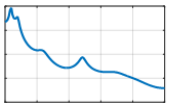

NEURO SPEECH

UNIVERSIDAD DE ANTIOQUIA

Technical Manual

Juan Camilo Vasquez-Correa
Juan Rafael Orozco-Arroyave
Jesus Francisco Vargas-Bonilla
Elmar Nöth

1 Introduction

Neuro-Speech is an open source software platform designed to perform speech analysis of people with neuro–degenerative disorders. Particularly patients with Parkinson's disease. The software is designed to be used by medical examiners such as speech therapists and neurologists, but it can also be used by patients to perform the analysis, and by general population interested in the analysis of pathological speech.

The software computes several measures to evaluate the communication capabilities of the patients and includes analyses of phonation, articulation, prosody, and intelligibility. The software calculates also specific bio-markers related to the dysarthria levels of the patients and perform a prediction of the Movement disorder society–Parkinson's disease rating scale, part III (MDS-UPDRS-III), which is a general evaluation of the motor capabilities of the patients.

All of the results obtained are compared to reference patterns obtained with a group of 50 healthy control speakers.

Finally a medical report can be generated, which describes the different speech deficits of the patients, and how the measures are deviated respect to those computed with information from healthy controls

Juan Camilo Vasquez-Correa
Faculty of Engineering, University of Antioquia UdeA, Medellín, Colombia. and Pattern Recognition Lab, Friedrich–Alexander Universität, Erlangen–Nuremberg, Germany, e-mail: jcamilo.vasquez@udea.edu.co

Juan Rafael Orozco-Arroyave
Faculty of Engineering, University of Antioquia UdeA, Medellín, Colombia. and Pattern Recognition Lab, Friedrich–Alexander Universität, Erlangen–Nuremberg, Germany, e-mail: rafael.orozco@udea.edu.co

Jesus Francisco Vargas-Bonilla
Faculty of Engineering, University of Antioquia UdeA, Medellín, Colombia. e-mail: jesus.vargas@udea.edu.co

Elmar Nöth
Pattern Recognition Lab, Friedrich–Alexander Universität, Erlangen–Nuremberg, Germany, e-mail: elmar.noeth@fau.de

1.1 Requeriments

Neuro–Speech is a software platform designed in C++, which runs python scripts in background for the speech analysis. The software uses some third party software that can be freely downloaded and installed for the correct operation of Neuro-Speech. The list of the third party software that must be installed previous to Neuro–Speech is as follows:

1. Ananconda: Python environment. It can be installed from https://www.continuum.io/downloads
2. Praat: software for speech analysis. Available at http://www.fon.hum.uva.nl/praat/
3. ffmpeg: a solution to record, convert and stream audio and video. Available at http://ffmpeg.org/download.html

To execute Neuro–Speech, please go to the folder *Release*, and then click in the icon of PDTool.exe. Then the main window shown in Figure 1.2 is displayed for the analysis.

1.2 Main Window

Figure 1.2 shows the main window of Neuro-Speech. It contains buttons to record and play the speech signals. The recording can be visualized also in this window. The main window contains also six different buttons to perform each one of the speech analysis: (1) phonation, (2) articulation, (3) prosody, (4) diadochokinetic (DDK), (5) intelligibility, (6) PD evaluation, and the last button to create the medical report. Table 1 details the description of each part of the main window.

2 Speech recording

The speech utterances are recorded by default with a sampling frequency of 16 kHz, and 16–bit resolution. After finishing the recording, the audio file is saved with the name of the task in a directory created with the name of the patient and the current date in the *data* directory.

The speech tasks performed by the patients are the same than those used in PC-GITA database [3], and include the phonation of sustained vowels, the phonation of modulated vowels, a diadochokinetic (DDK) evaluation, the pronunciation of isolated words and sentences, the pronunciation of a read text formed with 36 words, and finally the pronunciation of a monologue about the daily activities of the user. A total of 25 different tasks are recorded, which can be detailed in Table 2.

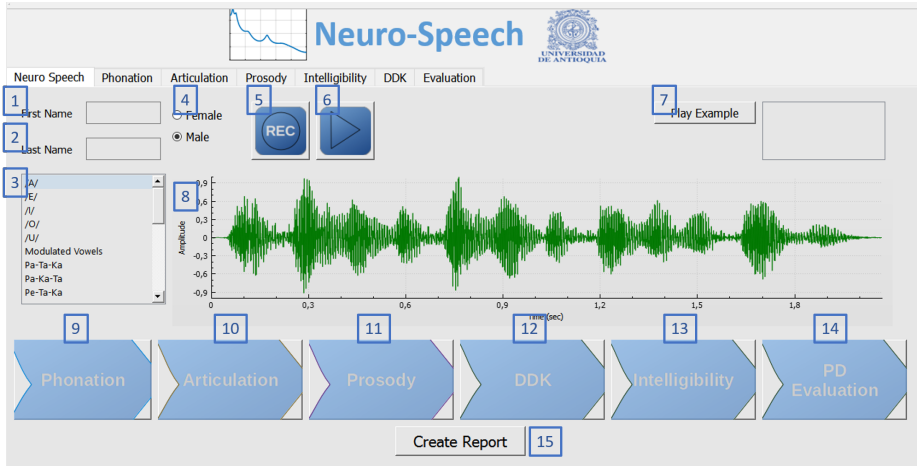

Fig. 1 Main Window of Neuro–Speech

Table 1 Description of each part of the main window

Button	Description
1	Input of the first name of the patient
2	Input of the last name of the patient
3	List of tasks for the speech recordings (one of them must be selected to record)
4	Select the gender of the speaker
5	Record button
6	Play button
7	Play an example of the task to record
8	Field to visualize the speech signal after recording
9	Perform the phonatory analysis
10	Perform the articulatory analysis
11	Perform the prosody analysis
12	Perform the DDK analysis
13	Perform the intelligibility analysis
14	Perform the dysarthria and PD evaluation
15	Generate the medical report

3 Phonation Analysis

Phonation is related with the capability of the speaker to make the vocal folds vibrate, and it has been analyzed in terms of features related to perturbation measures such as jitter, shimmer, the amplitude perturbation quotient, the pitch perturbation quotient, and non-linear dynamics measures.

Table 2 List of tasks for speech recording

Number	Task	Description
		Sustained Vowels
1	/A/	Sustained phonation of vowel A
2	/E/	Sustained phonation of vowel E
3	/I/	Sustained phonation of vowel I
4	/O/	Sustained phonation of vowel O
5	/U/	Sustained phonation of vowel U
6	Modulated Vowels	The five vowels pronounced with Kaiser effect
		Diadochokinetic evaluation
7	Pa-Ta-Ka	rapid repetitions of syllables pa-ta-ka
8	Pa-Ka-Ta	rapid repetitions of syllables pa-ka-ta
9	Pe-Ta-Ka	rapid repetitions of syllables pe-ta-ka
10	Pa	rapid repetitions of syllable pa
11	Ta	rapid repetitions of syllable ta
12	Ka	rapid repetitions of syllable ka
		Isolated words
13	Words	25 isolated words
		Continuous Speech
14	Sentence 1	”Mi casa tiene tres cuartos”
15	Sentence 2	”Omar, que vive cerca trajo miel”
16	Sentence 3	”Laura sube al tren que pasa”
17	Sentence 4	”Los libros nuevos no caben en la mesa de la oficina”
18	Sentence 5	”Rosita Nio, que pinta bien, donó sus cuadros ayer”
19	Sentence 6	”Luisa rey compra el colchón duro que tanto le gusta”
20	Sentence 7	”Viste las noticias, yo vi ganar la medalla de plata en pesas, ese muchacho tiene mucha fuerza”
21	Sentence 8	”Juan se rompió una pierna cuando iba en la moto”
22	Sentence 9	”Estoy muy triste, ayer vi morir a un amigo”
23	Sentence 10	”Estoy muy preocupado, cada vez me es ms dificil hablar”
24	Read Text	read text with 36 words *
25	Monologue	The patient says What he/she does in a normal day

* Read text: Ayer fu al médico. Qué le pasa? Me preguntó
Yo le dije: Ay doctor. Donde pongo el dedo me duele.
Tiene la uña rota? Sí. Pues ya sabemos que es.
Deje su cheque a la salida.

The computed measures include the jitter (temporal perturbation of the fundamental frequency), the shimmer (temporal perturbation of the amplitude of the signal), the APQ and PPQ (long term perturbation measures from amplitude and pitch), the degree of unvoiced (percentage of non–periodic utterance), and the variability of the fundamental frequency measured in semitones [1, 4]. Each one of them es explained in the following subsections.

3.1 Jitter and Shimmer

Temporal perturbations in the frequency and amplitude of the speech utterance are defined as jitter and shimmer, respectively. Jitter is computed with Equation 1, where N is the number of frames of the speech utterance, $\mathrm{M_f}$ is the maximum of the fundamental frequency, and $\mathrm{F_0(k)}$ corresponds to the fundamental frequency computed on the *k*–th frame.

$$\mathrm{Jitter}(\%) = \frac{100}{\mathrm{N} \cdot \mathrm{M_f}} \sum_{k=1}^{\mathrm{N}} |\mathrm{F_0(k)} - \mathrm{M_f}| \tag{1}$$

Shimmer is computed using Equation 2, where $\mathrm{M_a}$ is the maximum amplitude of the signal, and $\mathrm{A(k)}$ corresponds to the amplitude on the *k*–th frame.

$$\mathrm{Shimmer}(\%) = \frac{100}{\mathrm{N} \cdot \mathrm{M_a}} \sum_{k=1}^{\mathrm{N}} |\mathrm{A(k)} - \mathrm{M_a}| \tag{2}$$

3.2 Amplitude and Pitch Perturbation Quotients (APQ and PPQ)

The APQ measures the long–term variability of the peak-to-peak amplitude and pitch of the speech signal with a smoothing factor of 11 periods. APQ is computed as the absolute average difference between the amplitude of a frame and the average of the amplitudes of its neighbors, divided by the average amplitude.

Similar to APQ, the PPQ measures the long–term variability of the fundamental frequency, with a smoothing factor of five periods. PPQ is computed as the absolute average difference between the frequency of each frame and the average of its neighbors, divided by the average frequency. Both perturbation quotients (PQ) are computed using Equation 3, where $\mathrm{L} = \mathrm{N} - (\mathrm{k} - 1)$, S is the pitch period sequence (PPS) when computing the PPQ or the pitch amplitude sequence (PAS) when computing the APQ. N is the length of PPS or PAS, k is the length of the moving average (11 for PAQ or 5 for PPQ), and $\mathrm{n} = (\mathrm{k} - 1)/2$.

$$\mathrm{PQ} = \frac{1}{L} \sum_{i=1}^{L} \frac{\left| \frac{1}{k} \sum_{j=1}^{k} \mathrm{S}(i+j-1) - \mathrm{S}(i+n) \right|}{\left| \frac{1}{M} \sum_{j=1}^{M} \mathrm{S}(i) \right|} \tag{3}$$

3.3 degree of Unvoiced

This feature is computed as the percentage of the utterance of a sustained vowel that is detected as unvoiced using a voiced–unvoiced segmentation implemented

in Praat [2]. The utterance is segmented into voiced and unvoiced frames, and the degree of unvoiced is computed using Equation 4.

$$\text{degree U}\,(\%) = \frac{\text{Total duration of unvoiced frames}}{\text{Total duration of the utterance}} \tag{4}$$

3.4 Implementation

The phonation analysis in NeuroSpeech is performed with the python script called *phonVowels.py*, which is contained in the folder *phonVowels*. The syntax to perform the analysis is as follows

```
python phonVowels.py <file_audio> <filef0.txt> <file_features.txt>
<path_base>
```

Where <file_audio> corresponds to the audio file to be analyzed, <filef0.txt> is a result file which will contain the contour of the fundamental frequency (computed using Praat), <file_features.txt> is a file which will contain the features described previously (jitter, shimmer, APQ, PPQ, degree of unvoiced and the variablity of the fundamental frequency measured in semitones), and <path_base> corresponds to the path where the script is stored. The script creates also a plot of the contour of the fundamental frequency, and a radar plot to compare the features computed for the speaker with those computed from a database formed with 50 healthy speakers. Individual reference are computed for female and male speakers. Figure 3.4 show the figures generated for the contour of the fundamental frequency for a sustained phonation of vowel /A/ for a healthy speaker (left) and a PD patient (right). The speech signal of the PD patient is more variable in amplitude than the utterance of the healthy speaker, which causes higher values of shimmer and APQ for the patient. Note also the difference in the contour of the fundamental frequency, which is more stable in the healthy speaker.

Figure 3.4 shows the radar plot of the perturbation measures obtained with the phonation script also for a healthy speaker (left) and a PD patient (right). The green polygon shows the values of the features obtained for the patient, and the blue polygon is computed with the average values of the features from healthy speakers. When the green figure is completely inside of the blue one the features of the patient are in the same range (normal) than the reference (Figure 3.4 left). If there is a peak in one of the corners of the green polygon, there is a difference in the feature associated to such corner with the value obtained for the reference, This behaviour is observed in Figure 3.4. Note from the figures that for the healthy speaker, the green plot is completely contained by the blue figure, indicating that the speaker do not exhibit problems regarding the phonation, while for the PD patient, all of the vertexes are outised the reference.

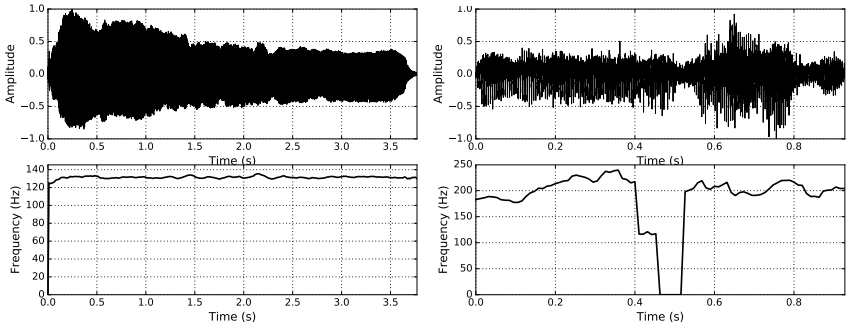

Fig. 2 Speech signal and fundamental frequency of a sustained phonation of vowel A for a healthy speaker (left), and for a PD patient (right)

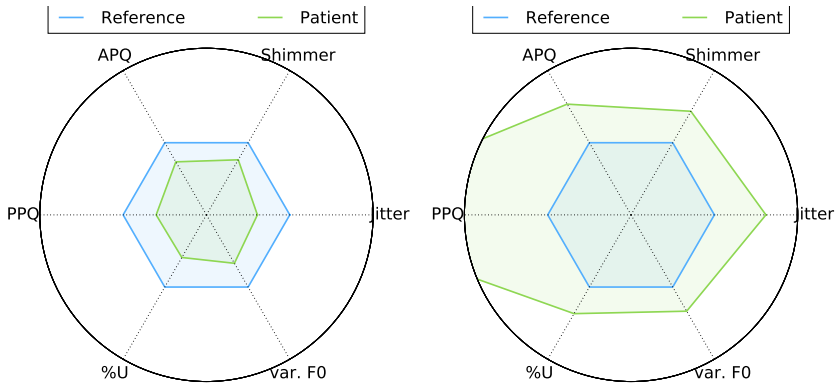

Fig. 3 Radar figures for a healthy speaker (left), and for a PD patient (right)

4 Articulation Analysis

Articulation is related with the modification of the position, stress, and shape of several limbs and muscles to produce the speech, and it has been described with features such as the vowel space area, the vowel articulation index, the formant centralization ratio, and the energy in the voiced/unvoiced transition. In Neuro–Speech, the articulation analysis computes measures from sustained vowels and continuous speech.

4.1 Articulation in sustained vowels

The articulation analysis computes measures such as the first two formant frequencies (F1 and F2)with the aim of model the movement of the vocal tract when the speaker pronounce the five Spanish vowels. Measures such as the vowel space area (VSA), vocal pentagon area (VPA), and formant centralization ratio (FCR) are also computed to evaluate the articulation capabilities of the speakers.

The VSA is a measure used to model possible reduction in the articulatory capability of the speaker. The reduction is observed as a compression of the area of the vocal triangle. VSA is computed by calculating F1 and F2 of the vowels /a/, /i/, and /u/. Then the VSA is estimated using Equation 5.

$$\text{VSA} = \frac{|F_{1i}(F_{2a} - F_{2u}) + F_{1a}(F_{2u} - F_{2i}) + F_{1u}(F_{2i} - F_{2a})|}{2} \tag{5}$$

When the first two formants of the five Spanish vowels are considered as the vertexes of a polygon, the vocal pentagon is formed, and the area of such pentagon is called VPA. This measure quantifies the articulatory capabilities of the speakers when they pronounce the five Spanish vowels. VPA was introduced in [5] to evaluate articulatory deficits in PD patients. This measure is calculated using Equation 6, where $\text{ps}_1 = F_{1a}F_{2o} - F_{1o}F_{2a}$, $\text{ps}_2 = F_{1o}F_{2u} - F_{1u}F_{2o}$, $\text{ps}_3 = F_{1u}F_{2i} - F_{1i}F_{2u}$, $\text{ps}_4 = F_{1i}F_{2e} - F_{1e}F_{2i}$, and $\text{ps}_5 = F_{1e}F_{2a} - F_{1a}F_{2e}$.

$$\text{VPA} = \frac{|\text{ps}_1 + \text{ps}_2 + \text{ps}_3 + \text{ps}_4 + \text{ps}_5|}{2} \tag{6}$$

Finally, the FCR is a measure introduced in [8] to analyze changes in the vocal formants with a reduced inter–speaker variability. FCR is computed with Equation 7.

$$\text{FCR} = \frac{F_{2u} + F_{2a} + F_{1i} + F_{1u}}{F_{2i} + F_{1a}} \tag{7}$$

4.2 Articulation in continuous speech

The articulation capabilities of the patients in continuous speech is evaluated considering the energy content both in the transition from unvoiced to voiced segments (onset), and in the transition from unvoiced to voiced segments (offset). The measures were introduced in [6, 7] with the hypothesis that PD patients have difficulty to start and stop the vocal fold vibration, and such difficulty can be observed on the speech signal by computing the energy content of the transitions between voiced and unvoiced sounds.

To compute the measures, first the fundamental frequency of the speech signal is estimated in Praat [2] to detect the voiced and unvoiced segments. Then the transitions from voiced to unvoiced (offset), and from unvoiced to voiced (onset) are

detected and 40 ms of the signal are taken to the left and to the right of each border. Finally, the spectrum of the transitions is distributed in 22 critical bands following the Bark scale, and the energy content is computed on each band.

4.3 Implementation

The articulation analysis of sustained vowels is performed with the python script called *artVowels.py*, which is contained in the folder *artVowels*. The syntax to perform the analysis is as follows.

```
python artVowels.py <file_audioA> <file_audioE> <file_audioI>
<file_audioO> <file_audioU> <file_resultsA> <file_resultsE>
<file_resultsI> <file_resultsO> <file_resultsU> <file_features>
<path_base>
```

Where <file_audioX> corresponds to the audio file of vowel X, $X \in \{A, E, I, O, U\}$, <file_resultsX.txt> is a result file which will contain the contour of the formant frequencies of vowel X (computed using Praat), <file_features.txt> is a file which will contain the features described previously (the average value of the formants of the five Spanish vowels, VSA, VPA, and FCR), and <path_base> corresponds to the path where the script is contained.

The script generates also figures of the vocal triangle and the vocal pentagon, and it performs a comparison with the polygons obtained with speech recordings of the 50 healthy speakers. Figure 4.3 shows the vocal triangle and the vocal pentagon for a healthy speaker (up), and for a PD patient (bottom). Note the reduction in the area of the triangle and the pentagon for the case of the PD patient (bottom), relative to the areas obtained for the healthy speaker. That fact indicates the reduction of the articulatory capabilities of the patient. Note also that the formant frequencies for each vowel are more spread out for the PD patient than for the healthy speaker where the formants are more centered, indicating that the patient may has lost the control capability of the limbs and muscles to produce the speech such as the tongue and the velum.

The articulation analysis in continuous speech is performed with the python script called *artCont.py*, which is contained in the folder with the same name. The syntax to perform the analysis is as follows.

```
python artCont.py <file_audio> <file_features> <path_base>
```

Where <file_audio> corresponds to the audio file to be analyzed, <file_features.txt> is a file which will contain the measures of the energy content in the transitions

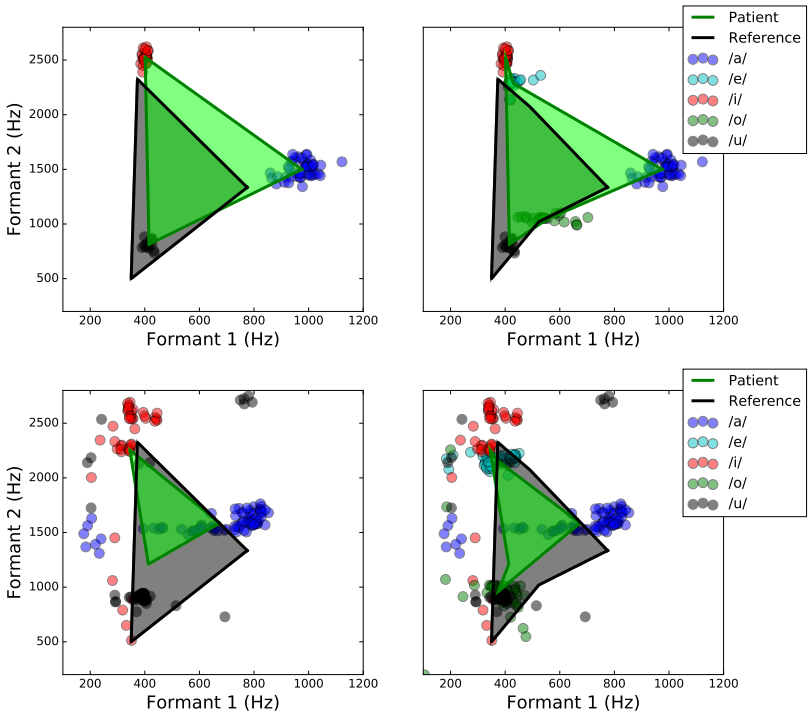

Fig. 4 Vocal triangle and pentagon for a healthy speaker (up) and for a PD patient (bottom)

between voiced to unvoiced segments, and <path_base> corresponds to the path where the script is contained.

The script creates also a radar figure to compare the features computed for the speaker with those computed from the reference. Figure 4.3 shows the radar plot of the articulation measures in continuous speech for a healthy speaker (left) and a PD patient (right). The green polygon shows the values of the features obtained for the patient, and the blue one corresponds to the reference. When the green figure is completely contained by the blue one, the features of the speaker are in the same range (normal) than the healthy speakers. If there is some vertexes that are outside of the green polygon, there is a difference in the feature associated to such vertex with the value obtained for the healthy speakers.

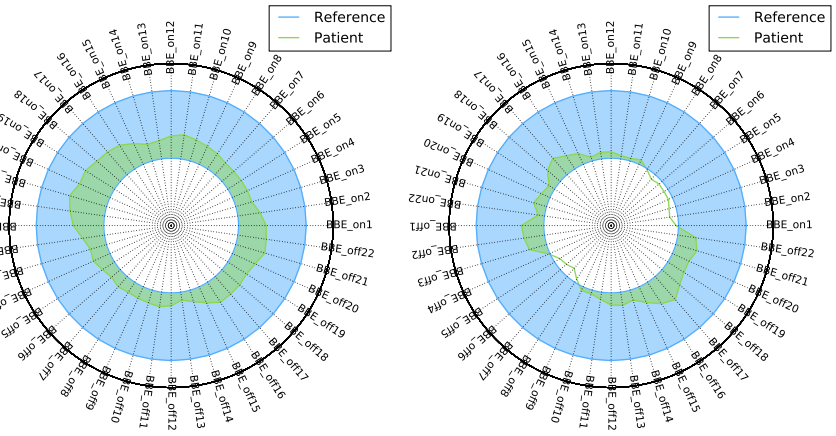

Fig. 5 Radar figures for a healthy speaker (left), and for a PD patient (right)

5 Prosody Analysis

Prosody reflects variation of loudness, pitch, and timing to produce natural speech and it is commonly evaluated with features related to the fundamental frequency, the energy contour, and the duration. The prosody analysis is performed with the read text, or the monologue.

5.1 Features related to fundamental frequency

With the aim of modeling intonation patterns of the PD patients, the contour of the fundamental frequency is computed, and statistic functionals are computed: the average, standard deviation, and maximum values are calculated to evaluate the monotonicity of the patient, and the maximum tone that can be reached by the speaker.

5.2 Features related to energy

Similar to the fundamental frequency, the energy contour of the speech utterance is computed, and statistical functionals are calculated from such contour. The average energy, the standard deviation of the energy, and the maximum value are calculated.

5.3 Features related to duration

Six different duration measures are computed from the speech signal, including among others, the average duration of silences, and the voiced rate. The complete list of the duration–based features is as follows.

- The voiced rate: how many voiced segments appear per second; it is a measure related to the speech velocity.
- The average duration of voiced segments.
- The standard deviation of the duration of voiced segments.
- The silence rate: how many silence segments appear per second; it is a measure also related to the speech velocity.
- The average duration of silence segments.
- The standard deviation of the duration of the silence segments.

5.4 Implementation

The prosody analysis is performed with the python script called *prosody.py*, which is contained in the folder with the same name. The syntax to perform the analysis is as follows.

```
python prosody.py <file_audio> <filef0.txt> <fileEn.txt>
<file_features.txt> <path_base>
```

Where <file_audio> corresponds to the audio file to be analyzed, <filef0.txt> is a result file which will contain the contour of the fundamental frequency, <fileEn.txt> will contain the energy contour, <file_features.txt> is the file that will contain the features described previously, and <path_base> is the path where the script is contained. The script creates also a figure of the contours of the fundamental frequency and energy, and a radar plot with the prosody features, just as the created for phonation and articulation analyses.

Figure 6 shows the contours of the fundamental frequency and energy for a healthy speaker (left) and for a PD patient (right) for a read text.

Figure 7 contains the radar figures for the prosody analysis for the healthy speaker (left), and for the PD patient (PD). The figure for the healthy speaker is more regular than the obtained for the PD patient. Note the peaks in the figure of the PD patient that are outside the normal range.

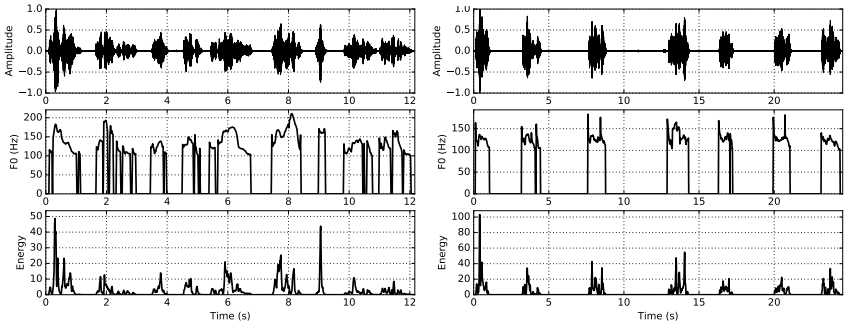

Fig. 6 Contours of the fundamental frequency and energy for a healthy speaker (left), and for a PD patient (right), for a read text

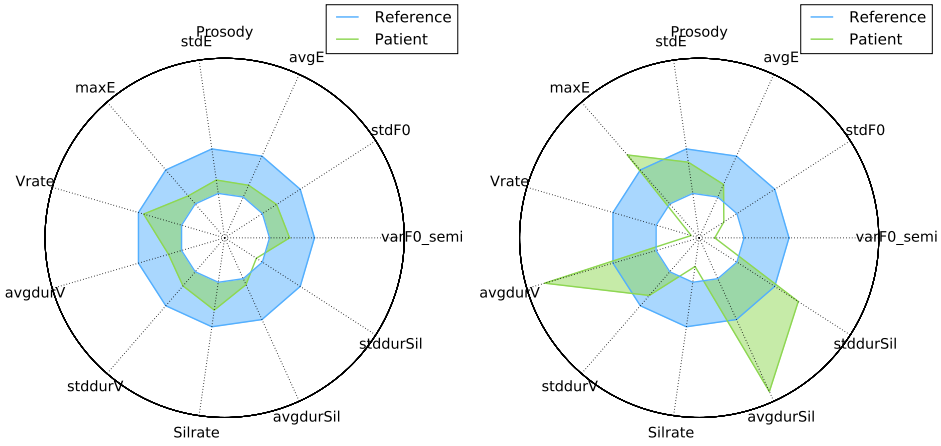

Fig. 7 Radar figures for a healthy speaker (left), and for a PD patient (right)

6 DDK Analysis

DDK analysis consists in the rapid repetition of syllables such as /pa-ta-ka/, /pa-ka-ta/, /pe-ta-ka/, /pa/, /ta/, and /ka/ to analyze the capability of the speaker to move articulators such as the velum, the jaw, and the tongue.

6.1 Features related to fundamental frequency

With the aim of modeling intonation patterns of the PD patients, the contour of the fundamental frequency is computed, and statistic functionals are computed: the

average, the variance measured both in Hz and semitones, and the maximum values are calculated to evaluate the monotonicity of the patient, and the maximum tone that can be reached by the speaker.

6.2 Features related to energy

Similar to the fundamental frequency, the energy contour of the DDK utterance is computed, and statistical functionals are calculated from such contour. The average energy, the standard deviation of the energy, and the maximum value are calculated.

6.3 Features related to duration

Six different duration measures are computed from the speech signal, including among others, the DDK rate, and the DDK regularity. The complete list of features computed from the DDK utterances is as follows

- The variability of the fundamental frequency measured in semitones.
- The variability of the fundamental frequency measured in Hz.
- the average energy of the DDK task.
- The variablity of the energy along the utterance.
- The maximum value of the energy.
- The DDK rate: how many syllables are uttered by second.
- The DDK regularity, which is measured as the variability of the duration of each syllable.
- The average duration of the syllables.
- The pause rate: how many silence segments appear per second.
- The average duration of pauses.
- The regularity of pauses, which is measured as the variability of the duration of the pauses.

6.4 Implementation

The DDK analysis is performed with the python script called *DDK.py*, which is contained in the folder with the same name. The syntax to perform the analysis is as follows.

```
python DDK.py <file_audio> <filef0.txt> <fileEn.txt> <file_features.txt>
<path_base>
```

Where <file_audio> corresponds to the audio file to be analyzed, <filef0.txt> is a result file which will contain the contour of the fundamental frequency, <fileEn.txt> will contain the energy contour, <file_features.txt> is the file that will contain the features described previously, and <path_base> is the path where the script is contained. The script creates also a figure of the contours of the fundamental frequency and energy, and a radar plot with the prosody features, just as the created for phonation and articulation analyses.

Figure 8 shows the contours of the fundamental frequency and energy for a healthy speaker (left) and for a PD patient (right) for a read text.

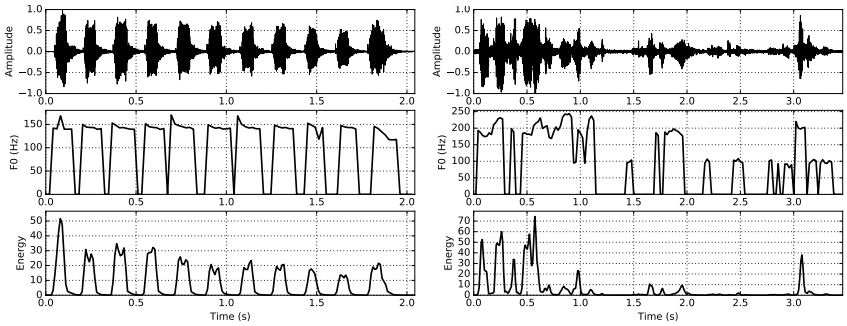

Fig. 8 Contours of the fundamental frequency and energy for a healthy speaker (left), and for a PD patient (right), for the rapid repetition of the syllables pa-ta-ka

Figure 9 contains the radar figures for the DDK analysis for the healthy speaker (left), and for the PD patient (PD).

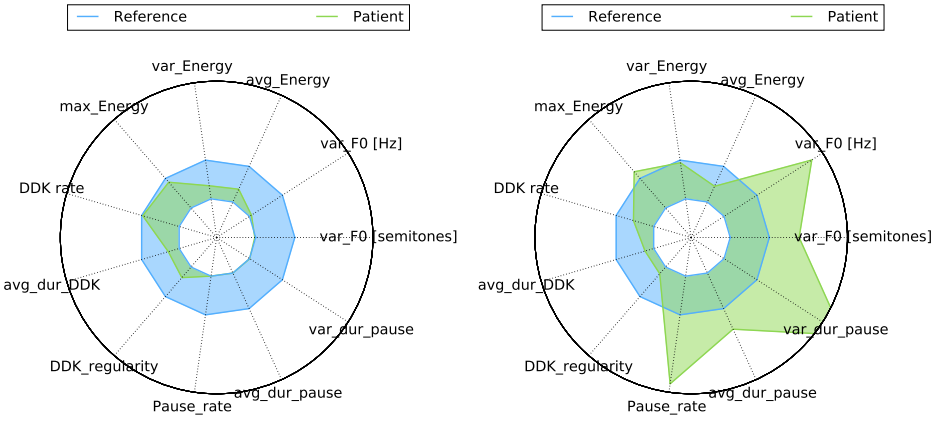

Fig. 9 Radar figures for a healthy speaker (left), and for a PD patient (right)

7 Intelligibility Analysis

Intelligibility is related to the capability of a person to be understood by other person or by a system. Intelligibility is also deteriorated in patients with neurological disorders causing loss of communication abilities and social isolation specially at advanced stages of the disease.

The intelligibility analysis is performed with the ten read sentences and with the read text, using the automatic speech recognizer (ASR) provided by Google Inc. (This analysis requires Internet connection). Two measures are calculated for the intelligibility analysis: the word accuracy (WA), and a similitude measure based on dynamic time warping (sDTW). The measures were introduced in [7, 9] to model the intelligibility deficits of PD patients.

7.1 Word accuracy

The WA has been established as a marker to analyze the performance of ASR systems and intelligibility of persons. The WA is defined as the number of words correctly recognized by the ASR system relative to the total of words in the original string. The WA is computed following Equation 8.

$$\text{WA} = \frac{\#\,\text{words correctly recognized}}{\#\,\text{of total words}} \tag{8}$$

7.2 Similitude based on Dynamic Time Warping

The DTW is a technique to analyze similarities between two time–series when both sequences may have differences in time, and number of samples, performing a time–alignment between the sequences. The DTW distance is computed between the predicted string i.e., the complete sentence using the ASR system and the original sentence read by the patients. The distance is computed over the all text, at grapheme level.

Then the distance is transformed to a similarity score using Equation 9. If the sequences are the same, the *DTW_distance* will be zero, and the similarity will be 1, and if the strings are very different, the *DTW_distance* will be high, and the similarity will be close to zero.

$$\text{sDTW} = \frac{1}{1 + \text{DTW\_distance}} \tag{9}$$

7.3 Implementation

The intelligibility analysis is performed with the python script called *intelligibility.py*, which is contained in the folder with the same name. The syntax to perform the analysis is as follows.

```
python intelligibility.py <file_audio> <file_txt.txt> <pred_txt.txt>
<file_features.txt>
```

Where <file_audio> is the audio file to be analyzed, <file_txt.txt> is the transcription of the audio file, <pred_txt.txt> will contain the predicted string by the ASR, and <file_features.txt> is the file that will contain the intelligibility measures. A radar figure is also created with the intelligibility features computed from different utterances, indicating the intelligibility capability of the speaker. Figure 7.3 contains the radar figures for the intelligibility analysis for the healthy speaker (left), and for the PD patient (PD). Note the high reduction in the intelligibility of the PD patient, compared to the figure obtained for the healthy speaker.

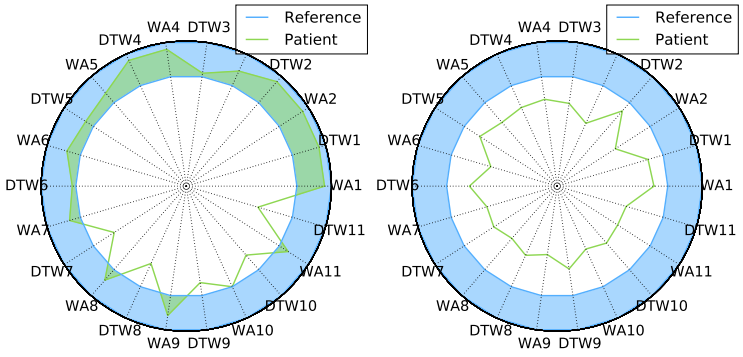

Fig. 10 Radar figures for a healthy speaker (left), and for a PD patient (right)

8 Dysarthria evaluation

Additionally to the UPDRS score, we introduce a modified version of the Frenchay Dysarthric Assessment (FDA) score. The main aim was to evaluate only the speech impairments that the patients develop. The original version of

the FDA needs the patient to be with the examiner. We introduced a modified version that considers only the speech recordings and evaluates 13 items including among others the movements of the lips, larinx, palate and tongue, the respiration, and the intelligibility. The evaluation of each item ranges from 0 to 4, for a total range from 0 to 52 (0 normal, and 52 completely dysarthric)

The dysarthria evaluation includes the assessment of the respiration capability, the lips movement, the palate movement, the larinx movement, the tongue velocity, the intelligibility, and the total dysarthria score. The prediction of the dysarthric subscores is performed using multi-class support vector machines (SVMs), which are trained with the articulation–based features described previously, extracted from utterances from 100 speakers (50 PD patients, and 50 healthy controls). The labeling process for training the SVMs was performed by three expert phoniatricians, who agreed in the first ten evaluations, and then performed the evaluation of the other recordings. The inter-rater reliability among the labelers is 0,75.

The software also performs a prediction of the MDS-UPDRS-III score of the patients. using a support vector regressor trained also with the articulation–based features extracted from 50 PD patients evaluated by a neurologist expert.

8.1 Implementation

The prediction of the m-FDA and UPDRS-III scores is performed with the python script called *predictPD.py*, which is contained in the folder called */evaluation/*. The syntax to perform the prediction is as follows.

```
Python predictPD.py <path_base> <file_audio>
```

Where <file_audio> is the audio file to be analyzed and <path_base> is the folder where the script is contained. The script will generate a file called <pred.txt> in the <path_base>, which will contain the following values: (1) the prediction of the total m-FDA score, (2) the item of the m-FDA related to respiration, (3) the item of the m-FDA related to the lips movement, (4) the item of the m-FDA related to the palate movement, (5) the item related to the larynx movement, (6) the item related to the tongue, (7) the item related to intelligibility, and (8) the prediction of the UPDRS-III score.

If the user wants to re-train the models to predict the UPDRS-III and the m-FDA scores using other features or other speech tasks, the script called <TrainSVRNeuroSpeech.py> should be run, in the following way.

```
Python TrainSVRNeuroSpeech.py <file_matrix.txt> <file_labels.txt>
<file_scaler.obj> <fileSVR.obj>
```

Where <file_matrix.txt> is a txt file with the feature matrix, <file_labels.txt> corresponds also to a txt file with the labels of the scores (in the folder there are files for the UPDRS-III (labelsUPDRS.txt) and mFDA scores (labelsmFDA.txt)). <file_scaler.obj> is an output file which will contain an object for the standardization of the feature matrix (mean and variance), and <fileSVR.obj> will contain the trained SVR. The <file_scaler.obj> should be called scalerUPDRS.obj or scalermFDA.obj for the prediction of the UPDRS or m-FDA scores, respectively, and the trained SVR file should be called SVRtrainedUPDRS.obj for the prediction of the UPDRS-III and SVRtrainedmFDA.obj for the mFDA scores.

To re-train the multi-class SVMs to predict the sub-scales of the mFDA the user should use the following script.

```
Python TrainSVMNeuroSpeech.py <file_matrix.txt> <sub-scale>
<file_labels.txt> <file_scaler.obj> <fileSVM.obj>
```

TrainSVMNeuroSpeech.py runs in a similar way than TrainSVRNeuroSpeech.py, including the parameter <sub-scale>, which refers to the sub-scale to be trained: (r for respiration, l for lips, p for palate, x for larynx, t for tongue, and i for intelligibility)

References

1. Arias-Vergara, T., Vasquez-Correa, J.C., Orozco-Arroyave, J.R.: Parkinson's disease and aging: analysis of their effect in phonation and articulation of speech. Submitted to Cognitive Computation (2017)
2. Boersma, P., et al.: Praat, a system for doing phonetics by computer. Glot international **5**(9/10), 341–345 (2002)
3. Orozco-Arroyave, J.R., Arias-Londoño, J.D., Vargas-Bonilla, J.F., Gonzalez-Rátiva, M.C., Nöth, E.: New Spanish speech corpus database for the analysis of people suffering from Parkinson's disease. In: Language Resources and Evaluation Conference, (LREC), pp. 342–347 (2014)
4. Orozco-Arroyave, J.R., Belalcazar-Bolaños, E.A., Arias-Londoño, J.D., Vargas-Bonilla, J.F., Skodda, S., Rusz, J., Daqrouq, K., Hönig, F., Nöth, E.: Characterization methods for the detection of multiple voice disorders: Neurological, functional, and laryngeal diseases. IEEE Journal of Biomedical and Health Informatics **19**(6), 1820–1828 (2015)
5. Orozco-Arroyave, J.R., Belalcázar-Bolaños, E.A., Arias-Londoño, J.D., Vargas-Bonilla, J.F., Haderlein, T., Nöth, E.: Phonation and articulation analysis of spanish vowels for automatic detection of parkinsons disease. In: International Conference on Text, Speech, and Dialogue, pp. 374–381. Springer (2014)
6. Orozco-Arroyave, J.R., Hönig, F., Arias-Londono, J.D., Vargas-Bonilla, J.F., Skodda, S., Rusz, J., Nöth, E.: Voiced/unvoiced transitions in speech as a potential bio-marker to detect parkin-

sons disease. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
7. Orozco-Arroyave, J.R., Vásquez-Correa, J.C., Hönig, F., Arias-Londoño, J.D., Vargas-Bonilla, J.F., Skodda, S., Rusz, J., Nöth, E.: Towards an automatic monitoring of the neurological state of the Parkinson's patients from speech. In: 41st International Conference on Acoustic, Speech, and Signal Processing (ICASSP) (2016)
8. Sapir, S., Ramig, L.O., Spielman, J.L., Fox, C.: Formant centralization ratio: a proposal for a new acoustic measure of dysarthric speech. Journal of Speech, Language, and Hearing Research **53**(1), 114–125 (2010)
9. Vasquez-Correa, J.C., Orozco-Arroyave, J.R., Nöth, E.: Word accuracy and dynamic time warping to assess intelligibility deficits in patients with Parkinson's disease. In: XXI Symposium on Image, Signal Processing and Artificial Vision (STSIVA) (2016)