

KIJK: A PRAAT PLUGIN TO VISUALISE ARTICULATORY TRAJECTORIES

Carolina Lins Machado¹, Lei He^{1 2}

¹ Dept. Computational Linguistics, University of Zurich, Zurich, Switzerland

² Dept. Phoniatrics and Speech Pathology, Clinic for Otorhinolaryngology, Head and Neck Surgery, University Hospital Zurich (USZ), Zurich, Switzerland
cmachado@ifi.uzh.ch, lei.he@uzh.ch

ABSTRACT

Electromagnetic articulography (EMA) is one of the most frequently employed methods for the collection of speech kinematic data. Likewise, Praat is one of the most used software for processing and analysing acoustic data, due to its relatively straightforward operation and portability across different operational systems. However, Praat does not possess an automatic procedure to visualise kinematic data collected with EMA devices. Therefore, in this paper we introduce *Kijk*, a Praat plugin for the visualisation of kinematic trajectories. Because it is completely implemented in Praat, *Kijk* offers the user a quick and simple approach to the combined visualisation and annotation of articulatory and acoustic data.

Keywords: Praat, plugin, EMA, kinematic trajectory

1. INTRODUCTION

Speech production is a process involving the movement of articulators – such as the lips, tongue, velum and jaw – resulting in an acoustic outcome encoding linguistic, paralinguistic and extralinguistic information, all vital to communication between individuals. The process of producing speech is broadly studied, for instance, by speech scientists, linguists and phoneticians at the articulatory and acoustic levels. Acoustic analysis entails investigating the speech wave itself, where characteristics of a language can be revealed, for instance, in the spectral and temporal features of the acoustic signal. Changes in the spectral features in this signal are achieved by the movements of the articulators. Thus, articulatory analysis offers an understanding of how, to some extent, these acoustic features are modulated.

Articulatory analysis can be performed with a variety of methods. Among a diverse array of approaches for the collection of articulatory data, the most frequently employed method is electromagnetic articulography (or EMA) [2], which can be used to study oral movements. By attaching sensors to the tongue, jaw and lips, the movement of multiple articulators can be recorded simultaneously (for a further description of how EMA works please see

[2]). The output of this method is kinematic data related to the displacement of articulators. The analysis of speech kinematic data is commonly performed using the native EMA device tools, specialised softwares [3, 11], or toolkits [4, 5, 9, 10] for Matlab [6] and R [8], to name a few.

Acoustic data on the other hand is widely processed in Praat [7], an open-source freeware for the analysis of acoustic data. It's popularity is without a doubt due to its portability, and because it requires a shallow learning curve from the user, being attractive to students and researchers alike. Regarding acoustic analyses, Praat can be used for visualising, annotating (through TextGrids) and measuring speech; offering the user a great number of features for these purposes. Moreover, the software allows the user to synthesise and manipulate speech, to perform some statistical analyses and to output results also in graphical form. Therefore, Praat offers the potential to be used for the visualisation and annotation of EMA trajectories. For instance, by examining kinematic trajectories, acoustic recordings and annotated TextGrids simultaneously it becomes easier to study and compare EMA data along with the acoustic output of specific speech sounds. However, this software does not yet possess an automatic method to handle EMA data. For that reason, we attempt to offer a solution with *Kijk* [keik] (meaning “look” in Dutch), a plugin completely implemented in Praat to automatically process EMA data for the visualisation of kinematic trajectories.

The plugin was developed under Praat version 6.2.13 and assumes that the EMA data was corrected for head movements and that it was translated to a coordinated system in relation to the occlusal plane.

2. PLUGIN STRUCTURE

Kijk's mechanism makes it possible to process kinematic trajectories completely in Praat, without requiring the user to directly manipulate the script. The plugin can be installed in the Praat preferences folder, or it can be called as a script. If the user decides to install the plugin, a button will appear on the dynamic menu after the user selects the EMA file (Figure 1). So far, it works for data collected with the EMA devices NDI Wave and Carsten's AG501, but it will soon be extended to support other devices, such

as Carsten's AG 100 and 200. The plugin is available via <https://github.com/linsca/kijk>.

2.1. Interface

Before running the plugin, the user should open the desired EMA file in Praat. For the NDI Wave this is a tab-separate file and for the AG501 a text file in tabular format. After the file is selected in Praat's object window, either click the plugin button on the dynamic menu (Figure 1) or run the script.

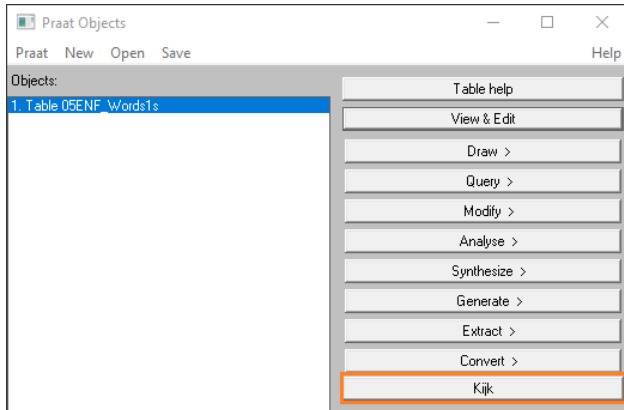


Figure 1: Partial screenshot of Praat's object window. If Kijk is installed as a plugin it appears as a button (encased in orange) on the dynamic menu when the table object is selected.

A pop-up form (Figure 2) will appear asking the user to (i) input the sensor number, (ii) the corresponding articulatory variable, (iii) the dimension (x - y - z position), (iv) the EMA device used, and (v) the sampling frequency. Further, if the user wishes to extract information from more than one sensor and combine it in a single output, there is an option to indicate that. It is important to mention that the plugin is case-sensitive. So, the inputted dimensions need to be given in uppercase.

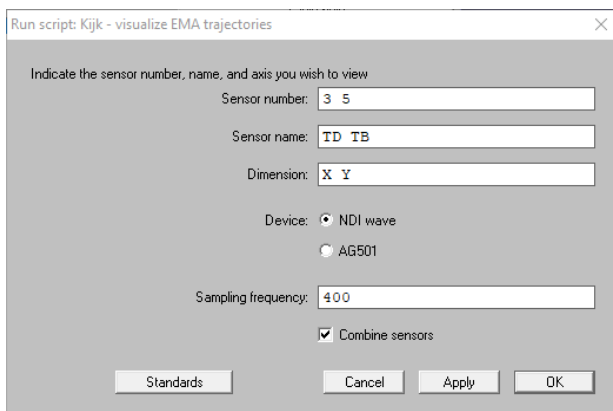


Figure 2: Screenshot of Kijk's initial pop-up form.

2.4. Algorithm

After the preferences have been entered in the pop-up form, the user can click "OK" or "Apply" to start the process. Depending on which EMA device the data stems from, preprocessing follows different procedures.

The output of the NDI Wave device includes nine values recorded per sensor: sensor ID, sensor status, sensor position in millimetres ($X[mm]$, $Y[mm]$, $Z[mm]$) and sensor orientation in quaternion format ($Q0$, Qx , Qy , Qz). The AG501 device outputs seven values per sensor: sensor position ($Ch\#_X$, $Ch\#_Y$, $Ch\#_Z$), orientation angles ($Ch\#_phi$, $Ch\#_theta$), root mean square ($Ch\#_RMS$), and a placeholder for an additional data column ($Ch\#_Extra$). As opposed to the AG501 device, the output file of the NDI Wave does not include the sensor number in the labels of its values, therefore, it is first necessary to rename the columns related to each sensor in order for Kijk's core algorithm to work.

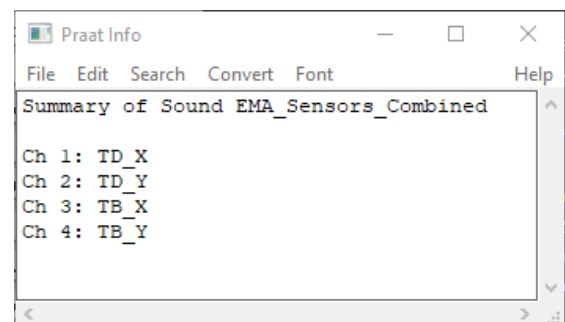
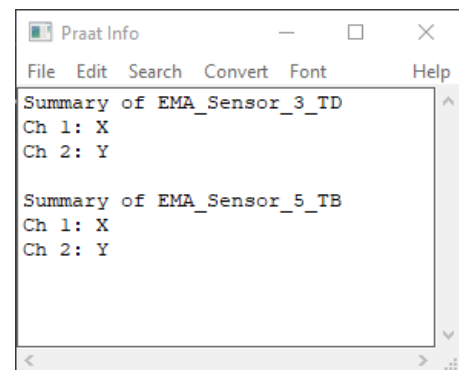


Figure 3: Praat Info windows displaying a summary of what each channel in the EMA object contains. The left screenshot is the output of sensors as independent objects and the right screenshot is the output of the sensors combined in one object.

The algorithm then commences the preparation of the data by first transposing the EMA file to a wide tabular format. Next, it surveys the transposed table to extract the information pertaining to the desired sensors and position axes (i.e. x - y - z dimensions). Afterwards, Praat converts the tabular data into a

matrix, which is then casted to a sound object. Finally, the sampling frequency of this object is overridden by the frequency inputted by the user.

3. OUTPUT

The output of the plugin is twofold. The first is a summary of the extraction procedure displayed on a Praat Info window. Depending on the user's selection of the "combine sensors" option in the initial form,

If the output is a combination of two (or more) sensors, the name of the EMA object is generic (EMA_Sensors_Combined). The following lines specify what each channel entails. In this example (Figure 3 bottom screenshot), channels 1 and 3 contain anteroposterior movement trajectories, and channels 2 and 4 contain vertical movement trajectories of tongue blade and dorsum, respectively TD_X, TB_X, TD_Y, TB_Y.

The secondary output of *Kijk* is the EMA

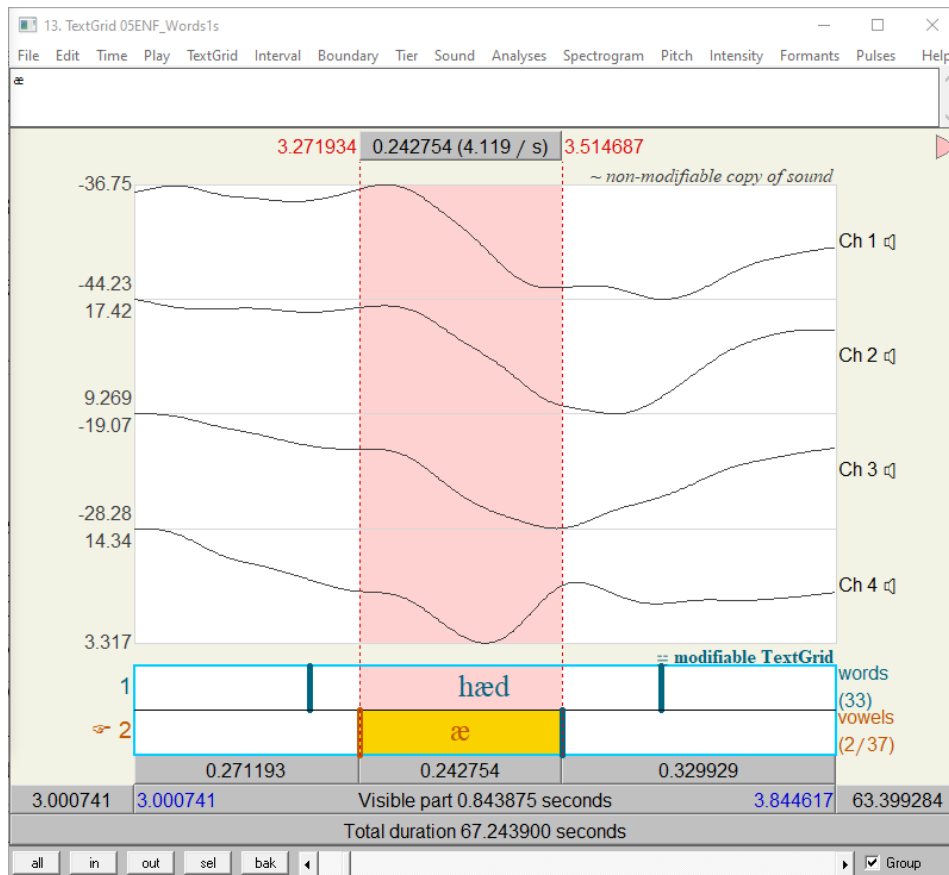


Figure 4: Screenshot of the Praat editor window displaying the combined EMA data from the EMA-MAE corpus [1] and TextGrid annotations; The channels contain tongue dorsum anteroposterior (Ch 1) and vertical (Ch 2) directions, and tongue blade anteroposterior (Ch 3) and vertical (Ch4) directions.

the output is either for each sensor (Figure 3 top screenshot) or of the combined sensors (Figure 3 bottom screenshot).

If the output is not a combined object; i.e. for each sensor, the name of the EMA object includes the sensor number and the articulatory variable it represents. EMA_Sensor_3_TD indicates that this object refers to the third sensor placed on a speaker's tongue dorsum (TD). The following lines indicate what each channel of the EMA object contains. In the object EMA_Sensor_5_TB channel 1 (Ch 1) contains the movement trajectory of the tongue blade (TB) in the anteroposterior direction (x) and channel 2 (Ch 2) TB's trajectory in the vertical direction (y).

trajectory object itself. It can be treated like any other audio signal in Praat. For instance, to eliminate some of the noise found in the data prior to visualisation, the signal encoding the kinematic trajectories was low pass filtered at 10 Hz (smoothing = 5Hz) using Praat's Hann filter. Similarly, this output can be visualised like any other sound object in Praat, by selecting it and clicking on "View and Edit" (Figure 4). In older versions of Praat the EMA object may appear off-scale. To remedy this issue, in the editor window the user can change the sound scaling strategy to "by window and channel".

Visualising EMA trajectories and corresponding acoustic data is also possible without needing to have

all the data stored in the same object. By selecting “Group” at the bottom right corner of the sound editor windows, selections done in a particular region of the acoustic object will automatically be reflected on the EMA object (Figure 5). This is possible because the collection of acoustic and kinematic data using these EMA devices is time-synchronized. In the example shown in the screenshot below, users can simultaneously visualise the oscillogram, the spectrogram, the annotations, and tongue dorsum’s anteroposterior (x) and vertical (y) kinematic trajectories of the selected segment /æ/.

4. CONCLUSION AND OUTLOOK

In this paper we introduced *Kijk*, a Praat plugin for the visualisation of EMA trajectories. Although limited to displaying only position data, a future

iteration will include the calculation of velocity and acceleration. Nonetheless, at the current stand, *Kijk* is a simple and straightforward alternative for students and researchers wishing for a quick visualisation of kinematic trajectories, or as an educational tool for introducing undergraduate students to EMA data along with the acoustic signal. Finally, *Kijk* was conceptualised as an open tool, therefore we welcome feedback and collaborations in future versions of this plugin.

5. ACKNOWLEDGEMENTS

This work was supported by the Forschungskredit of the University of Zurich (Grant No. FK-20-078), and the Swiss National Science Foundation (Grant #PZ00P1_193328) to LH.

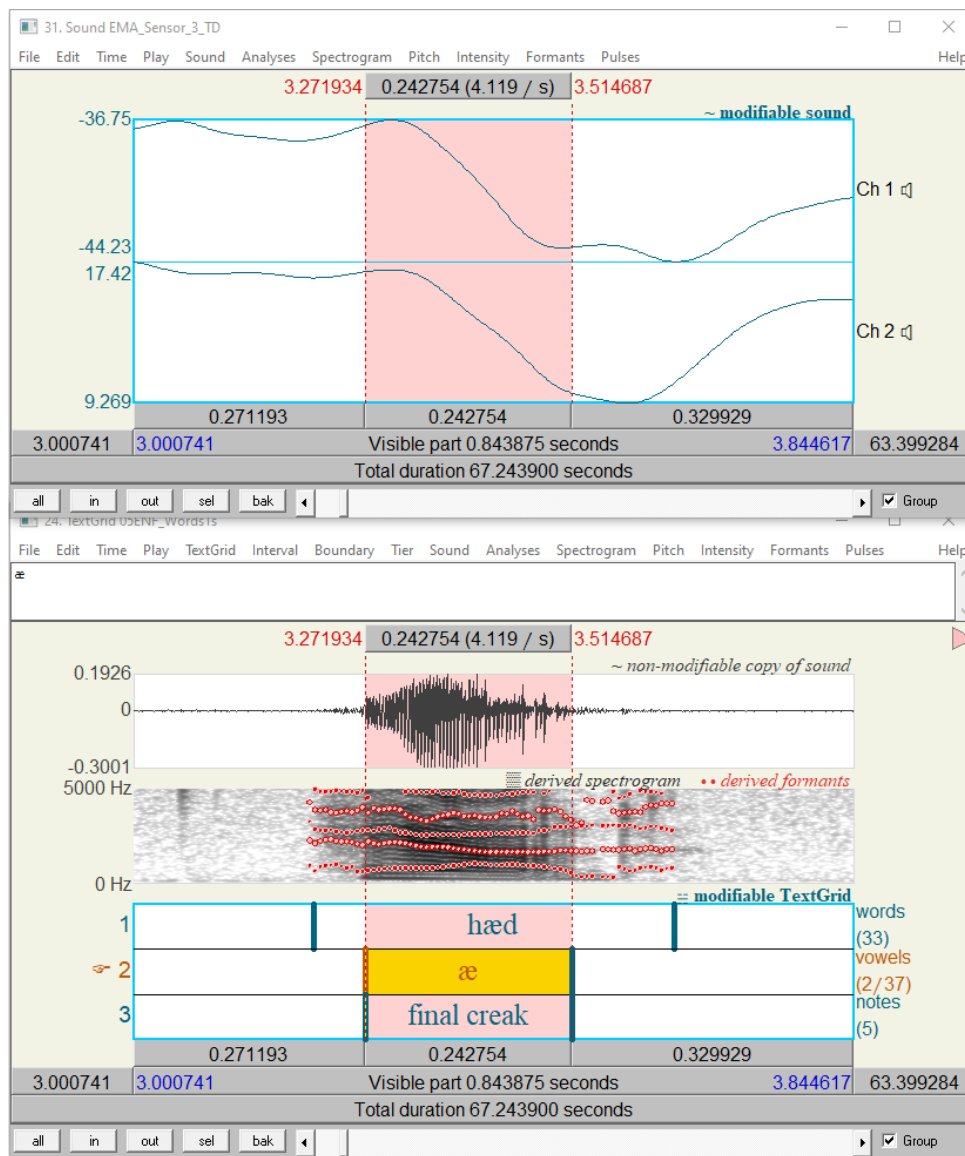


Figure 5: Screenshot of Praat editor windows grouped together. The top window displays filtered tongue dorsum trajectories in the x and y directions and the bottom window shows the oscillogram, spectrogram and TextGrid.

6. REFERENCES

- [1] A. Ji, J. J. Berry, and M. T. Johnson, “The Electromagnetic Articulography Mandarin Accented English (EMA-MAE) corpus of acoustic and 3D articulatory kinematic data,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 7719–7723. doi: 10.1109/ICASSP.2014.6855102.
- [2] A. Kochetov, “Research methods in articulatory phonetics I: Introduction and studying oral gestures,” *Language and Linguistics Compass*, vol. 14, no. 4, p. e12368, 2020, doi: <https://doi.org/10.1111/lnc3.12368>.
- [3] Articulate Instruments Ltd. Articulate Assistant Advanced User Guide: Version 2.14, Edinburgh, UK, 2012.
- [4] J. R. Green, J. Wang, and D. L. Wilson, “SMASH: a tool for articulatory data processing and analysis,” in *Proc. Interspeech 2013*, 2013, pp. 1331–1335. doi: 10.21437/Interspeech.2013-353.
- [5] M. Tiede, *MVIEW: Software for visualization and analysis of currently recorded movement data*. Haskins Laboratories, 2005.
- [6] “MATLAB.” The MathWorks Inc., Natick, Massachusetts, 2022.
- [7] P. Boersma and D. Weenink, “Praat: doing phonetics by computer.” Nov. 15, 2022. [Online]. Available: <http://www.praat.org/>.
- [8] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2022. [Online]. Available: <https://www.Rproject.org/>
- [9] R. N. Henriques and P. van Lieshout, “A Comparison of Methods for Decoupling Tongue and Lower Lip From Jaw Movements in 3D Articulography,” *Journal of Speech, Language, and Hearing Research*, vol. 56, no. 5, pp. 1503–1516, 2013, doi: 10.1044/1092-4388(2013/12-0016).
- [10] R. Winkelmann, K. Jaensch, S. Cassidy, and J. Harrington, *emuR: Main Package of the EMU Speech Database Management System*. 2021.
- [11] S. Ouni, L. Mangeonjean, and I. Steiner, “VisArtico: a visualization tool for articulatory data,” in *13th Annual Conference of the International Speech Communication Association - InterSpeech 2012*, Portland, OR, United States, Sep. 2012. [Online]. Available: <https://hal.inria.fr/hal-00730733>