

YAAPT Pitch Tracking MATLAB Function

1. INTRODUCTION

The YAAPT (Yet Another Algorithm for Pitch Tracking) is a fundamental frequency (Pitch) tracking algorithm, which is designed to be highly accurate and very robust for both high quality and telephone speech.

The YAAPT program is developed by the Speech Communication Laboratory at the State University of New York (SUNY) at Binghamton. The entire algorithm is available at <http://www.ws.binghamton.edu/zahorian> as MATLAB functions. Further information including algorithm overview, implementation details, parameter settings and performance comparison can be found at S. A. Zahorian and H. Hu "A spectral/temporal method for robust fundamental frequency tracking," J.Acosut.Soc.Am. 123(6), June 2008 [1].

To cite YAAPT in your publications, please refer to:

A spectral/temporal method for robust fundamental frequency tracking

Stephen A. Zahorian, Hongbing Hu

The Journal of the Acoustical Society of America, 123, 4559-4571

2. DOWNLOAD

- Download the YAAPT package ([yaapt.zip](#), ver 3.0, Dec 11, 2013)
- Download this instruction in Word format ([YAAPT_Pitch_Tracking.docx](#))

Revision History

- **June 10, 2012:** Initial online version
- **Nov 1, 2012:** Added the FAQ and revision sections
- **Dec 3, 2013:** Updated the YAAPT package and associated document changes

3. USAGE

The YAAPT pitch tracking algorithm is implemented as a MATLAB function `yaapt()`, which checks input parameters and invokes a number of associated routines for the YAAPT pitch tracking. The function call format is described as follows:

[Pitch, numfrms, frmrate] = yaapt(Data, Fs, VU, ExtrPrm, fig)

INPUTS:

Data: Input speech acoustic samples

Fs: Sampling rate of the input data

VU: Whether to make voiced/unvoiced decisions with 1 for True and 0 for False. The default is 1. As 0, the entire acoustic segment is considered all or nearly all voiced

ExtrPrm: Extra parameters in a struct type for performance control. Default values are given in a later section in a table. Some examples are given of details are given here.

```
ExtrPrm.f0_min = 60;           % Change minimum search F0 to 60Hz
ExtrPrm.fft_length = 8192;    % Change FFT length to 8192
```

fig: Whether to plot pitch tracks, spectrum, energy, etc. The parameter is 1 for True and 0 for False. The default is 0.

OUTPUTS:

Pitch: Final pitch track in Hz. Unvoiced frames are assigned to 0s.

numfrms: Total number of calculated frames, or the length of output pitch track

frmrate: Frame rate of output pitch track in ms

3.1 Algorithm Control Parameters

The YAAPT utilizes a set of parameters for algorithm control, such as frame length, FFT length, and dynamic programming weights. By choosing different parameters, the YAAPT can be tuned to compute a pitch track with voiced/unvoiced decision for the minimum big error, or a track without voiced/unvoiced decision for the minimum gross error. The gross error is computed as the percentage of voiced frames, such that the pitch estimate of the tracker significantly deviates (20% is generally used) from the pitch estimate of the reference, while the big error is equal to the number of voiced frames with gross errors plus the number of unvoiced frames erroneously labeled as voiced frames divided by the total number of frames [1].

The parameters and their optimal values for both the minimum big and gross errors as listed in Table 1 in the work of S. A. Zahorian and H. Hu [1]. In the program, the corresponding parameters are declared in a struct type. As shown below, Prm_VU contains the default values for the tracking with voiced/unvoiced decision (gross error) and Prm_aV for the tracking with all frames voiced (big error).

```
% Default values for the tracking with voiced/unvoiced decision
Prm_VU = struct(...
    'frame_length',    25, ... % Length of each analysis frame (ms)
    'frame_space',    10, ... % Spacing between analysis frame (ms)
    'f0_min',          60, ... % Minimum F0 searched (Hz)
    'f0_max',          400, ... % Maximum F0 searched (Hz)
    'fft_length',      8192, ... % FFT length
    'bp_forder',       150, ... % Order of bandpass filter
    'bp_low',          50, ... % Low frequency of filter passband (Hz)
    'bp_high',         1500, ... % High frequency of filter passband (Hz)
    'nlfer_thresh1',   0.75, ... % NLFER boundary for voiced/unvoiced decisions
    'nlfer_thresh2',   0.1, ... % Threshold for NLFER definitely unvoiced
    'shc_numharms',     3, ... % Number of harmonics in SHC calculation
    'shc_window',      40, ... % SHC window length (Hz)
    'shc_maxpeaks',    4, ... % Maximum number of SHC peaks to be found
    'shc_pwidth',      50, ... % Window width in SHC peak picking (Hz)
    'shc_thresh1',     5.0, ... % Threshold 1 for SHC peak picking
    'shc_thresh2',     1.25, ... % Threshold 2 for SHC peak picking
    'f0_double',       150, ... % F0 doubling decision threshold (Hz)
    'f0_half',         150, ... % F0 halving decision threshold (Hz)
    'dp5_k1',          11, ... % Weight used in dynamic program
    'dec_factor',       1, ... % Factor for signal resampling
    'nccf_thresh1',    0.3, ... % Threshold for considering a peak in NCCF
    'nccf_thresh2',    0.9, ... % Threshold for terminating search in NCCF
    'nccf_maxcands',   3, ... % Maximum number of candidates found
    'nccf_pwidth',     5, ... % Window width in NCCF peak picking
    'merit_boost',     0.20, ... % Boost merit
    'merit_pivot',     0.99, ... % Merit assigned to unvoiced candidates in
    ... % definitely unvoiced frames
    'merit_extra',     0.4, ... % Merit assigned to extra candidates
    ... % in reducing F0 doubling/halving errors
    'median_value',    7, ... % Order of median filter
    'dp_w1',           0.15, ... % DP weight factor for V-V transitions
    'dp_w2',           0.5, ... % DP weight factor for V-UV or UV-V transitions
    'dp_w3',           0.1, ... % DP weight factor of UV-UV transitions
    'dp_w4',           0.9, ... % Weight factor for local costs
    'end', -1);
```

A convenient way to modify the above parameters is to use the input ExtrPrm parameter to yaapt(). For instance, assign desired parameter values using a struct ExtrPrm in the MATLAB command window, and pass ExtrPrm to the yaapt() function as follows,

```
>> ExtrPrm.f0_min = 60;           % Change minimum F0 searched to 60Hz
>> ExtrPrm.f0_max = 400;          % Change maximum F0 searched to 400Hz
>> ExtrPrm.fft_length = 8192;     % Change FFT length to 8192
>> [pitch] = yaapt(data, fs, ExtrPrm); % Execute YAAPT with the changes
```

4. EXAMPLES

A number of examples are provided here to demonstrate how to use the YAAPT program in MATLAB for different scenarios. Two sample speech files (flnw0000pes_short.wav and mlnw0000pes_short.wav, 16 bit, 20 kHz sampling) are also provided in the sample

folder of the YAAPT package available here for download. These examples can also be used to assist in verifying whether you have a proper YAAPT setup.

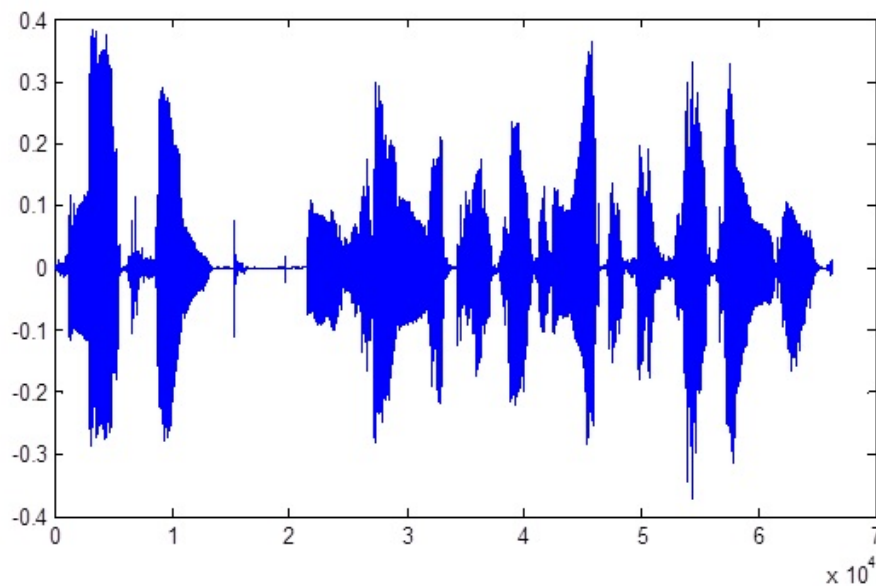
4.1 Computing pitch track with voiced/unvoiced decision for the minimum big error

- 1) In the MATLAB command window, go the folder where the YAAPT program is located.
- 2) Read speech data from the sample sample/flnw0000pes_short.wav file.

```
>> [Data, Fs] = wavread ('sample/flnw0000pes_short.wav');
```

Plot the data as shown in the figure below to verify the data has been read correctly.

```
>> plot(Data);
```

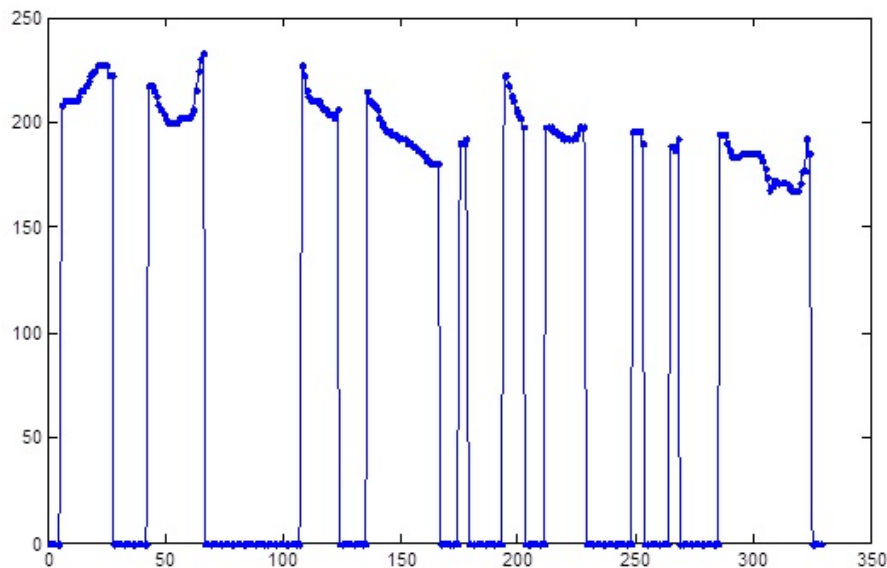


- 3) Compute the pitch track with the yaapt() function. The computed pitch tracking is saved in an array Pitch of length nf.

```
>> [Pitch, nf] = yaapt(Data, Fs);
```

The plot of the pitch track is shown in the figure below.

```
>> plot(Pitch, '-');
```



4.2 Computing pitch track without voiced/unvoiced for the minimum gross error

- 1) In the MATLAB command window, go the folder where the YAAPT program is located.
- 2) Read speech data from the sample sample/ flnw0000pes_short.wav file.

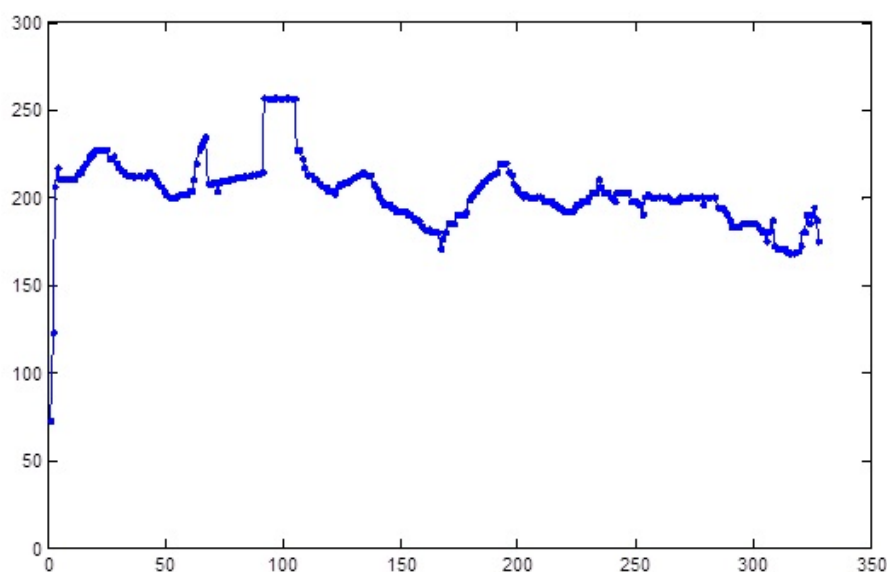
```
>> [Data, Fs] = wavread ('sample/flnw0000pes_short.wav');
```

- 3) Compute pitch track with the yaapt() function. The third parameter (VU) is set to 0 so that no voiced/unvoiced decision will be performed. The computed pitch tracking is saved in an array Pitch of length nf.

```
>> [Pitch, nf] = yaapt(Data, Fs, 0);
```

The plot of the pitch track is shown in the figure below.

```
>> plot(Pitch, '-');
>> ylim([0, 300]); % Set the Y axis to [0, 300]
```



4.3 Adjusting algorithm by passing different parameters

- 1) In the MATLAB command window, go the folder where the YAAPT program is located.
- 2) Read speech data from the sample/sample/flnw0000pes_short.wav file.

```
>> [Data, Fs] = wavread ('sample/flnw0000pes_short.wav');
```

- 3) Use a struct type ExtraPrm to define the parameters needs to be modified. The optimized parameters for this case are listed in Table 1 in the work of S. A. Zahorian and H. Hu [1].

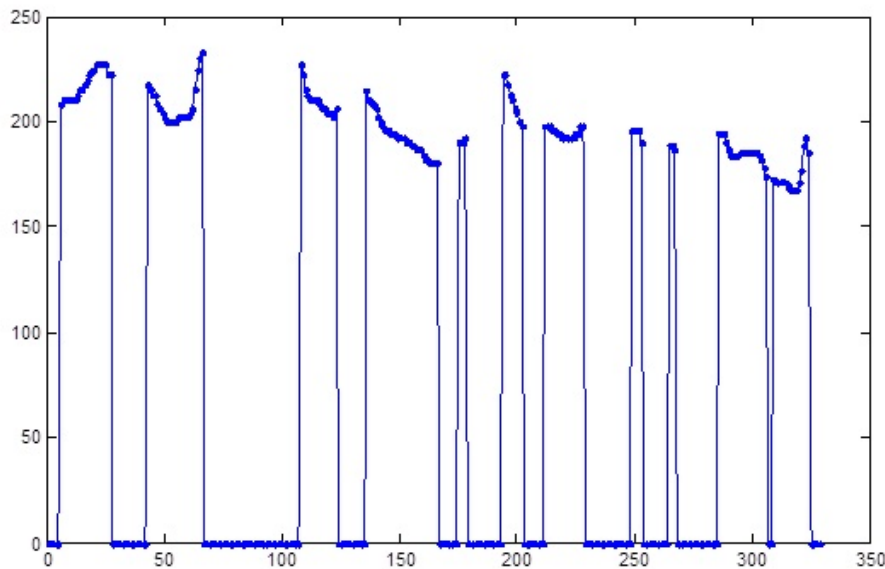
```
>> ExtrPrm.f0_min = 60;           % Change minimum F0 searched to 60Hz
>> ExtrPrm.fft_length = 4096;    % Change FFT length to 8192
>> ExtrPrm.dp_w1 = 0.5;          % Change DP V-V transition weight to 0.5
```

- 4) Compute pitch track with the yaapt() function with modified parameters. The computed pitch tracking is saved in an array Pitch of length nf.

```
>> [Pitch, nf] = yaapt(Data, Fs, 1, ExtrPrm);
```

The plot of the pitch track is shown in the figure below.

```
>> plot(Pitch, '-');
```



4.4 Enabling plots during pitch tracking

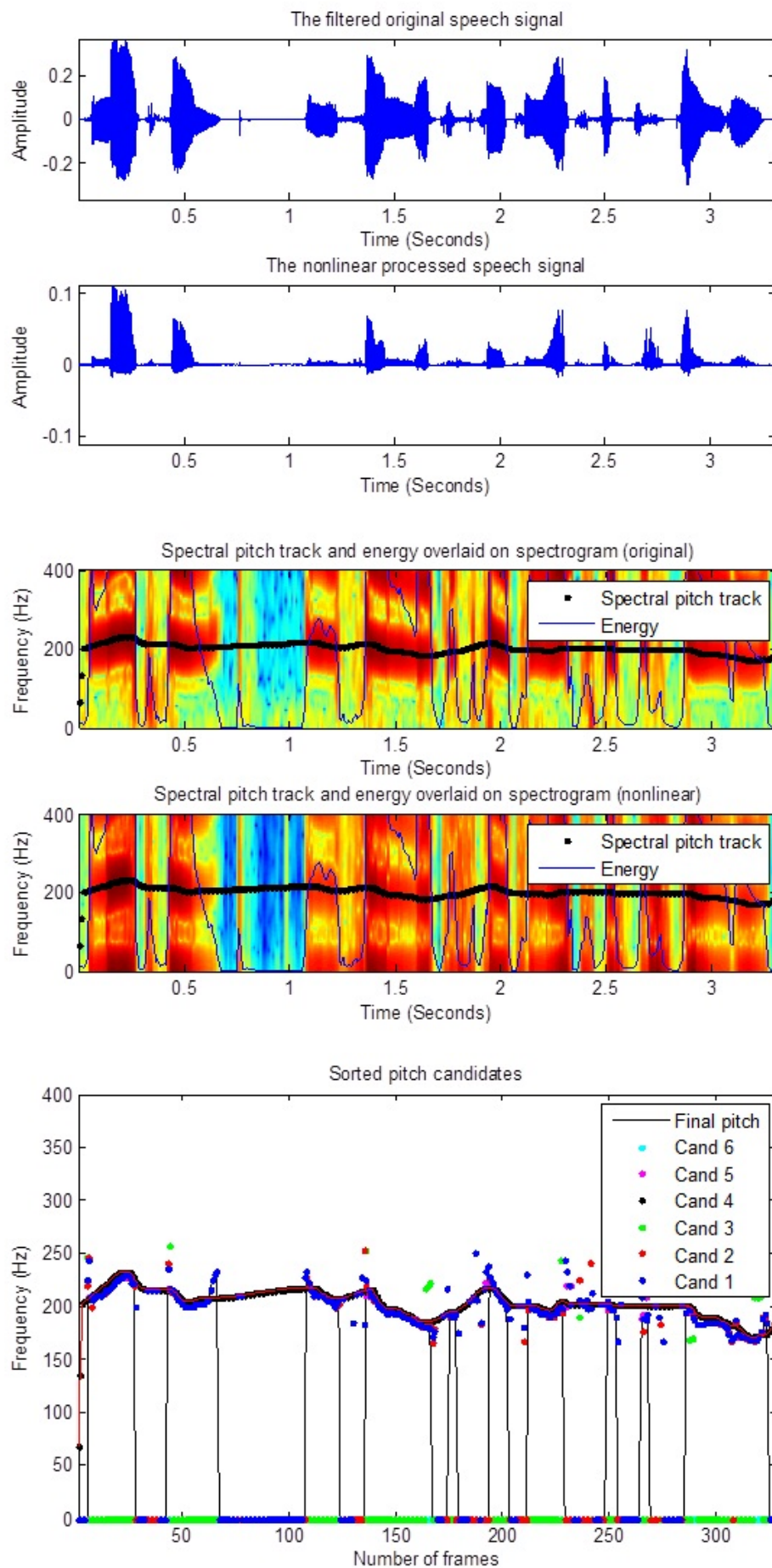
- 1) In the MATLAB command window, go the folder where the YAAPT program is located.
- 2) Read speech data from the sample/sample/flnw0000pes_short.wav file.

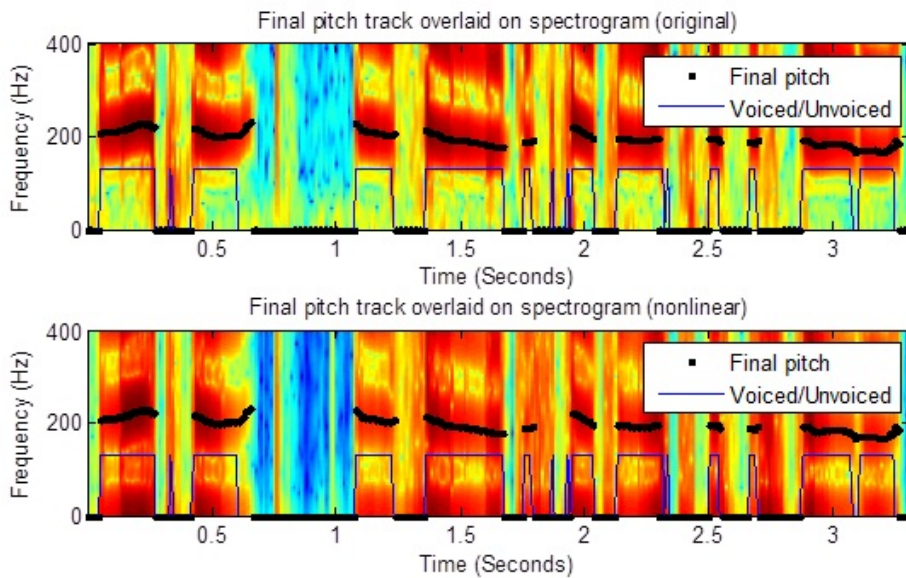
```
>> [Data, Fs] = wavread ('sample/flnw0000pes_short.wav');
```

- 3) Enable plots during pitch tracking by setting the fifth parameter (fig) to 1.

```
>> [Pitch, nf] = yaapt(Data, Fs, 1, [], 1);
```

The YAAPT creates figures to show the original speech, nonlinear processed speech, spectral pitch track, pitch candidates and final pitch track as shown below.





5. REFERENCES

1. Stephen A. Zahorian, and Hongbing Hu, "A spectral/temporal method for robust fundamental frequency tracking," J. Acoust. Soc. Am. 123(6), June 2008. [\[URL\]](#) [\[PDF\]](#)
2. Stephen A. Zahorian, Princy Dikshit, and Hongbing Hu, "A Spectral-Temporal Method for Pitch Tracking," International Conference on Spoken Language Processing, Pittsburgh, PA, Sep. 2006 [\[URL\]](#) [\[PDF\]](#) [\[Presentation\]](#)
3. Hongbing Hu, Peter Guzewich, and Stephen Zahorian, "A further comparison of fundamental frequency tracking algorithms," ASA Fall 2013 Meeting [\[Presentation\]](#)
4. Dr. Stephen A. Zahorian's Website <http://www.ws.binghamton.edu/zahorian>
5. Dr. Hongbing Hu's Binghamton Website, <http://bingweb.binghamton.edu/~hhu1/>.

6. FAQ

1. Why is the following error observed when executing wavread as shown in the EXAMPLES section With MATLAB R2012a?

```
>> [Data, Fs] = wavread ('sample/flnw0000pes_short.wav');

Error using wavread (line 164)
Too many output arguments. Output only given when using "QUERY".
```

Ans: it is due to a MATLAB R2012a issue. Try a different MATLAB version, e.g. R2009a, or please don't include folder name in the file path. Instead, copy the wav file into the current folder.

Last updated: Dec 11, 2013