STN to regions which were stimulated. Based on current knowledge regarding somatotopic organization of STN, most somatotopic connections are localized in the dorsal upper two thirds of the structure [16]. The upper third of the structure is mainly responsible for the legs, while upper-limb neurons were more often located in both mid and upper parts.

Based on this information, we can hypothesize that stimulation of the contact with index 1 in correlation with the electrode as seen in figure 4 will stimulate parts of the STN that correspond to movements of the upper limbs (f.e., arm). We need to note here that it is difficult to assess exact regions that are stimulated by selected contact amplitude because the exact correlation between stimulation radius and stimulation amplitude is unknown.

As previously mentioned, we assume that 1,000 neurons in our first population represent those parts of STN that were adjacent to the stimulation electrode. Assuming that we have dorsal indexing of neurons, we want neurons between 200 and 300 to correspond to our stimulation region. Using the network model described earlier, we can achieve it by setting the appropriate values to variables `stimStart` and `stimStop`.

The resulting simulation showed desynchronization in selected regions and, at the same time, simulated alleviation of the given patient symptoms.

In Figure 5, we can see the signal corresponding to 100 neurons that were the target of the stimulation starting in  $t = 500$ . As we can see in the chart, spikings are grouped into bursts occurring in a periodical synchronized manner during the first half of simulation. At the same time, this indicates incorrect behavior in STN. While after turning on the stimulation, we can observe constant desynchronization of the selected part of the network, indicating that stimulation has brought the expected effects. Both populations showed similar effects, while only one of them was actually stimulated while the other was driven by it.



**Figure 6.** The chart above shows a simulated neuron' signal during the process of stimulation.

Figure 6 shows us how the signal of simulated neurons change in the course of DBS. Stimulation was turned on in  $t = 500$ . The upper chart shows the value of variable  $v$  in time  $t$ , where the one below shows the value of synchronization measure in time as defined earlier.  $R = 0$  stands for no synchronization, whereas  $R = 1$  stands for full synchronization. We observe that, before stimulation, neurons remain synchronized most of the time; this may be related to PD symptoms. After stimulation, synchronization is permanently decreased and does not reach 1.

## 4. Conclusions and future work

### 4.1. Comparison of data-mining methods

In this work, we have presented our recent results for predicting contact parameters using a Weka Random Forest classifier and compared them to our previously published results using RSES. Using different subsets of data, we have obtained different accuracy, with Weka being more accurate when predicting specific UPDRS attributes and RSES being more accurate when using UPDRS III. Since these results have been based on a small data set containing only 20 objects, further work is required to perform more creditable statistics and verification in regards to over fitting.

### 4.2. NP simulation model

We have presented an example of the use of the proposed NP while simulating DBS to alleviate selected patient' symptoms. This simple case was based on data received from one patient and aimed only at presenting the possibilities of this approach in verification and prediction of simulation parameters. We did it in a similar way as previously done using our data-mining approach.

Furthermore, this approach needs more work to be verified against multiple patients and correlated with neurological data. Multiple elements of this kind of simulation need to be taken into consideration; e.g., the exact localization of STN in relation to electrodes or the correct assessment of neurological effects of stimulation and parameters of tractography projections.

## Acknowledgements

*This work was partly supported by grant Dec-2011/03/B/ST6/03816 from the Polish National Science Center.*

## References

- [1] Bazan J. G., Szczuka M.: RSES and RSESLib – A Collection of Tools for Rough Set Computations. In: W. Ziarko, Y. Yao, eds., *Rough Sets and Current Trends in Computing*, no. 2005 in Lecture Notes in Computer Science, pp. 106–113, Springer–Berlin–Heidelberg, 2001, [http://link.springer.com/chapter/10.1007/3-540-45554-X\\_12](http://link.springer.com/chapter/10.1007/3-540-45554-X_12).

- [2] Cassidy M., Mazzone P., Oliviero A., Insola A., Tonali P., Lazzaro V.D., Brown P.: Movement-related changes in synchronization in the human basal ganglia. *Brain*, vol. 125(6), pp. 1235–1246, 2002, <http://dx.doi.org/10.1093/brain/awf135>.
- [3] Cauda F., Giuliano G., Federico D., Sergio D., Katiuscia S.: Discovering the somatotopic organization of the motor areas of the medial wall using low-frequency BOLD fluctuations. *Human brain mapping*, vol. 32(10), pp. 1566–1579, 2011, <http://dx.doi.org/10.1002/hbm.21132>.
- [4] Ciecierski K., Raś Z.W., Przybyszewski A.W.: Selection of the Optimal Microelectrode during DBS Surgery in Parkinson's Patients. In: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, M. Kryszkiewicz, H. Rybinski, A. Skowron, Z.W. Raś, eds., *Foundations of Intelligent Systems*, vol. 6804, pp. 554–564, Springer, Berlin-Heidelberg, 2011, [http://rd.springer.com/chapter/10.1007/978-3-642-21916-0\\_59](http://rd.springer.com/chapter/10.1007/978-3-642-21916-0_59).
- [5] Coenen V.A., Allert N., Mädler B.: A role of diffusion tensor imaging fiber tracking in deep brain stimulation surgery: DBS of the dentato-rubro-thalamic tract (drt) for the treatment of therapy-refractory tremor. *Acta Neurochirurgica*, vol. 153(8), pp. 1579–1585; discussion 1585, 2011, <http://dx.doi.org/10.1007/s00701-011-1036-z>.
- [6] Coenen V.A., Schlaepfer T.E., Allert N., Mädler B.: Diffusion tensor imaging and neuromodulation: DTI as key technology for deep brain stimulation. *International Review of Neurobiology*, vol. 107, pp. 207–234, 2012, <http://dx.doi.org/10.1016/B978-0-12-404706-8.00011-5>.
- [7] Frank E., Hall M., Holmes G., Kirkby R., Pfahringer B., Witten I.H., Trigg L.: Weka-A Machine Learning Workbench for Data Mining. In: O. Maimon, L. Rokach, eds., *Data Mining and Knowledge Discovery Handbook*, pp. 1269–1277, Springer US, 2010, [http://link.springer.com/chapter/10.1007/978-0-387-09823-4\\_66](http://link.springer.com/chapter/10.1007/978-0-387-09823-4_66).
- [8] Izhikevich E.: Which model to use for cortical spiking neurons? In: *IEEE Transactions on Neural Networks*, vol. 15(5), pp. 1063–1070, 2004, <http://dx.doi.org/10.1109/TNN.2004.832719>.
- [9] Izhikevich E.M.: Simple model of spiking neurons. In: *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 14(6), pp. 1569–1572, 2003, <http://dx.doi.org/10.1109/TNN.2003.820440>.
- [10] Jenkinson M., Bannister P., Brady M., Smith S.: Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, vol. 17(2), pp. 825–841, 2002, [http://www.psych.umn.edu/research/tricam/2\\_Jenkinson\\_NI\\_2002\\_MCFLIRT.pdf](http://www.psych.umn.edu/research/tricam/2_Jenkinson_NI_2002_MCFLIRT.pdf).
- [11] Krauth A., Blanc R., Poveda A., Jeanmonod D., Morel A., Székely G.: A mean three-dimensional atlas of the human thalamus: Generation from multiple histo-

- logical data. *NeuroImage*, vol. 49(3), pp. 2053–2062, 2010, <http://dx.doi.org/10.1016/j.neuroimage.2009.10.042>.
- [12] Latteri A., Arena P., Mazzone P.: Characterizing Deep Brain Stimulation effects in computationally efficient neural network models. *Nonlinear biomedical physics*, vol. 5(1), p. 2, 2011, <http://archive.biomedcentral.com/1753-4631/5/2/about>.
- [13] Lester H., Arridge S.R.: A survey of hierarchical non-linear medical image registration. In: *Pattern Recognition*, vol. 32(1), pp. 129–149, 1999, [http://dx.doi.org/10.1016/S0031-3203\(98\)00095-8](http://dx.doi.org/10.1016/S0031-3203(98)00095-8).
- [14] Maintz J.B.A., Viergever M.A.: An overview of medical image registration methods, 1998, [http://igitur-archive.library.uu.nl/math/2007-0118-200453/maintz\\_98\\_an\\_overview.pdf](http://igitur-archive.library.uu.nl/math/2007-0118-200453/maintz_98_an_overview.pdf).
- [15] Mayer A.R., Zimelman J.L., Watanabe Y., Rao S.M.: Somatotopic organization of the medial wall of the cerebral hemispheres: a 3 Tesla fMRI study. *Neuroreport*, vol. 12(17), pp. 3811–3814, 2001.
- [16] Rodriguez-Oroz M.C., Rodriguez M., Guridi J., Mewes K., Chockkman V., Vitek J., DeLong M.R., Obeso J.A.: The subthalamic nucleus in Parkinson’s disease: somatotopic organization and physiological characteristics. *Brain*, vol. 124(9), pp. 1777–1790, 2001, <http://dx.doi.org/10.1093/brain/124.9.1777>.
- [17] Zheng Z., Shemmassian S., Wijekoon C., Kim W., Bookheimer S.Y., Pouratian N.: DTI correlates of distinct cognitive impairments in Parkinson’s disease. *Human Brain Mapping*, vol. 35(4), pp. 1325–1333, 2014, <http://dx.doi.org/10.1002/hbm.22256>.

## Affiliations

### Artur Szymański

Polish-Japanese Academy of Information Technology, Warsaw, Poland,  
[artur.szymanski@pja.edu.pl](mailto:artur.szymanski@pja.edu.pl)

### Anna Kubis

AGH University of Science and Technology, Cracow, Poland, [anna.kubis@fis.agh.edu.pl](mailto:anna.kubis@fis.agh.edu.pl)

### Andrzej W. Przybyszewski

University of Massachusetts Medical School, Dept. Neurology, Worcester, USA,  
[andrzej.prybyszewski@umassmed.edu](mailto:andrzej.prybyszewski@umassmed.edu), Polish-Japanese Academy of Information Technology,  
Warsaw, Poland, [przy@pja.edu.pl](mailto:przy@pja.edu.pl)

**Received:** 19.01.2015

**Revised:** 23.03.2015

**Accepted:** 23.03.2015