Documentation of Nonlinear_sweep_sim_and_charac_routine Matlab® script

Jérôme Juillard, 2025

Acknowledgment

This work has benefited from the financial support of the LaSIPS LabEx (ANR-10-LABX-0032-LaSIPS) managed by the French National Research Agency under the "Investissements d'avenir" program (ANR-11-IDEX-0003) within the scope of the CARAC-ATAC project.

References

Please refer to the accompanying code and to the following publications for theoretical details and experimental examples:

[1] J. Juillard et al., “MEMS resonator parameter estimation from fast frequency sweeps”, Joint Conference of the European Frequency and Time Forum and IEEE International Frequency Control Symposium, 2022, 5 pages

[2] J. Juillard et al., "Electrical characterization of MEMS gyroscopes with fast frequency sweeps", Eurosensors 2022, poster

[3] J. Juillard et al., "Efficient parameter estimation techniques for nonlinear MEMS resonators", European Frequency and Time Forum, 2024, 6 pages

[4] A. Brenes et al., "Experimental validation of a novel characterization technique for linear resonators with a large time constant", Mechanical Systems and Signal Processing, 2024

Simulation of frequency sweeps

We consider a resonator with resonance frequency f_0, angular resonance frequency $\omega_0 = 2\pi f_0$, quality factor Q, nonlinear restoring forces (Duffing coefficient γ) and damping forces (quadratic damping coefficient α), a driving force proportional to a drive voltage v_{drv} and a random acceleration noise with white spectrum $a_n(t)$. The displacement x is then governed by

$$\ddot{x} \approx -\omega_0^2(1+\gamma x^2)x - \frac{\omega_0}{Q}(1+\alpha x^2)\dot{x} + g_{drv}v_{drv} + a_n(t) \qquad (1)$$

where g_{drv} is a gain with units of $\mathrm{m.s^{-2}.V^{-1}}$.

For a given set of resonator parameters $(\omega_0, Q, g_{drv}, \gamma, \alpha)$, the code simulates a frequency sweep over bandwidth $f_{sweep} \pm \Delta f_{sweep}$, in time T_{sweep}, with constant amplitude V_{drv}, and the recovery of the motional signal through an idealized front-end with sensing gain g_{sen}, feedthrough gain g_{ft} (both complex-valued), and white detection noise $V_n(t)$. It outputs a complex voltage

$$V_{meas}(t) = g_{sen}X(t) + g_{ft}V_{drv} + V_n(t) \qquad (2)$$

where $X(t)$ is the complex amplitude of $x(t)$.

A parameter estimation procedure is then applied to this simulated data, as described further on.

Note that:

- this part of the code is <u>not</u> a transient simulation of (1), but rather a simulation of the slowly-varying complex amplitude approximation of (1)
  $$2j\Omega\dot{X} - \Omega^2 X \approx -\left(\omega_0^2\left(1+\frac{3}{4}\gamma|X|^2\right) + j\Omega\frac{\omega_0}{Q}\left(1+\frac{\alpha}{4}|X|^2\right)\right)X + g_{drv}V_{drv} \qquad (3)$$
  where Ω is the instantaneous angular frequency of the stimulus.
- this approximation itself relies mostly on two assumptions: that $Q \gg 1$, and that the sweep time be larger than the resonator response time, i.e. $T_{sweep} > 2Q/\omega_0$.

Here are a few guidelines to obtain meaningful / exploitable simulation results. Make sure:

- the slowly-varying assumption holds,
  $$Q \gg 1$$
  $$T_{sweep} > 2Q/\omega_0$$
- the resonator bandwidth is contained in the swept bandwidth[1]
  $$f_{sweep} - \Delta f_{sweep} < f_0 - \frac{f_0}{2Q} < f_0 + \frac{f_0}{2Q} < f_{sweep} + \Delta f_{sweep}$$
- the SNR at the input is large enough
  $$|g_{drv}|V_{drv} \gg |A_n|$$
  where $|A_n|$ is the rms value of the acceleration noise.
- the SNR at the output is large enough
  $$|g_{sen}X| \approx |g_{sen}|g_{drv}V_{drv}\frac{Q}{\omega_0^2} \gg |V_n|$$
  where $|V_n|$ is the rms value of the voltage noise, and the approximation is valid close to resonance, in the absence of nonlinear damping.
- nonlinear parameters contribute significantly to the resonator's response, i.e.

[1] This is an « a minima » requirement. Experimental considerations also have to be taken into account, as explained further.

$$|X| \approx g_{drv} V_{drv} \frac{Q}{\omega_0^2} \geq \frac{2}{\sqrt{3\gamma Q}}, \frac{2}{\sqrt{\alpha}}$$

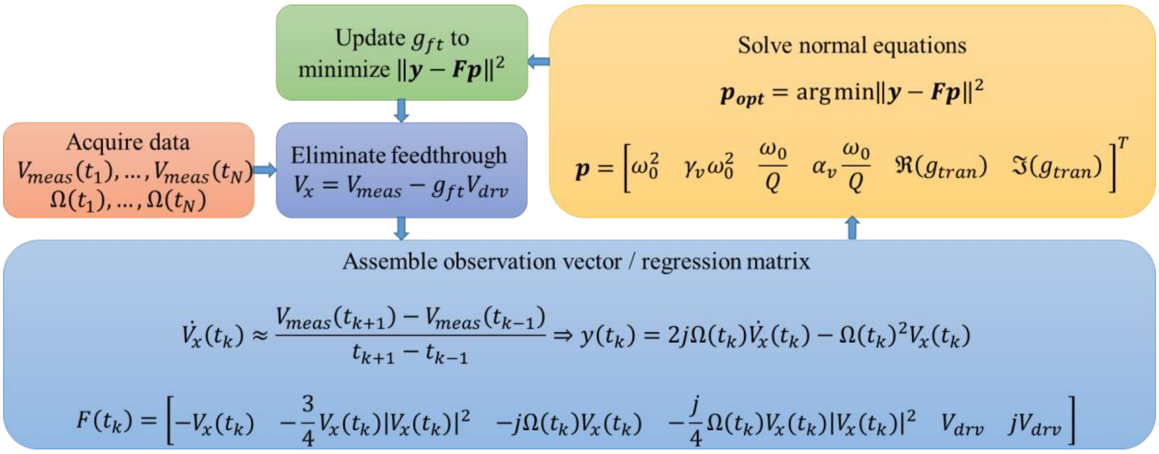
where the approximation is (again) valid close to resonance and the right-hand sides respectively correspond to the critical Duffing amplitude and the critical damping amplitude.

Matlab simulation details

The Matlab code is pretty straightforward, the only variables that require attention are:

- SNRin. This may either be:
  - Inf (no acceleration noise)
  - a numeric value equal to $|g_{drv}| V_{drv} / |A_n|$
- SNRout. This may either be:
  - Inf (no voltage noise)
  - a numeric value equal to $|g_{sen}| g_{drv} V_{drv} \frac{Q}{\omega_0^2} / |V_n|$
- Direction. This may either be:
  - 'up'. Sweeps from $f_{sweep} - \Delta f_{sweep}$ to $f_{sweep} + \Delta f_{sweep}$ in T_{sweep} seconds.
  - 'down'. Sweeps from $f_{sweep} + \Delta f_{sweep}$ to $f_{sweep} - \Delta f_{sweep}$ in T_{sweep} seconds.
  - 'updown'. Sweeps from $f_{sweep} - \Delta f_{sweep}$ to $f_{sweep} + \Delta f_{sweep}$ in $T_{sweep}/2$ seconds, from $f_{sweep} + \Delta f_{sweep}$ to $f_{sweep} - \Delta f_{sweep}$ in $T_{sweep}/2$ seconds.
  - 'downup'. Sweeps from $f_{sweep} + \Delta f_{sweep}$ to $f_{sweep} - \Delta f_{sweep}$ in $T_{sweep}/2$ seconds, from $f_{sweep} - \Delta f_{sweep}$ to $f_{sweep} + \Delta f_{sweep}$ in $T_{sweep}/2$ seconds.
- InitialState. This may either be:
  - 'zero'. The simulation starts with the resonator at rest ($X = 0$).
  - 'high' / 'low'. The simulation starts from the steady-state solution with the largest / smallest amplitude. In many instances, if starting the sweep far enough from resonance, there is only one steady-state solution.
  - a complex numeric value corresponding to a measured voltage.

  Note that it is preferable to run 'updown' or 'downup' sweeps to correctly estimate the parameters of strongly nonlinear resonators (operated above the critical Duffing amplitude).
- NStep. This is the number of steps used in the simulation. The strict conditions for the stability of the proposed predictor-corrector scheme still have to be worked out. Loosely speaking, the larger Δf_{sweep}, the larger NStep should be, in order to account for faster beat phenomena. NStep should also be at least one order of magnitude higher than the number of samples one seeks to simulate, so that the approximation error of the simulation can be neglected compared to other approximations used in the characterization routine (e.g. finite differences).



Parameter estimation procedure for frequency sweep data (from [3]).

Parameter estimation routine

The parameter estimation routine can be used on the simulated sweep data, or as a standalone routine for processing experimental data. It is agnostic of what type of stimulus is used (constant- or varying-amplitude sweeps, ringdowns, etc.), of whether data is uniformly sampled or not, etc. The user provides a “data” structure made of:

- a sequence of sampling times (data.Time),
- the corresponding frequencies (data.Freq), measured voltages (data.V_meas) and drive voltages (data.V_drv),

from which the routine estimates:

- a resonance frequency f_0 and a quality factor Q.
- a complex feedthrough gain g_{ft} and a complex transduction gain $g_{tran} = g_{sen}g_{drv}$.
- “output-referred” nonlinear coefficients $\alpha_v = \alpha/|g_{sen}|^2$ and $\gamma_v = \gamma/|g_{sen}|^2$.

The routine relies on a 6-dimensional linear least squares procedure nested inside a 2-dimensional nonlinear least squares procedure, as depicted above. The only fundamental differences between the actual code and the depicted procedure is:

- the approximation of $\dot{V}_x$. First of all, no assumption is made about a constant-valued V_{drv} so that the code can be used for varying-amplitude excitation. Moreover, no assumption is made about a uniform sampling, as this is a limitation with several experimental setups (e.g. HF2LI in “sweeper” mode). Nonetheless, the finite difference scheme used for approximating $\dot{V}_x$ has 2nd order accuracy, even with non-uniform sampling. It relies on “A simple finite-difference grid with non-constant intervals”, by Sundqvist and Veronis, in *Tellus* 1970.

- the normalization of the columns of the regression matrix. This results in a better-conditioned matrix, with reduced sensitivity to noise.

The nonlinear optimization relies on Matlab’s lsqnonlin function: convergence is extremely fast and “almost guaranteed”, even starting from a trivial guess for g_{ft}. The “slow” convergence of this nonlinear optimization either denotes poor input data or poor model choice (or both).

The following options are available:

- FTOptim
  - 'yes' or 'no', depending on whether one wants to estimate g_{ft} or not.
- FTValue
  - a complex numeric value. If FTOptim is 'yes', FTValue is used as an initial guess for g_{ft} in the nonlinear optimization routine. Otherwise, g_{ft} is assumed to be equal to FTValue and the parameter estimation boils down to linear least squares.
- DuffingOn
  - 'yes' or 'no'. If set to 'no', γ_v is assumed to be 0, otherwise it is estimated along with the other parameters.
- QuadDampingOn
  - 'yes' or 'no'. If set to 'no', α_v is assumed to be 0, otherwise it is estimated along with the other parameters.
- Display
  - 'iter' or 'none'. If set to 'iter', the convergence of the Levenberg-Marquardt nonlinear optimization routine can be monitored.

Note that if the drive voltage argument is 0 or a zero-valued vector, a ringdown is automatically detected, so that neither g_{ft} nor g_{tran} are estimated.

Experimental considerations

In an experimental context, one should keep in mind that:

- the parameters appearing in (1) and (2) depend on the operating point of the system: to first order on the resonator's bias voltage, its temperature, its "quantity of interest" (in the case of a resonant sensor). The amplitude of the drive voltage may also impact ω_0 through the voltage nonlinearity of electrostatic actuation, for example.
- equations (1-2) do not account for excitation or detection nonlinearity. Their validity in the case of resonators with gap-closing electrostatic transduction is limited to $x \ll d$, where d is the electrostatic gap. Outside of this range g_{drv}, g_{sen} and g_{ft} all become displacement-dependent and the parameter estimation routine is no longer as straightforward.
- equation (2) holds provided the sweep is performed over a limited bandwidth, across which the transfer function of the front-end can be approximated as frequency-independent. Thus, a "good" sweep bandwidth results from a tradeoff. It should be larger than the resonator bandwidth for estimating resonator parameters (e.g. feedthrough is all the better estimated as there is little motional signal), and also account for nonlinearity (both types of nonlinearities increase the resonator bandwidth). On the other hand, too large a bandwidth limits the validity of (2) – although one may possibly account for a dependence of g_{sen} on Ω – and also results in excessive measurement noise (the demodulation bandwidth must at least be equal to the swept bandwidth).

Assessing fit quality

The quality of the fit can be assessed by comparing the original parameters (in the resonator structure) to those that were estimated (in the resonator_optim structure). Of course, when experimental data is used, the "original parameters" are unknown. Still, the validity of the estimated parameters can be assessed:

- as already mentioned, by verifying that the nonlinear least squares procedure has converged fast. The IterCount variable should typically be smaller than 10.
- comparing the residue vector to the observations vector, in the optimout structure.
- comparing the sweep data used for the parameter estimation to the “retrosimulated” sweep data, i.e. the data obtained when simulating the sweep with the parameters that have been estimated. Note that, with this approach, the difference between the original data and the retrosimulated data inevitably increases over time, due to the integration of local errors.

In the code, there are 2 output plots:

- the first one compares the original simulated displacement (including input acceleration noise) to the retrosimulated displacement (assuming no input acceleration noise).
- the second one compares the original simulated voltage (including input acceleration noise, output voltage noise and feedthrough) to the retrosimulated voltage (assuming no input acceleration noise, nor output voltage noise, but accounting for the estimated feedthrough).