Automated RFF Estimation from Accelerometer Signals
(last modified: 31st August 2020)

For questions/comments, email Matti Groll at **mgroll@bu.edu**

The original MATLAB-based automated RFF algorithm was developed for high-quality acoustic recordings of the exact stimuli described in Lien (2015) and Lien et al. (2017). The algorithm was modified to improve fundamental frequency estimation and account for differences in speech sample characteristics, described in Vojtech et al. (2019). The algorithm was then further modified to be used specifically with accelerometer signals placed on the surface of the speaker's neck. Below are the instructions for the accelerometer-based algorithm as detailed in Groll et al. (2020). These instructions will provide recommendations for acceptable changes to the algorithm; any additional changes beyond these recommendations (e.g., different stimuli combinations, lower quality sound recordings) may alter your output. Exercise caution when interpreting results after applying additional changes to this algorithm.

MATLAB Toolbox Requirements:

- Econometrics Toolbox
- Curve Fitting Toolbox
- Signal Processing Toolbox
- Statistics Toolbox

Input Signal Requirements:

- All audio files must be a series of VCV utterances with an appropriate pause between each utterance (e.g., /ufu ufu ufu/ is appropriate, but /ufufufu/ is not)
- Each audio file must have the exact same number of VCV utterances; audio files with a different number of VCV utterances should be put into different folders and run separately

Automated RFF estimation instructions:

1. Save each of the audio files to perform automated RFF estimation on within the same folder. Ensure that each of the audio file names are distinguishable, such as **participant_parameter_stimuli**.wav (e.g., "**C01_typical_afa**.wav")

   Note. Do not use "+" or "–" in the filenames.

2. Open **main_RFF.m**

3. Change lines 23–25 to indicate the minimum possible pitch (min_pitch_set), maximum possible pitch (max_pitch_set), and the number of VCV instances in a recording (maxpkn). For example, a recording containing the utterance /ufu ufu ufu/ would have a maxpkn = 3.

```
min_pitch_set = 50;     %default min pitch is 50 Hz
max_pitch_set = 400;    %default max pitch is 400 Hz
maxpkn        = 3;      %initially assume that there will be 3 VCV instances
```

4. **Input:** Change line 34 (variable "folname") to the path of your wav files; remember to include a backslash ("\") at the end of the path.

5. **Output:** Create a performance folder in the location where the RFF variables and figures should be saved. Change line 31 (variable "perform_fol") to indicate the location; remember to include a backslash ("\") at the end of the path.

Note. All automated RFF analyses will be saved in a single Excel document "aRFF.xlsx" ("aRFF" stands for "automated RFF") with separate sheets for each audio file analyzed.

6. If you made separate recording files for /afa/ /ifi/ /ufu/ (e.g., "C01_healthy_afa.wav", "C01_healthy_ifi.wav", and "C01_healthy_ufu.wav") and want to combine them into one file, set stimulicomb = 1, otherwise set stimulicomb = 0 in line 26.

   Note. When stimulicomb = 0*, the analysis of each audio file will be saved in a separate sheet within the "aRFF.xlsx" Excel file. By setting* stimulicomb = 1*, the analysis of each audio file will be saved in the same sheet of "aRFF.xlsx" in a side-by-side format.*

7. After altering these variables click **Run** in the MATLAB editor to run the automated RFF estimation algorithm. The algorithm should run to completion without any further user input.

8. Wait until the algorithm finishes running, then open the "aRFF.xlsx" Excel file to see the results of the automated RFF estimation algorithm.

Additional Notes:

RFF instances may be rejected on three occasions during automated estimation. These occasions are summarized below. For more information, see Groll et al. (2020).

1. An input file may be rejected if the signal is deemed to be too noisy to reliably estimate RFF. In this instance, RFF will not be estimated for all VCV utterances in the input file, and the algorithm will report that the input file is "Not above the threshold".

2. An input file may be rejected if the appropriate number of fricatives cannot be identified. In this instance, RFF will not be estimated for all VCV utterances in the input file, and the algorithm will report that it "Cannot identify fricative location; skipping file".

3. The RFF of an individual VCV utterance may be rejected if the resulting RFF is not physiologically valid. The rejection criteria is summarized in **main_RFF.m**, lines 61-67. The reason for rejection is shown next to each VCV utterance in "aRFF.xlsx".

Groll, M. D., Vojtech, J. M., Hablani, S., Mehta, D. D., Buckley, D. P., Noordzij, J. P., & Stepp, C. E. "Automated relative fundamental frequency algorithms for use with neck-surface accelerometer signals," *Journal of Voice*, In Press.

Lien, Y.S. "Optimization and automation of relative fundamental frequency for objective assessment of vocal hyperfunction," *Diss.* Boston University, 2015.

Lien Y.S., Heller Murray E.S., Calabrese C., Michener C.M., Van Stan J., Mehta D.D., Hillman R.E., Noordzij J.P., Stepp C.E. "Validation of an algorithm for semi-automated estimation of relative fundamental frequency," *Annals of Otology, Rhinology & Laryngology*, Vol. 126(10), pp. 712-716, 2017.

Vojtech, J.M., Segina, R.S., Buckley, D.P., Kolin, K.R., Tardif, M.C., Noordzij, J.P., & Stepp, C.E. "Refining algorithmic estimation of relative fundamental frequency: accounting for sample characteristics and fundamental frequency estimation method," *Journal of the Acoustical Society of America*, Vol. 146(5), pp. 3184-3202, 2019.