Articulatory Data Recorder: A Framework for Real-Time Articulatory Data Recording

Alexander Wilbrandt, Simon Stone, Peter Birkholz

Institute of Acoustics and Speech Communication, Technische Universität Dresden, Germany
alexander.wilbrandt@tu-dresden.de

Abstract

Articulatory data can be collected using numerous modalities, such as video, ultrasound, electromagnetic articulography, or palatographic techniques. Every measurement technique requires software to visualize the incoming data and export the data for further analysis. This has led to an increase of available recording software over the past decades, including properly maintained software in regular use but also many abandoned and dead projects. In this paper, we present a new framework for real-time, simultaneous recording of acoustic and articulatory data. With the release of the Articulatory Data Recorder, our aim is to provide the experimental phonetics and articulatory research community with a common framework that is simple to use and easy to extend. It is specifically designed to cover the most common use cases in experimental phonetics: Elicit speech utterances using text prompts and record simultaneous audio and articulatory data. By following the FURPS+-system, we offer a combination of high performance and a low barrier of entrance for enrollment of any new articulatory measurement technique. The current version already supports various palatographic measurement techniques in use at our institute and future work will incorporate feedback and feature requests from the community.

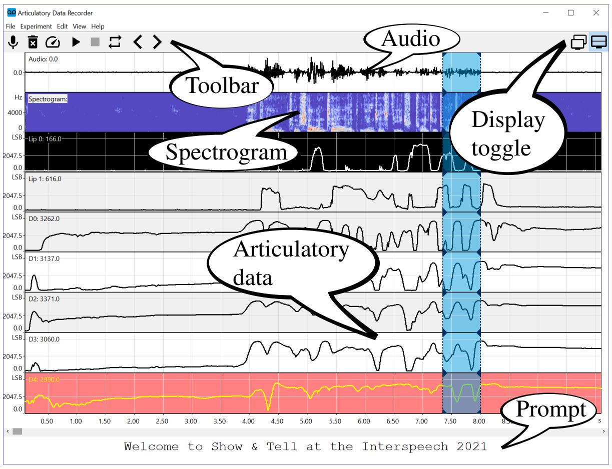
Index Terms: articulography, articulatory data, experimental phonetics, human-computer interaction

1. Introduction

Articulatory data is of great interest in numerous fields of research from speech recognition to speech production and synthesis and many technologies have been developed or repurposed to capture it [1]. Mirroring the great variety of measuring techniques and systems, a myriad of software applications have been developed to enable the communication between data acquisition devices and the processing unit (usually a personal computer) and to analyze the measured data (e.g., for electromagnetic articulography data [2, 3, 4] or for ultrasound and electropalatography data [5]). The lack of compatibility between the various applications and software setups makes the reproduction or validation of findings from articulatory studies more difficult and hinders collaboration between research groups. The goal of the Articulatory Data Recorder (ADR) is to unify the communication and visualization of any kind of articulatory data. By introducing a universal data format, downstream analysis software may become interchangeable and thus the whole pipeline more open.

2. Articulatory Data Recorder

The ADR is a PC software written in C++ (using the free wxWidgets library available from `www.wxwidgets.org`). It is developed following the FURPS+-system [6], which makes it highly adaptable and easy to extend. For now, data recording and visualization is supported for four different palatographic measurement technologies in use at the TU Dresden [7, 8, 9]. Future releases will include other devices, prioritized based on community feedback and provided that the necessary hardware protocols are available. The ADR provides a flexible and adaptable data acquisition backend and a graphical user interface that is highly customizable and easy to extend by implementing specialized pictures to display virtually any kind of articulatory data. The ADR also supports an important use case for articulatory studies: Triggering a subject to produce speech using text prompts. Future releases will also support other types of prompts, e.g. audio or pictures.



Welcome to Show & Tell at the Interspeech 2021

Figure 1: *Screenshot of the graphical user interface of the ADR, showing a sequence recorder with an optopalatographic setup.*

2.1. Graphical User Interface

Figure 1 shows the recording of optopalatographic data visualized using the graphical user interface (GUI) of the ADR. Various configuration and visualization options are accessible via dropdown menus and toolbar buttons. The GUI is mainly separated into a data page, on which acoustic and articulatory data received from the hardware are visualized, and a prompts page, on which the current prompt is shown if an experiment is currently set up. Using the top right buttons, the layout of the software can be toggled between a single- and a dual-screen setup. Dual-screen setup is especially helpful during user studies to prevent cluttered screen facing the subject. The data page layout depends on the currently selected hardware backend and changing the hardware backend will lead to an automatic refresh of the data page. Tracks can be hidden, which simplifies focusing on specific tracks if needed. The user can choose to save and export a selection of the data or the complete recording as an ADR-file, which is a novel data format holding articulatory data. It is text-based and follows the same basic format across all kinds of articulatory input modalities. Audio is saved as WAV-file to provide maximum compatibility.

2.2. Architecture

The framework mainly consists of three major parts, namely backend, frontend, and an interface connecting backend and frontend.

2.2.1. Backend

The software backend generally consists of abstract base classes that can be specialized to adapt them to a specific kind of hardware. To acquire real-time articulatory data from a measurement device, two steps are necessary regardless of the underlying technology: A chunk of data needs to be collected and then parsed. Every hardware backend (i.e., an articulatory measurement device) is defined by a DataCollector and a DataParser. The type of raw data coming from the hardware can either be ”Byte” or ”String”, which therefore requires specializations of DataCollector and DataParser for these types (see Figure 2 and Figure 3).

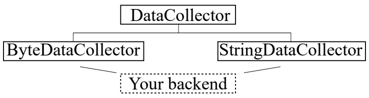


Figure 2: *Inheritance diagram for classes derived from DataCollector.*

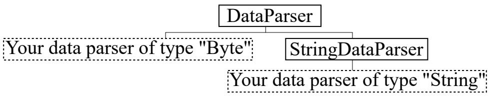

DataParser
Your data parser of type "Byte"
StringDataParser
Your data parser of type "String"

Figure 3: *Class structure for definition of data parsers used in the ADR.*

Therefore, to connect a new hardware backend to the ADR, a specialized DataCollector and DataParser need to be implemented. Depending on the chosen data format (”Byte” or ”String”), simply using a ByteDataCollector or a StringDataCollector should suffice in most cases. The DataParser, which needs to be aware of the communication protocol and the interpretation of each piece of data in the collected chunk, is expected to be far more application-dependent and thus requires a bit more work to implement. If the data is transmitted as a simple formatted string, however, the class StringDataParser may be usable as is by simply providing the format string.

2.2.2. Frontend

Different sensor types require different visualizations. Although there is great variety of possible sensors and thus visualizations, all pictures in the ADR are derived from the same BasicPicture. For now, four different kinds of pictures have been implemented (see Figure 4). Most pictures in the ADR show time-dependent signals and are therefore derived from GraphPicture, but other kinds of pictures can also be implemented by deriving directly from BasicPicture. An example where that is useful is the visualization of EPG data: A ContactPatternPicture visualizes a single, 2D contact pattern captured by a EPG device, and a ContactSignalPicture combines these individual patterns into a sequence over time and is therefore derived from GraphPicture. The ScalarSignalPicture class is used to visualize all kinds of one-dimensional signals (e.g., an audio signal or a scalar sensor signal). A SpectrogramPicture calculates and shows a spectrogram of the audio track.

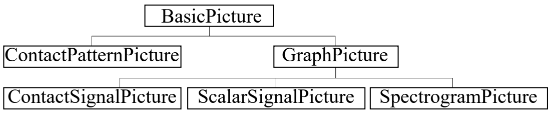


Figure 4: *Class structure for pictures used in the ADR.*

3. Conclusions and Outlook

The presented software and the underlying C++ framework provides functionality for the simultaneous recording of acoustic and articulatory data with different articulometric devices. The visualization is adapted to the connected hardware device automatically. The software is highly flexible and enables joint development and integration of additional articulatory data recording hardware with minimal effort using the presented classes. The ADR framework can serve as a common basis for the articulatory data recording community to work on a jointly used recording software, which can boost data exchange possibilities and thus, hopefully, speed up the progress in the field as a whole. The software is available from `https://gitlab.hrz.tu-chemnitz.de/spreuss--tu-dresden.de/ArticulatoryDataRecorder`.

4. Acknowledgements

This work was partly funded by the German Federal Ministry of Education and Research, reference number 01IS19019B.

5. References

[1] B. Denby, T. Schultz, K. Honda, T. Hueber, J. Gilbert, and J. Brumberg, “Silent Speech Interfaces,” *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[2] S. Ouni, L. Mangeonjean, and I. Steiner, “VisArtico: a visualization tool for articulatory data,” in *13th Annual Conference of the International Speech Communication Association*, 2012.

[3] N. Nguyen, “A MATLAB toolbox for the analysis of articulatory data in the production of speech,” *Behavior Research Methods, Instruments, Computers*, vol. 32, pp. 464–467, 2000.

[4] M. Tiede, “MVIEW: Multi-channel visualization application for displaying dynamic sensor movements,” *development*, 2010.

[5] A. Wrench, “Articulate assistant advanced user guide: Version 2.17,” 2017.

[6] R. B. Grady, *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Inc., 1992.

[7] S. Stone and P. Birkholz, “Cross-speaker silent-speech command word recognition using electro-optical stomatography,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 7849–7853.

[8] K. Große and P. Birkholz, “Tongue mouse - comparison of physical measurement principles,” in *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*, 2020.

[9] C. Wagner, L. Stappenbeck, H. Wenzel, P. Steiner, B. Lehnert, and P. Birkholz, “Evaluation of a non-personalized optopalatographic device for prospective use in functional post-stroke dysphagia therapy,” 2021, Manuscript submitted for publication.