User manual: Operational Resoncance Decay (ORD) post-processing algorithm for hardenning and softenning BBC identification, in MATLAB environment

Zoltan Gabos,
Budapest University of Technology and Econmics,
Department of Applied Mechanics,
zoltan.gabos@mm.bme.hu

November 13, 2023

1 Foreword

This manual is there to guide the user of the ORD algorithm to acquire the fitted BBC from forced measurements, by perturbing the system through transient burst excitations on top of the base excitation, mentioned in the paper: https://dx.doi.org/10.2139/ssrn.4573777.

The following sections will show a step by step tutorial how to use the program.

Note that the capabilities of the program limited for no less or more than 4 perturbation cycles during measurements. Further to that, the ORD method can identify both forced and unforced cases, with proper burst harmonic perturbations. This is only the guide to understand the technical part of the post-process algorithm. For the theoretical background and the measurement techniques the reader is guided to the aforementioned article. There, a detailed explanation is given.

2 Input data

For the input data, measured or simulated displacement is strongly recommended with sufficiently high sampling frequency (in the demo, $f_s = 51.2$ kHz). Other type of time signal is also possible, but the labels on the figure are set for post-processing displacement time signals. Therefore the **recommended input data is y (m) displacement and t (s) time**. The ORD_hardenning_demo.mat file is there to demonstrate the required data structure and to give a demonstration about the algorithm.

Upon starting the program, it will ask to give the file (through windows file explorer) that contains the measured or simulated data in the recommended format.

Once the file is chosen, the calculation will start automatically.

3 ORD algorithm

The following sections will introduce the technical steps of the ORD method, created in MATLAB environment, through a provided example with example set options (which should be chosen by the user).

3.1 Cut initial transient

As the measured or simulated data can have transient oscillations at the beginning of the time signal, it can be thrown away by setting the *cut* variable to 1.

To truncate the signal by **cutting the initial transients a figure will appear**, shown in Fig. 1 a), where **one can choose the length of the transient behaviour manually**.

3.2 Transient limit identification

After cutting the initial transients, the program will use a comb filter to eliminate the apparent excitation from the signal, leaving only the burst transient behaviour for the BBC analysis. Here, one can chose the number of periods for which the program calculates the forcing frequency and the response amplitude of the periodic orbit. Based on the calculated averaged values, the comb filter can be applied. The variables $comb = 1$ and $comb_{\mathrm{n}} = 50$ plots the raw signal compared to the filtered one and sets the number of periods for averaging.

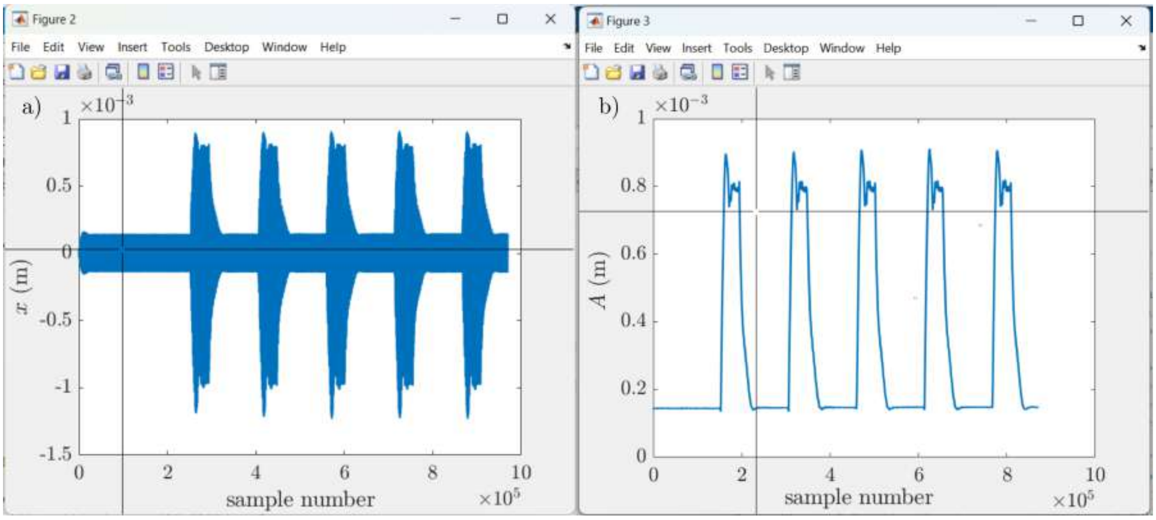

Figure 1: Panel a) shows how to pick the time instance with the vertical line, where the initial transients are settled. The raw time signal is plotted with the sample number on the x axes. Panel b) shows how to pick the minimum amplitude to find the peaks of the burst signals above this horizontal line.

Then the envelope of the signal is calculated, based on the Hilbert FIR filter length (fl = 700 in the example).

Now, a new plot will appear that requires a user input to choose a minimum amplitude vale for the peak extracting algorithm. The minimum value can be set by the horizontal line as shown in Fig. 1 b). Here, the *findpeaks()* MATLAB function is used for identify all the peaks above the limit.

Unfortunately, the *findpeaks()* function often gives more than one peaks from which we have to choose the desired ones as shown in Fig. 2.

To find the initials of the transient signals, we can pick the closest point to the desired peaks. In this program **four burst measurements are considered**, which means that one have to **pick four points closest to the assumed initials**. **Then another four points have to be picked that defines the end of the transient signals**. The found end points are plotted in Fig. 3.

Note, that there is no option for more or less burst analysis other than the default four.

3.3 Backbone curve acquisition

The backbone curve of the underlying system is acquired from the envelope of the transient section A_i (the same Hilbert FIR filter is used as before) and from the calculation of the instantaneous frequency f_i through zero crossing (ZC), https://dx.doi.org/10.2139/ssrn.4573777. The latter one can be captured through other methods as well, but the ZC can be used more generally compared to the other solutions (comparison between other solutions: https://doi.org/10.1016/j.jsv.2019.04.021).

Once, the instantaneous frequency f_i and amplitude A_i are calculated, the program plots the decaying oscillation (f_i - A_i), showing the BBC of the underlying system. As in Fig. 4 the program

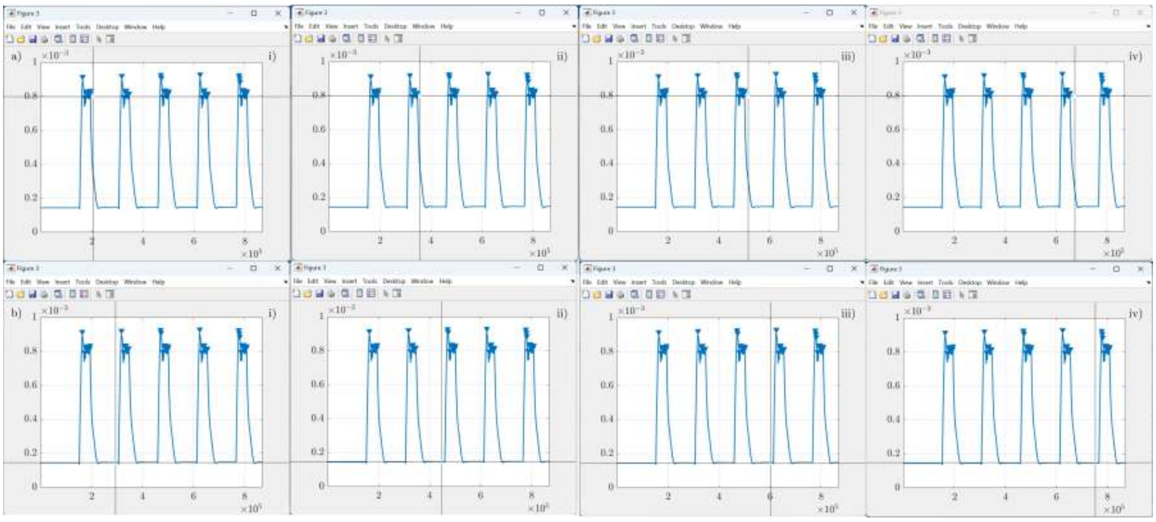

Figure 2: Row a) shows how to pick the initials of the transient decays for the four burst perturbations. Row b) show how to pick the ends of the transients.

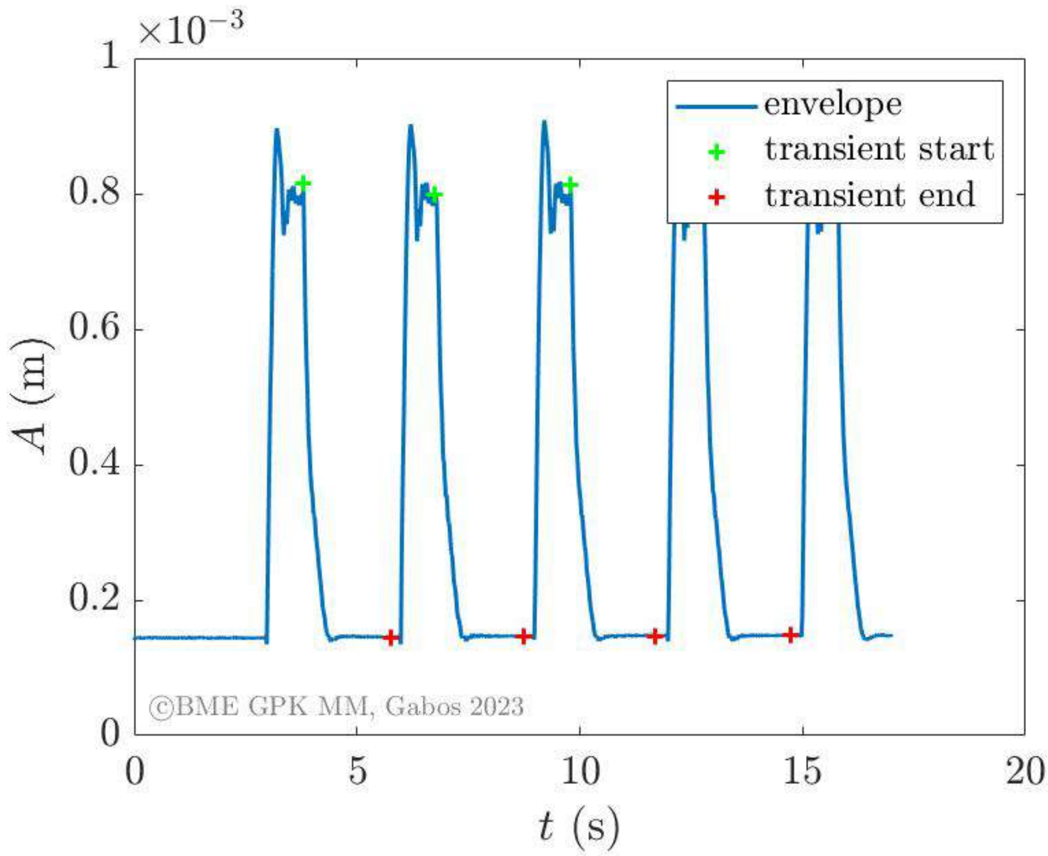

Figure 3: The program plots the envelope of the time signal along with the identified initials and ends of the transient decays.

will plot the captured BBCs one by one onto each other, where **the user has to manually pick a frequency value (via the vertical line) to cut the decay**. This is necessary due to the lack of interest about the low energy oscillations, which should be linear anyway due to the Hartman-Grobman lemma. If the low amplitude vibrations were to be considered too, there might be some inaccuracies in the curve fitting step.

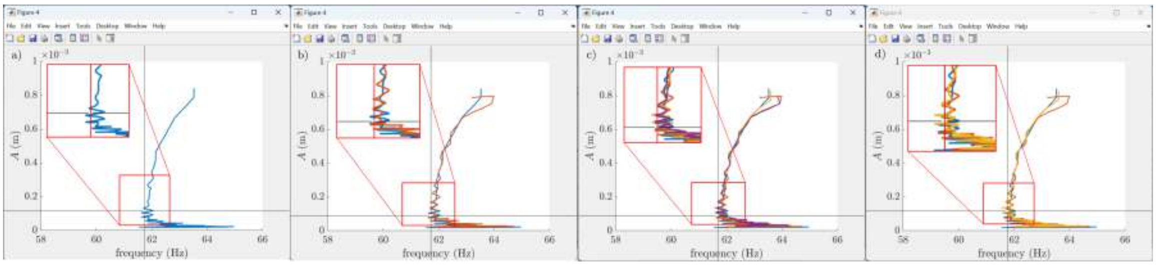

Figure 4: Plots of the decaying transient oscillations that draws out the BBC. Panles a) to d) show how to truncate the decaying signal to eliminate the low amplitude vibrations.

It is strongly advised to cut into the signal as it is highlighted in Fig. 4 for the aforementioned reasons. After picking the limits for the BBC curve fitting, every curve is estimated by the equation $A_\mathrm{i} = 2\sqrt{\frac{\omega^2-\omega_\mathrm{n}^2}{3\mu}}$ and then averaged. The μ nonlinear parameter is the averaged fit of the burst decays.

3.4 Outputs

The ORD post-processing algorithm gives the fitted and averaged nonlinear parameter μ from the equation $\ddot{x} + 2\zeta\omega_\mathrm{n}\dot{x} + \omega_\mathrm{n}^2 x + \mu x^3 = f_0\omega_\mathrm{n}^2\cos(\omega t)$, that describes the underlying nonlinear normal mode of the system dynamics. The program also gives the captured and fitted BBCs alongside with the raw and filtered time signal, shown in Fig. 5. Furthermore, the envelope of the signal with the identified initials and ends of the transients are also plotted, where the green crosses are the initials and the red ones are the ends (Fig. 3).

Note, that the nonlinear parameter μ appears both in thhe workspace and command window.

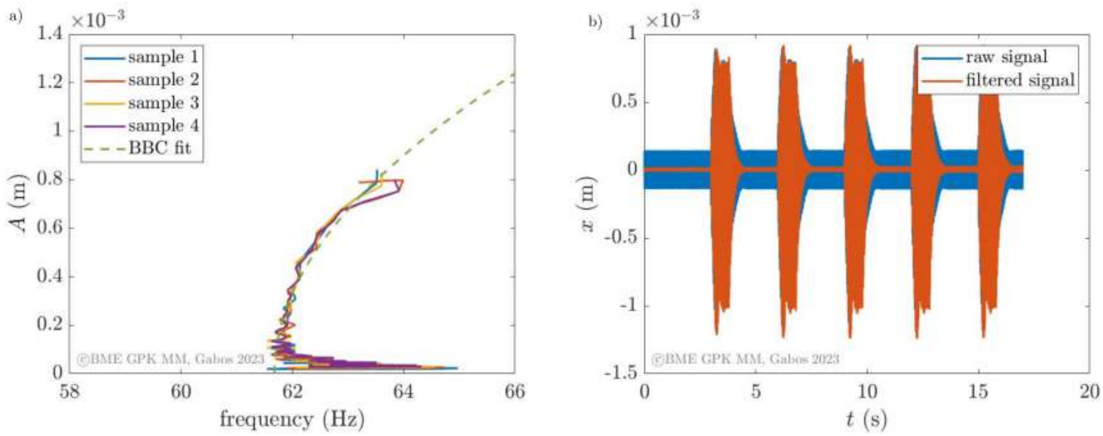

Figure 5: Resulting plots of the post-processing program. While panel a) shows the BBC curves and the averaged fit, panel b) shows the raw and filtered time signals of the measurement or simulation.