

Utilising temporal signal features in adverse noise conditions: Detection, estimation and the reassigned spectrogram

Supporting Code Documentation

Robert Mill • Guy J Brown

Introduction

This document accompanies the paper “Utilising temporal signal features in adverse noise conditions: Detection, estimation and the reassigned spectrogram” in the *Journal of the Acoustical Society of America* and the Matlab/C code package **JASA-DetEstTemporal.zip**.

Herein we describe how to build the code which produces the figures in the paper.

Getting Started

Unzip the contents of the file **JASA-DetEstTemporal.zip** in a convenient location.

Directory contents

Start Matlab and navigate to the root folder.

ACTION: Typing `dir` one should see

```
.          ..          C          Doc          Matlab  Mex
```

The folder **Matlab** contains Matlab scripts and is required to produce Figures 1 – 8. The folders **C** and **Mex** contain C and Matlab-executable (MEX) C code, respectively. The code in these files and a C/MEX compiler are required to produce Figures 6 – 8. (The code listings in **C** are Matlab-independent and can be compiled without Matlab.) The folder **Doc** contains a copy of this file.

Add directories to Matlab path

Add the contents of all directories to the Matlab path.

ACTION: To add all directories and change to the **Matlab** directory type

```
addpath(genpath(pwd));  
cd Matlab
```

ACTION: Typing `dir` one should see

```
.          data   ifr     make   swdft  zci  
..         env    ifrh   mgm    util   zcpa
```

These are the main directories that group code files with related functions. To get a description of the contents type `help <folder>`, for instance, `help zci`. (This does not apply to the folder **data**.) Further details about the code is supplied below.

Compiling the C code

The code that produces Figures 6 – 8 requires the C code to be compiled. From the **Matlab** directory (which should be the current directory following steps above), proceed as follows.

ACTION: Type

```
        buildC
then
        buildMEX
```

If the scripts above run without error and conclude with the message **Done** , then the compilation has succeeded.

Producing Figure Files

Figure 1

ACTION: To produce Figure 1, type

```
make_figure1
```

The figure should be produced after a few seconds.

Figure 2

ACTION: To produce panel B of Figure 2, type

```
make_figure2B
```

The figure should be produced after a few seconds.

Figure 3

ACTION: To produce Figure 3, type

```
make_figure3
```

The figure should be produced after a few seconds.

Figure 4

ACTION: To produce Figure 4, type

```
make_figure4
```

The figure should be produced after a few seconds. Figure 4 relies on pre-computed data which resides in the **data** folder. This data takes a while to produce but results in small files, so these have been included in the package. However, these data files can be deleted and reconstituted by running the following from the **Matlab** folder:

```
make_figure4_data1
make_figure4_data2
```

Please note that the production of this data takes several minutes.

Figure 5

ACTION: To produce Figure 5, type the first line below, wait, and then type the second.

```
make_figure5_data  
make_figure5
```

The first line computes and writes the numerical contents of Figure 5 to the **data** folder. The data file takes a few minutes to produce. The second line produces the figure using this pre-computed data. The figure should be produced instantly.

Figure 6

This figure requires the successful compilation of the C/MEX code (please see above).

ACTION: To produce Figure 6, type

```
make_figure6
```

The figure should be produced after a couple of minutes. A large quantity of data is produced so it may be necessary to clear the working memory in Matlab by first typing **clear**, (*return*) then **pack**. The code displays its progress as it runs.

Figure 7

This figure requires the successful compilation of the C/MEX code (please see above) for production of the data file.

ACTION: To produce Figure 7, type the first line below, wait, and then type the second.

```
make_figure7and8_data  
make_figure7
```

The first line computes and writes the numerical contents of Figures 7 and 8 to the **data** folder. The data file takes a few minutes to produce and is substantial in size (~160MB). The second line produces the figure itself using this pre-computed data.

Figure 8

This figure requires the successful compilation of the C/MEX code (please see above) for production of the data file.

ACTION: If the data file for Figures 7 and 8 has already been produced, then to produce Figure 8, type

```
make_figure8
```

The figure should be produced after a few seconds.