Simulation of microphonic effects in high Q_L TESLA cavities during CW operations.

A. Bellandi, J. Branlard, H. Schlarb, C. Schmidt, S. Pfeiffer, A. Nawaz,
W. Cichalewski, R. Rybaniec

Introduction:

XFEL and FLASH are the two main superconductive LINACs at DESY used for producing short-wave X-ray laser light through the FEL process. These machine are currently used in pulse mode and they produce a burst of short spaced bunches every 10Hz. Because of the interest of relaxing the spacing between bunches there are proposals to turn XFEL and FLASH in Continuos wave (CW) machines. In such machines the accelerating gradient is always present and an arbitrary long train of particle can be accelerated. In order to do that the loaded quality factor (QL) of the cavities has to be increased to keep the power and thermal requirements within reasonable limits. Also for such machines there is the need to keep the error on amplitude and phase of the RF field below 0.01% and 0.01°

Issues:

At the moment tests for CW operations on cavity modules are performed at DESY, but varying QL above the nominal tuning range is a time consuming process and verifying the controller behaviour under beam loading is impossible at our current test stand.

A code to simulate the cavity module with different QL and beam loading can be useful to speed-up the optimization of the controller.

Code Requirements:

- Use of IQ parameters to describe input and output parameters of the cavity signals.
- Simulation of multiple microphonic sources
- Simulation of multiple mechanical resonances that are excited by LFD and can couple with microphonic sources
- Simulation of beam loading
- Reasonable speed: the simulation time is in the order of 1 s, with a sampling speed of 9 MHz so there are some constraints on how the code is written.
- Easily modifiable controller code without impacting on the performance.

Simulated system:

Using envelope description of the cavity dynamics with the mechanical resonances[1] and microphonic sources a *canonical form* nonlinear ODE can be written:

$$
\begin{bmatrix} \dot{V}_I \\ \dot{V}_Q \\ \Delta\dot{\omega}_{mech} \\ \Delta\dot{\omega}_1 \\ \Delta\dot{\omega}_2 \\ \vdots \\ \Delta\dot{\omega}_N \\ \Delta\ddot{\omega}_1 \\ \Delta\ddot{\omega}_2 \\ \vdots \\ \Delta\ddot{\omega}_N \end{bmatrix} = \begin{bmatrix} \omega_{1/2}V_I + ((2\pi k_{static})|V|^2 - \Delta\omega_{micro}(t) - \Delta\omega_{mech})V_Q + \frac{R}{Q}\omega_0 I_I(t) \\ \omega_{1/2}V_Q - ((2\pi k_{static})|V|^2 - \Delta\omega_{micro}(t) - \Delta\omega_{mech})V_I + \frac{R}{Q}\omega_0 I_Q(t) \\ \sum_{i=1}^{N}\Delta\dot{\omega}_i \\ \Delta\dot{\omega}_1 \\ \Delta\dot{\omega}_2 \\ \vdots \\ \Delta\dot{\omega}_N \\ -\frac{(\omega_0^{mech})_1}{Q_1^{mech}}\Delta\dot{\omega}_1 - (\omega_0^{mech})_1^2\Delta\omega_1 - 2\pi(\omega_0^{mech})_1^2 k_1|V|^2 \\ -\frac{(\omega_0^{mech})_2}{Q_2^{mech}}\Delta\dot{\omega}_2 - (\omega_0^{mech})_2^2\Delta\omega_2 - 2\pi(\omega_0^{mech})_2^2 k_2|V|^2 \\ \vdots \\ -\frac{(\omega_0^{mech})_N}{Q_N^{mech}}\Delta\dot{\omega}_N - (\omega_0^{mech})_N^2\Delta\omega_N - 2\pi(\omega_0^{mech})_N^2 k_2|V|^2 \end{bmatrix}
$$

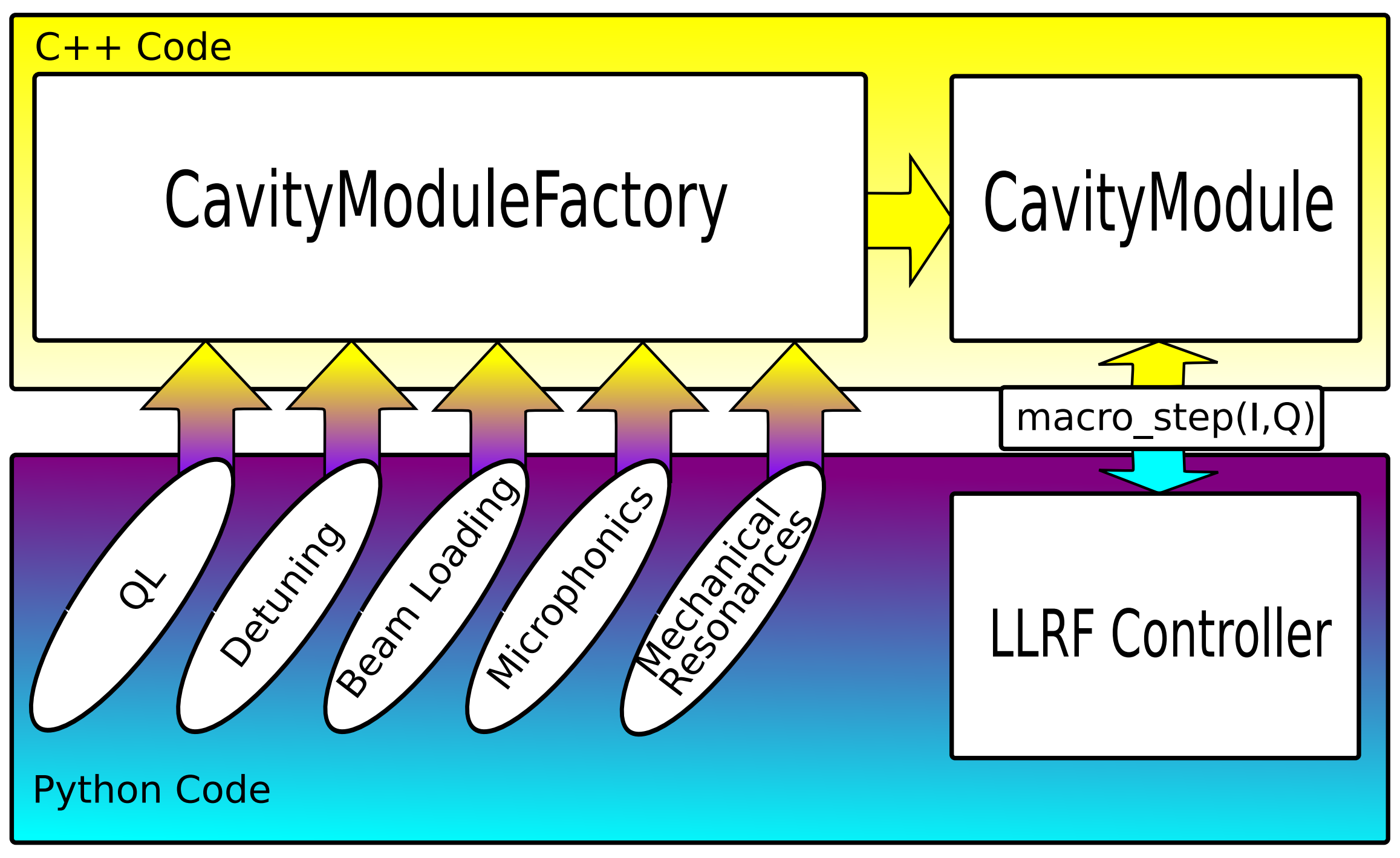
Where:

$$
\Delta\omega_{micro}(t) = \sum_{i=1}^{M}(\Delta\omega_{ampdetuning})_i \sin\left((\omega_{micro})_i t + (\phi_{micro})_i\right)
$$

This system can be used in conjunction of *Boost C++ OdeInt* package[2] to make a step simulator

Code Implementation:

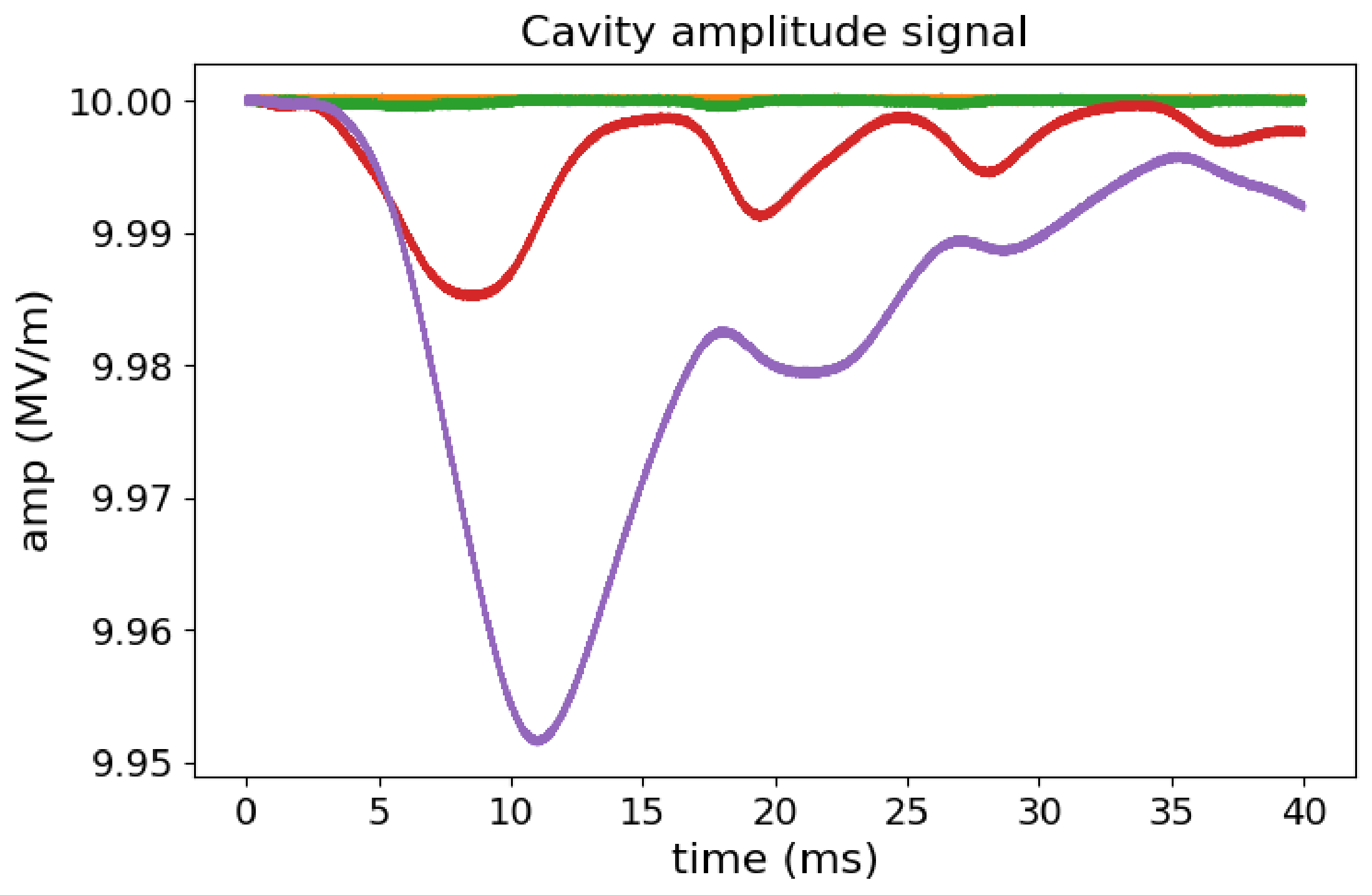
In order to meet the requirements it was decided to make the simulation code as a mixed C++/python library :

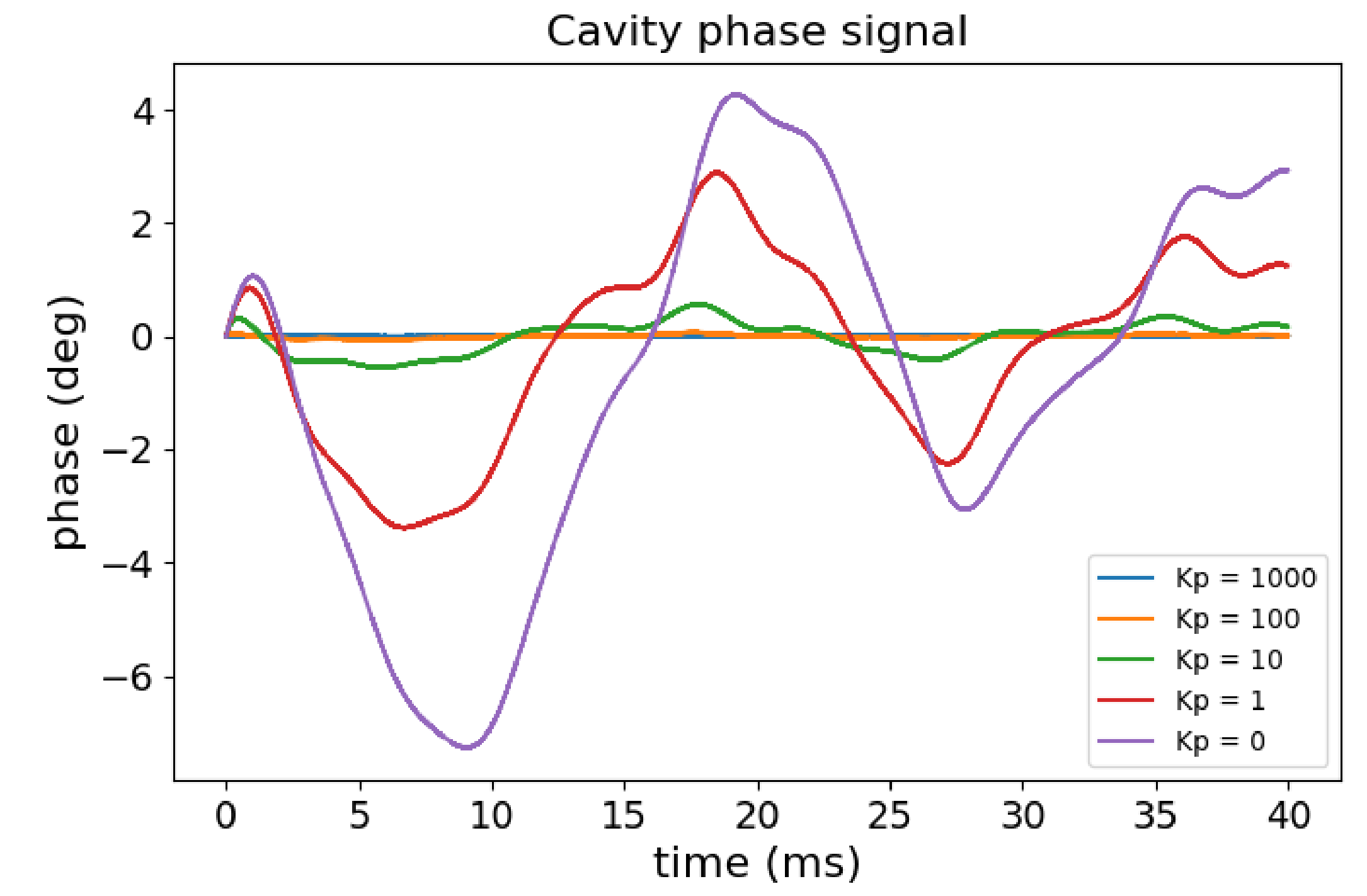


From the python code the properties of a 'Factory' object are set. Then a 'Module' object is created from the factory and used to simulate a cavity module.

The advantage of such a solution is that the controller part is written and easily modifiable in python whereas the cavity code is fixed and written in C++

Comparison between simulated proportional controllers:

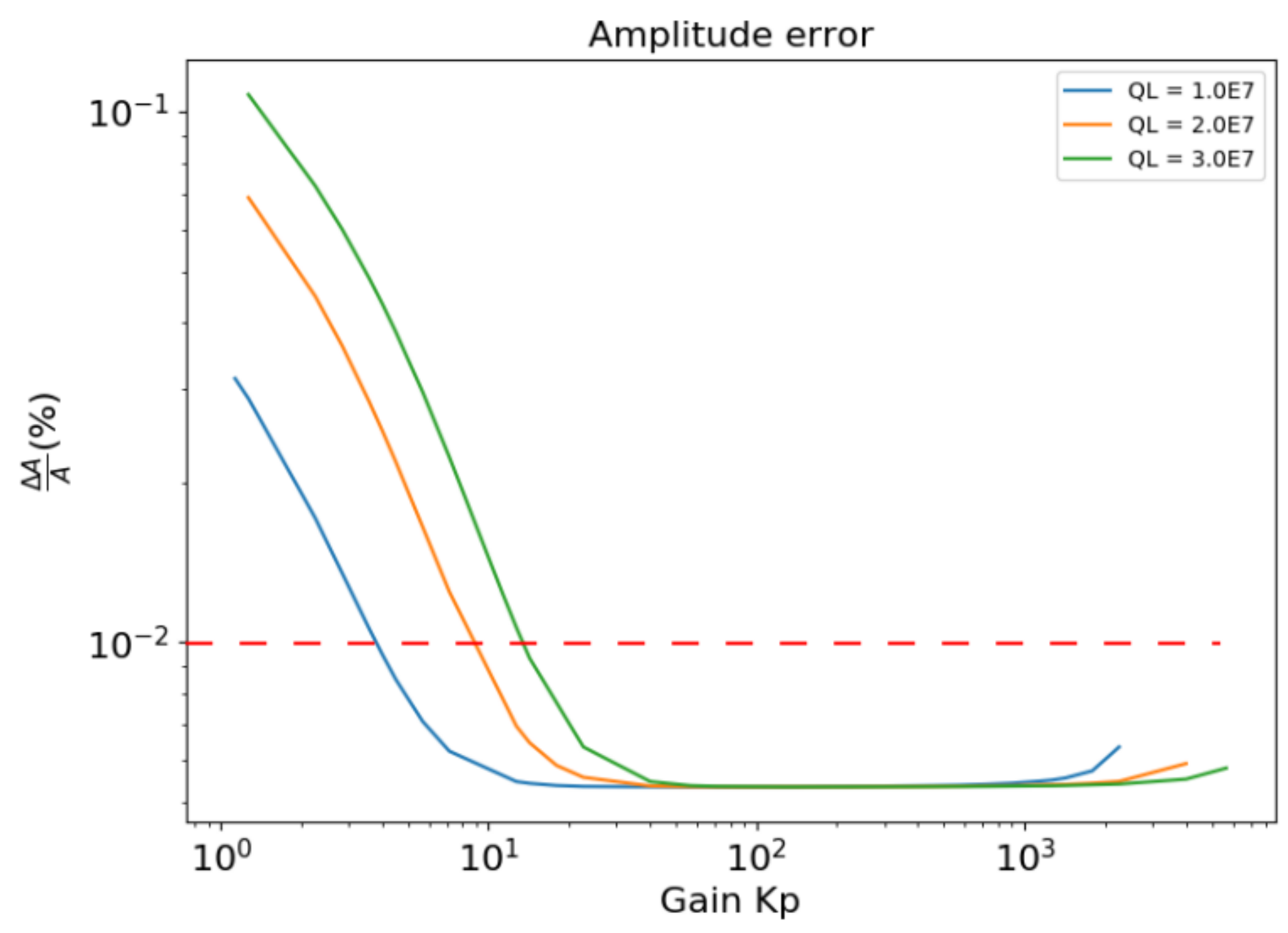


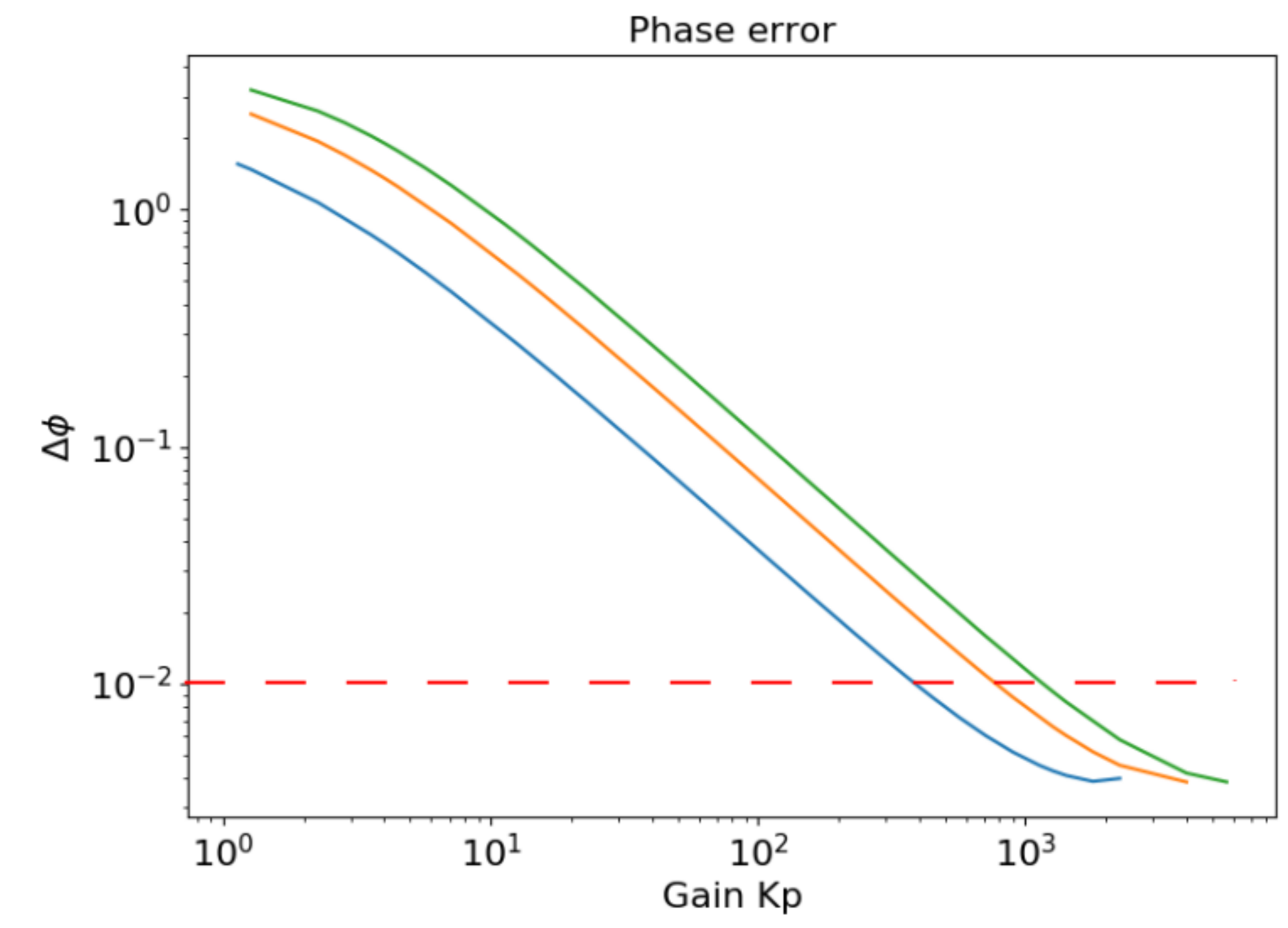


The graphs show simulated amplitude and phase data for a superconductive cavity for various value of the proportional controller. The step (sampling) time used is 9 MHz with 1.55 µs of loop delay. The QL is equal to 2•10^7 and the accelerating gradient is equal to 10MV/m. For the mechanical resonances the HOBICAT[3] parameters were used. For the microphonics two frequencies measured at DESY in CW tests were used:

- f_{micro1}: 49Hz with 5.0Hz amplitude detuning
- f_{micro2}: 31Hz with 1.25 Hz amplitude detuning

Simulated field errors in function of the loop gain:





In the graphs the amplitude and phase errors in function of the proportional gain are shown. The simulated systems are equal except for the QL. The systems were simulated for 1s.

Possible further Improvements:

- Make tests on more complex feedback (ex. PI, MIMO) to find the best algorithm to minimize the amplitude and phase errors
- Simulate the piezo tuner through detuning control in the simulation.
- Add the possibility to simulate multiple bunch trains.
- Make MATLAB bindings to the C++ core to allow the use of preexisting code.

Final words:

The code is completed and is able to run on DESY computing infrastructure.
Next simulations will take the piezo tuner and and beam loading compensation in account.

The code is available in the following github link:
https://github.com/Bellaz/ModuleSimulator

Presenter:

Andrea Bellandi
DESY
22607 Hamburg
andrea.bellandi@desy.de

References:

- [1] Liepe, M., W. D. Moeller, and Stefan N. Simrock. "Dynamic Lorentz force compensation with a fast piezoelectric tuner." Particle Accelerator Conference, 2001.
- [2] Ahnert, K., & Mulansky, M. (2011, September). Odeint–solving ordinary differential equations in C++
- [3] Neumann, Axel. Compensating microphonics in SRF Cavities to ensure beam stability for future Free-Electron-Lasers. Diss. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2008.