

## Noël Nguyen: A Matlab package for the analysis of EMA/EPG data

### *Abstract*

The goal of this paper is to present a set of programs for displaying and annotating acoustic and articulatory data simultaneously in studies on speech production. These programs were developed in Matlab, a technical computing environment for numeric computation and visualisation.

The system is equipped with a mouse-driven graphical interface made up of a number of windows. This interface can be easily customised to speed up routine tasks. The programs can also be used in a non-interactive way, as standalone Matlab commands, when operations on large numbers of files are required. Output data can be imported into any standard spreadsheet.

The programs described in this paper have been already used on Unix workstations and PCs, and are freely available by ftp.

### *1 Introduction*

This paper describes a signal editor suitable for displaying, annotating and processing articulatory and acoustic data recorded in studies on speech production. While it is in theory possible to use it with any type of articulatory signal (e.g. EMG, kinematic data collected using an optical device, X-ray microbeam data, etc.), the system was specifically designed for the analysis of EMA and EPG data.

EMA (electromagnetic articulography) monitors articulatory movements by means of small electromagnetic transducers attached to the articulators in the mid-sagittal plane (Perkell et al., 1992; Hoole, 1993). EPG (electropalatography) is a well-established technique for recording the timing and location of tongue contacts with the hard palate during speech (Hardcastle et al., 1989). When combined with each other, EMA and EPG provide complementary information about the tongue movements in two orthogonal planes (the mid-sagittal and coronal planes, respectively). They both also have the advantage of being non-invasive, therefore enabling the experimenter to collect large amounts of data for a significant number of speakers.

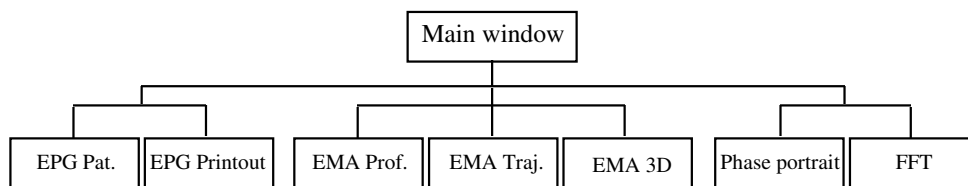
Our system was developed using the Matlab programming language. Matlab is a technical computing environment for numeric computation and visualisation. It integrates numerical analysis, matrix computation, signal processing, and graphics in the same environment. One of the advantages of Matlab is that it runs on all the current main platforms (Unix workstations, PCs, Macs). Matlab programs are in fact platform-independent, being simply made up of lists of instructions stored in ASCII files (thus, our own system is currently being used on SUN workstations and PCs). An increasing number of Matlab applications specific to speech are being made freely available by ftp (see Appendix for a non-exhaustive list).

## 2 The Graphical Interface

The main functionalities of the system are listed in the table below:

<i>Signal edition</i>	Edition/annotation of up to 15/20 synchronised audio/EMA/EPG signals
<i>Audio tools</i>	FFT, intensity, zero-crossing rate
<i>Signal processing</i>	moving average / smoothing, low-pass filtering
<i>Kinematic analyses</i>	Velocity, acceleration, tangential velocity, curvature, phase angles, standard EPG indices
<i>Displays</i>	EPG patterns, tongue “profiles”, trajectories, 3D combined EPG/EMA displays, phase planes, 2-sigma ellipses

These operations can be performed using a graphical interface, or in a non-interactive way, as indicated in the next section. The graphical interface is of a modular kind, i.e. it is made up of a number of windows each associated with a particular type of display.



*Figure 1: Architecture of the graphical interface*

The entire interface is controlled through one single window standing on top of the others, and which is called the Main window. In the first place, this window allows a set of acoustic and articulatory signals to be edited and annotated. Each signal is plotted in a subregion of the window which is referred to as a “track”. The number of signals simultaneously displayed in the window can be modified and its maximal value mainly depends on the screen size.

Most of the usual editing functions (zooming, annotating, etc.) are performed using two cursors which can be displaced along the time axis with the mouse. Markers can be placed at any particular location in a signal, and be stored in ASCII files for later use. It is also possible to measure the sample value at the cursor location for a given signal, and save this value in a file, which can be imported into any standard spreadsheet for further processing. In order to make the annotation of signals a little less tedious, the system also enables to locate local extrema on a curve in an automatic manner, by means of an extremum-identification function.



- 5) The ‘EMA 3D’ window: a display of the  $x$  and  $y$  coordinates of the coils in the midsagittal plane combined with a 3D representation of the artificial palate (when such a representation is available), as shown in Figure 3;
- 6) The ‘FFT’ window: displays an FFT of the acoustic signal over a time window centred at the location in time of each cursor;
- 7) The ‘Phase portrait’ window: a display of the trajectory associated with any given articulatory parameter in the corresponding phase plane (velocity against displacement).

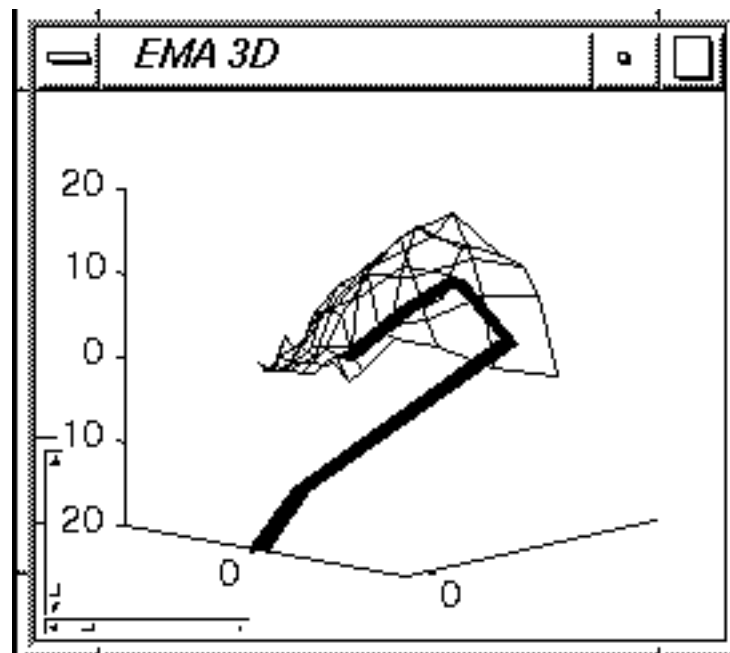


Figure 3: The EMA 3D window

These displays are all dynamically linked to the Main window, i.e. they are permanently refreshed as cursors are moved along the time axis in this window. The software also allows displays to be copied and assembled with each other in a new window, and/or to be saved in external files for later use. Figures can also be printed in a large variety of formats.

### 3 Customisation / Macros / standalone functions

In designing the present system, we attempted to associate each function with a particular Matlab program, or ‘M-file’, whenever this was possible. As a result, most of these functions can be performed by typing the name of the corresponding command in the Matlab command window, therefore bypassing the pop-up menus of the graphical interface. For example, the command `clrtrack(3)` will clear the third track in the Main window, whereas the command `ematrj` will generate a new window displaying the trajectories of the EMA coils between the two cursors.

These commands can themselves be stored into a new M-file, which can then be executed in exactly the same way than those already available. This is a convenient method for creating ‘macros’ to streamline repetitive tasks. By way of example, here is a short list of instructions for loading a file of markers and opening three new different windows:

```

mrkrload('kip001.mrk'); % load markers from the file kip001.mrk
epgfig;                % open EPG window
emafig;                % open EMA window with tongue profiles
ematraj;               % open EMA window with coils traj.

```

If these instructions are stored in an ASCII file called `myfunct.m` for example, typing `myfunct` in the Matlab command window will produce the expected result. As an alternative solution, the new function can be made directly available from the graphical interface as a new, user-specific menu. This can be achieved using the two following commands:

```

M = uimenu('Label', 'My menu');           % create a new menu
uimenu(M, 'Label', 'My function', ...    % add "myfunct" to this menu
        'Callback', 'myfunct');

```

As can be seen, the interface can easily be tailored to the specific needs of each user. The two above examples only show how to use that interface in a more efficient manner by writing short M-files that “mimic” a series of operations that normally have to be accomplished by hand. However, the system is built upon a series of functions whose range of application was intended to be wider than that of the interface. It is up to a certain extent possible to use these functions in a non-interactive way, as standalone Matlab commands, when operations on large numbers of files are required. In particular, we have started to develop procedures for pooling and processing EMA data associated with different repetitions/items. Consider for example the following commands:

```

[x,t] = readsig('kip001.x?',200);
[y,t] = readsig('kip001.y?',200);
[xx,t] = smooth(y,t,11);
[yy,t] = smooth(yy,t,11);
plot(xx,yy);
el2sig(xx(:,1),yy(:,1));
el2sig(xx(:,2),yy(:,2));
el2sig(xx(:,3),yy(:,3));
el2sig(xx(:,4),yy(:,4));
el2sig(xx(:,5),yy(:,5));
el2sig(xx(:,6),yy(:,6));

```

The first two `readsig` commands allow the  $x$  and  $y$  coordinates of all the available coils to be read and stored into two different arrays for the sentence “it's a kip again” (as spoken by a female English speaker). The EMA parameters are then smoothed using a cubic spline function, and plotted as a set of trajectories in the midsagittal plane. Finally, a 2-sigma ellipse is drawn around the trajectory associated with each coil. The result is shown in Figure 4.

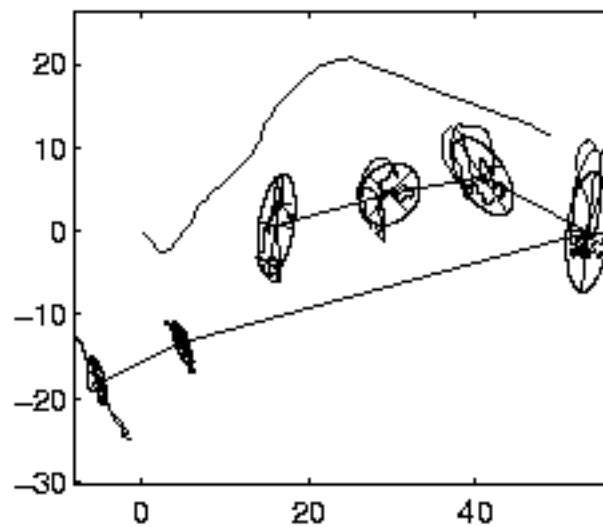


Figure 4: Coil trajectories in the midsagittal plane for the sentence "It's a kip again" spoken by a female speaker. Anterior is to the right. An outline of the subject's palate is also shown. A  $2\text{-}\sigma$  ellipse was drawn around the trajectory associated with each coil.

#### 4 Conclusion - future developments

In this paper, we have presented a new set of programs for editing, annotating and processing acoustic, EMA and EPG data in an integrated environment. While being still at the experimental stage, this system has already proven to be a useful tool in studies on speech production (Hardcastle et al., 1996). It is developed with a view of taking full advantage of Matlab numerical and graphical capabilities. Further improvements will include the integration to the system of data reduction techniques, such as principal-component analysis and multilayer neural networks.

#### Acknowledgements

Work supported by Grant #8210-043017 from the Swiss National Foundation for Research.

#### References

- Hardcastle, W.J., Jones, W., Knight, C., Trudgeon, A., & Calder, G. (1989). "New developments in electropalatography: a state-of-the-art report," *Clinical Linguistics & Phonetics* **3**, 1-38.
- Hardcastle, W.J., Vaxelaire, B., Gibbon, F., Hoole, P., & Nguyen, N. (1996). "Tongue kinematics in /kl/ clusters and singleton /k/: A combined EMA/EPG study," *Proceedings of the Australian Speech Science and Technology Conference*, Adelaide, Dec. 96. (in press).
- Hoole, P. (1993). "Methodological considerations in the use of electromagnetic articulography in phonetic research," *Forschungsberichte des Instituts für Phonetik und Sprachliche Kommunikation der Universität München* **31**, 43-64.
- Hoole, P., & Nguyen, N. (1992). "Reduction and parameterization of electropalatographic and electromagnetic data for the investigation of articulatory-acoustic correlations," *Forschungsberichte des Instituts für Phonetik und Sprachliche Kommunikation der Universität München* **30**, 45-72.

Nguyen, N. (1992). 'L'analyse électromagnétométrique des mouvements articulatoires dans la production de la parole,' *Travaux de l' Institut de Phonétique d' Aix-en-Provence* **14**, 165-186.

Nguyen, N., & Marchal, A. (1993). "Assessment of an alternating magnetic fields device for tracking articulatory movements in speech production," *J. Acoust. Soc. Am.* **94**, 1152-1155.

Nguyen, N., Marchal, A., & Content, A. (1996). 'Modeling tongue-palate contact patterns in the production of lingual consonants,' *Journal of Phonetics* **24**, 77-97.

Perkell, J.S., Cohen, M.H., Svirsky, M.A., Matthies, M.L., Garabieta, I., & Jackson, M.T.T. (1992). 'Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements,' *J. Acoust. Soc. Am.* **92**, 3078-3096.

*Appendix: ftp-available softwares*

The programs described in this paper are freely available on the Web page <http://www.unige.ch/fapse/psycholing/noel/ematools.html>.

Other Matlab ftp-available programs of interest include the following:

**SPC Toolbox:** A collection of Matlab GUI (Graphical Use Interfaced) applications for performing signal processing, developed by D. Brown. Available by anonymous ftp from <ftp.nps.navy.mil>, directory /pub/ece/spctools/software.

**Osprey:** A Matlab program for computing spectrograms, developed by D. Mellinger. Available by anonymous ftp from <ftp.mathworks.com>, directory /pub/contrib/signal.

**Track-Draw:** Matlab implementation of the Klatt formant synthesiser, developed by P. Assmann. Available from Peter Assmann (e-mail: [assmann@utdallas.edu](mailto:assmann@utdallas.edu)).

**Speech Tool:** A set of speech analysis functions written in Matlab by L. Arslan (e-mail: [larslan@ee.duke.edu](mailto:larslan@ee.duke.edu)).

**Auditory Toolbox:** A set of functions useful to people interested in auditory/cochlear modelling, and using auditory models in their work. Developed by Malcolm Slaney, and available by anonymous ftp from <ftp.apple.com>, directory /pub/malcolm.