

CRYMODULE-ON-CHIP SIMULATION ENGINE*

C. Serrano[†], L. Doolittle, V. K. Vytla, LBNL, Berkeley, CA 94720, USA

Abstract

The Cryomodule-On-Chip (CMOC) simulation engine is a Verilog implementation of a cryomodule model used for Low-Level RF development (LLRF) for superconducting cavities. The model includes a state-space model of the accelerating fields inside a cavity, the mechanical resonances inside a cryomodule as well as their interactions. The implementation of the model along with the LLRF controller in the same FPGA allows for live simulations of an RF system. This allows for an interactive simulation framework, where emulated cavity signals are produced at the same rate as in a real system and therefore providing the opportunity to observe longer time-scale effects than in software simulations as well as a platform for software development and operator training.

INTRODUCTION

The CMOC simulation engine has been used throughout the design process of a SRF cavity controller, first for the Next Generation Light Source (NGLS) proposal at LBNL [1] and then for the Linear Coherent Light Source Linac (LCLS-II) [2]. LCLS-II is an X-ray Free Electron Laser (FEL) under construction at SLAC, driven by a superconducting RF Linac [2]. The electron beam quality will directly translate to the quality of the X-ray beams produced in undulators and used for scientific research in the end stations; hence strict requirements have been placed on the stability of the accelerating cavity fields. An initial stability goal of 0.01° in phase and 0.01% amplitude has been set for the main Linac, composed of 280 nine-cell 1300 MHz superconducting cavities [3].

The difficulty resides in providing the ability to reject disturbances from the cryomodule, which is incompletely known as it depends on the cryomodule structure itself (currently under development at JLab and Fermilab) and the harsh accelerator environment. Previous experience in the field and an extrapolation to the cavity design parameters (relatively high $Q_L \approx 4 \times 10^7$, implying a half-bandwidth of around 16 Hz) suggest the use of strong RF feedback to reject the projected noise disturbances, which in turn demands careful engineering of the entire system.

MOTIVATION

LBNL has developed tools to first perform analytical stability analysis using basic control theory, software [4] for the numerical analysis and CMOC. CMOC takes the design of a SRF cavity controller one step closer to the final result, which is its implementation in an FPGA. It runs the actual

cavity controller in a live FPGA and provides the capability of exercising the functionality of the controller as it interacts with a SRF cavity model, also implemented in the FPGA.

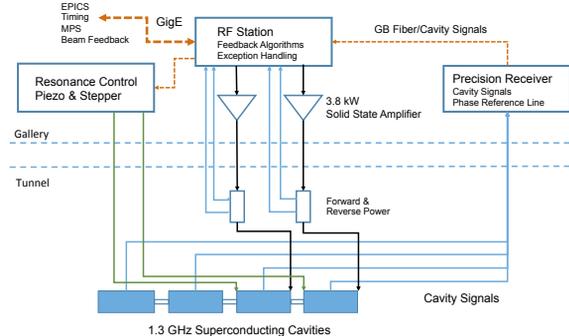


Figure 1: System hardware configuration supporting half of a cryomodule (one of two RF Station chassis shown).

Waveform display and user interaction with CMOC demands for communication logic with software, which is included in CMOC to extract cavity waveforms and to set configuration parameters for both the cavity controller and the models. This communication link is the same used in operations and can be used to build the software infrastructure for the LLRF installation. The physically meaningful live waveforms being generated by CMOC and its ability to provide the user with the same level of interactivity as when operating a real cavity make it useful for several purposes: initial tests of the cavity controller for FPGA designers, a complete environment to exercise the communications between the FPGA and a control system (both on the FPGA and control system ends) and as a training platform for students, non-LLRF experts and operators.

Figure 1 shows a the hardware configuration supporting half of a cryomodule (one of two RF Station chassis shown) for the LCLS-II LLRF system. CMOC includes provisions to include the cavity controller (in the RF Station), a model of the RF plant (RF amplifier saturation and cavity models), the cavity probe sensing through the Precision Receiver and a mechanical model of the cryomodule as well as its interactions with the cavities' electrical properties through ponderomotive forces. The resonance control logic is not integrated in CMOC at the moment but an emulation of a piezo actuator is possible from software for cavity tuning purposes.

CAVITY MODEL

The cavity model responds to a multi-cell cavity structure, with couplings to the RF source, a cavity field probe and the beam. The cavity field is decomposed in its eigenmodes, where each mode is represented by the traditional RLC par-

* This material is based upon work supported by the U.S. Department of Energy, Office of Science, under Contract No. DE-AC02-76SF00515.

[†] CSerrano@lbl.gov

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

allel circuit, coupled to two current sources representing the High Power Amplifier (HPA) and the beam respectively.

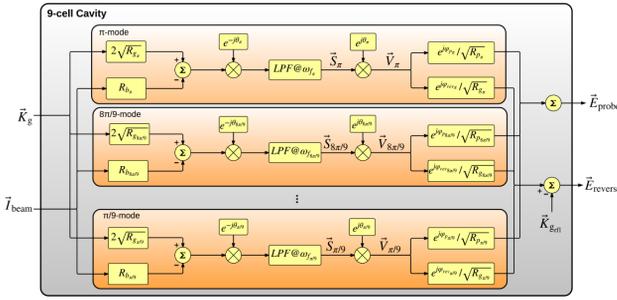


Figure 2: Software implementation of the cavity model.

Ideally, we would like to measure the EM fields from each mode present in the cavity in order to control them appropriately. However, the best we can do is measure the overall field in the cavity, designated here by \vec{E}_{probe} . It is measured in practice using a probe antenna, and is theoretically given by

$$\vec{E}_{probe} = \sum_{\mu} \vec{V}_{\mu} / \sqrt{Q_{p\mu} (R/Q)_{\mu}} \quad (1)$$

where \vec{V}_{μ} is a representative measure of the energy stored in each electrical eigenmode μ , designated as mode cavity voltage, and where $Q_{p\mu} (R/Q)_{\mu}$ is the coupling impedance of the probe port for that mode.

Alternatively, the expression for reverse (*a.k.a.* reflected) wave traveling outward from the fundamental port includes a prompt reflection term, yielding

$$\vec{E}_{reverse} = \sum_{\mu} \vec{V}_{\mu} / \sqrt{Q_{g\mu} (R/Q)_{\mu}} - \vec{K}_g \quad (2)$$

where $Q_{g\mu} (R/Q)_{\mu}$ is the coupling impedance of the drive port of mode μ .

Equations 1 and 2 express the measured probe and reverse fields as a function of eigenmode voltages (\vec{V}_{μ}) and their respective port couplings, as illustrated in Figure 2. The state-variable equation governing the accelerating voltage for each eigenmode at baseband is formulated next.

Electromagnetic Eigenmode

A multi-cell cavity is represented by a series of coupled resonators (one per cell in the cavity), each represented by an RLC circuit [5]. Decomposing the EM cavity fields into eigenmodes and applying the principle of superposition we obtain the representation expressed by equation 1 [6]. The equivalent circuit used to represent one cavity eigenmode is shown in Fig. 3, where each mode's accelerating voltage is added in order to obtain the cavity overall accelerating voltage, as deduced from Eq. 1. Each mode has its own value of \vec{V} , (R/Q) , Q_x , and other characteristics that will be introduced later.

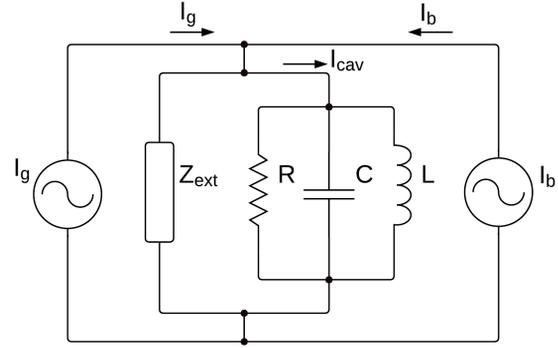


Figure 3: Electromagnetic eigenmode equivalent circuit.

If we apply Kirchoff's current law to the mode's RLC equivalent circuit (see figure 3, μ refers to a particular eigenmode), we get:

$$\vec{I}_{\mu} = \vec{I}_{C_{\mu}} + \vec{I}_{R_{\mu}} + \vec{I}_{L_{\mu}} \quad (3)$$

where:

$$\frac{d\vec{I}_{C_{\mu}}}{dt} = C_{\mu} \cdot \frac{d^2\vec{V}_{\mu}}{dt^2}, \quad (4)$$

$$\frac{d\vec{I}_{R_{\mu}}}{dt} = \frac{1}{R_{L_{\mu}}} \frac{d\vec{V}_{\mu}}{dt} \quad (5)$$

$$\text{and } \frac{d\vec{I}_{L_{\mu}}}{dt} = \vec{V}_{\mu} / L_{\mu} \quad (6)$$

Differentiating both sides of equation 3 and substituting using Eq. 6, the full vector (complex) differential equation for the cavity accelerating voltage \vec{V}_{μ} can be written as:

$$\frac{d^2\vec{V}_{\mu}}{dt^2} + \frac{1}{R_{L_{\mu}} C_{\mu}} \frac{d\vec{V}_{\mu}}{dt} + \frac{1}{L_{\mu} C_{\mu}} \vec{V}_{\mu} = \frac{1}{C_{\mu}} \frac{d\vec{I}_{\mu}}{dt} \quad (7)$$

which can be expressed as a function of the mode's nominal resonance frequency $\omega_{0\mu}$ ($1/L_{\mu} C_{\mu} = \omega_{0\mu}^2$) and loaded Q ($1/R_{L_{\mu}} C_{\mu} = \omega_{0\mu} / Q_{L_{\mu}}$):

$$\frac{d^2\vec{V}_{\mu}}{dt^2} + \frac{\omega_{0\mu}}{Q_{L_{\mu}}} \frac{d\vec{V}_{\mu}}{dt} + \omega_{0\mu}^2 \vec{V}_{\mu} = \frac{\omega_{0\mu}^2 R_{L_{\mu}}}{Q_{L_{\mu}}} \frac{d\vec{I}_{\mu}}{dt} \quad (8)$$

Taking the slowly varying envelope approximation [7] ($\omega_{f\mu} \ll \omega_{0\mu}$), separating voltage and current into real and imaginary parts, assuming that the detune frequency varies slowly with respect to the carrier frequency ($\omega_{d\mu} \ll \omega_{0\mu}$) and that $Q_{L_{\mu}} \gg 1$, we can reduce the order of equation 8 (a second-order band-pass filter centered at the resonance frequency) to a first-order low-pass filter at baseband [8]:

$$\left(1 - j \frac{\omega_{d\mu}}{\omega_{f\mu}}\right) \vec{V}_{\mu} + \frac{1}{\omega_{f\mu}} \frac{d\vec{V}_{\mu}}{dt} = R_{L_{\mu}} \vec{I}_{\mu} \quad (9)$$

where $\omega_{f\mu} = \omega_{0\mu} / 2Q_{L_{\mu}}$ is the mode's bandwidth and $\omega_{d\mu} = 2\pi\Delta f_{\mu}$ is the (time varying) detune frequency, given

as $\omega_{d\mu} = \omega_{0\mu} - \omega_{ref}$, i.e., the difference between actual eigenmode frequency $\omega_{0\mu}$ and the accelerator's time base ω_{ref} .

Transposing the cavity drive term into a combination of the RF source incident wave and beam loading (opposite sign indicating energy absorption by the beam), we can express Eq. 9 as:

$$\left(1 - j \frac{\omega_{d\mu}}{\omega_{f\mu}}\right) \vec{V}_\mu + \frac{1}{\omega_{f\mu}} \frac{d\vec{V}_\mu}{dt} = 2\vec{K}_g \sqrt{R_{g\mu}} - R_{b\mu} \vec{I}_{beam} \quad (10)$$

where \vec{K}_g is the incident wave amplitude in $\sqrt{\text{Watts}}$, $R_{g\mu} = Q_{g\mu} (R/Q)_\mu$ is the coupling impedance of the drive port, \vec{I}_{beam} is the beam current, and $R_{b\mu} = Q_{L\mu} (R/Q)_\mu$ is the coupling impedance to the beam. ω_f is the mode's bandwidth in rad/s and the $\frac{\omega_{d\mu}}{\omega_f}$ term (the imaginary component of the mode's pole at baseband) represents detuning.

The overall $Q_{L\mu}$ is given as $1/Q_{L\mu} = 1/Q_{0\mu} + 1/Q_{g\mu} + 1/Q_{p\mu}$, where $1/Q_{0\mu}$ represents losses to the cavity walls, $1/Q_{g\mu}$ represents coupling to the input coupler, and $1/Q_{p\mu}$ represents coupling to the field probe. $(R/Q)_\mu$ is the shunt impedance of the mode in Ohms, a pure geometry term computable for each particular eigenmode using E&M codes like Superfish. Physically, shunt impedance relates a mode's stored energy U_μ to the accelerating voltage it produces, according to

$$U_\mu = \frac{V_\mu^2}{(R/Q)_\mu \omega_{0\mu}} \quad (11)$$

The only assumptions in the above formulation are that the cavity losses are purely resistive, and thus expressible with a fixed $Q_{0\mu}$, and that no power is launched into the cavity from the field probe. If other ports have incoming power, there would be additional terms of the same form as $2\vec{K}_g \sqrt{R_g}$.

The $\frac{\omega_{d\mu}}{\omega_f}$ term in 10 (the imaginary component of the cavity pole at baseband) represents detuning. In software or hardware implementations, we can alternatively modulate that term with Lorentz perturbations, or use a purely real pole (ω_f) and modulate the frequency of the drive term. We prefer the latter, more convenient in computational terms. We then define a vector \vec{S}_μ such that:

$$\vec{V}_\mu = \vec{S}_\mu e^{j\theta_\mu} \quad \text{and} \quad \frac{d\theta_\mu}{dt} = \omega_{d\mu} \quad (12)$$

yielding:

$$\left(1 - j \frac{\omega_{d\mu}}{\omega_{f\mu}}\right) \vec{S}_\mu e^{j\theta_\mu} + \frac{1}{\omega_{f\mu}} \left(\frac{d\vec{S}_\mu}{dt} e^{j\theta_\mu} + \vec{S}_\mu \cdot j\omega_{d\mu} e^{j\theta_\mu}\right) =$$

$$= 2\vec{K}_g \sqrt{R_{g\mu}} - R_{b\mu} \vec{I}_{beam} \quad (14)$$

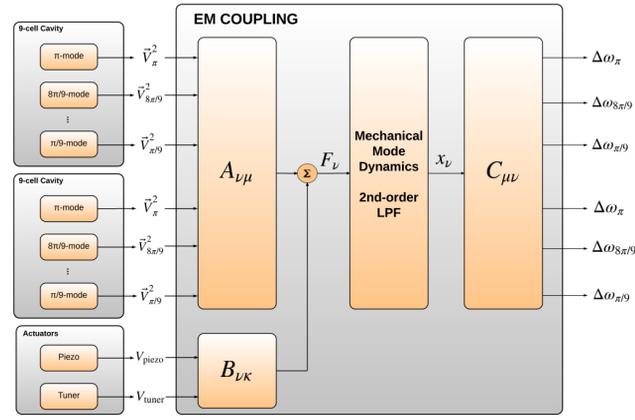


Figure 4: Electro-mechanical coupling block diagram.

The governing equation for the mode's accelerating voltage can thus be written as a set of two first order differential equations (Eq. 15):

$$\frac{d\vec{S}_\mu}{dt} = -\omega_{f\mu} \vec{S}_\mu + \omega_{f\mu} e^{-j\theta_\mu} \left(2\vec{K}_g \sqrt{R_{g\mu}} - R_{b\mu} \vec{I}_{beam}\right) \quad (15)$$

Note that this state-variable equation is a pure low-pass filter, an advantage especially in the FPGA implementation.

CRYOMODULE

Here we describe the state-space model representing the dynamics of the mechanical resonances (also, as in the case of the electrical modes in a cavity, decomposed into eigenmodes) as well as the interactions between these mechanical eigenmodes and the cavity electrical eigenmodes (through Lorentz forces), piezos and tuners.

Electro-mechanical Interactions

The presence of an EM field inside the cavity generates forces on the cavity walls, resulting in deformation of the cavity and subsequently in a shift of the cavity resonance frequency [9], designated as detune frequency $\omega_{d\mu}$. Each mode's fields generate a force proportional to $V_\mu^2 = |\vec{V}_\mu|^2$, and mechanical displacements influence each mode's instantaneous detune frequency. Construct ω_{d0} as a baseline ω_{d0} from the electrical mode solution (e.g., $-2\pi(800 \text{ kHz})$ for the TTF cavity's $8\pi/9$ mode), plus a perturbation ω_μ contributed from the mechanical mode deflections.

Consider the electrical mode index μ to include not only electrical eigenmodes of one cavity, but modes of all cavities in the mechanical assembly (e.g., cryomodule). Also include the dependence on piezoelectric actuator voltages V_k . Then if the assembly's mechanical eigenmodes are indexed by ν , mechanical forces F_ν and displacements x_ν of those eigenmodes are related to the electrical system by

$$F_\nu = \sum_\mu A_{\nu\mu} V_\mu^2 + \sum_k B_{\nu k} V_k \quad (16)$$

$$\omega_\mu = \sum_\nu C_{\mu\nu} x_\nu \quad (17)$$

where A , B , and C are constant matrices.

These matrix calculations are represented in Fig. 4. Note that in the same way any electrical eigenmode can be coupled to any mechanical eigenmode, one can configure the matrix to define couplings only present for intra-cavity interactions. We have described in detail how to solve for the accelerating voltages for each electrical mode independently, and the translation between Lorentz forces (F_ν) and mechanical displacements (x_ν) is represented by the state-space model of the mechanical eigenmode described next.

Mechanical Eigenmodes

Equations 16 and 17 are understood to apply at every time instant; the quantities V , F , x , and ω all vary with time. The differential equation governing the dynamics of each mechanical eigenmode is that of a textbook second order low-pass filter. In Laplace form,

$$k_\nu x_\nu = \frac{F_\nu}{1 + \frac{1}{Q_\nu} \frac{s}{\omega_\nu} + \left(\frac{s}{\omega_\nu}\right)^2} \quad (18)$$

where k_ν is the spring constant. For computational purposes, we want it expressed in terms of the state-space formulation

$$\frac{d}{dt} \begin{pmatrix} x_\nu \\ y_\nu \end{pmatrix} = \begin{pmatrix} a_\nu & -b_\nu \\ b_\nu & a_\nu \end{pmatrix} \begin{pmatrix} x_\nu \\ y_\nu \end{pmatrix} + c_\nu \begin{pmatrix} 0 \\ F_\nu \end{pmatrix} \quad (19)$$

where a scaled velocity coordinate y_ν has been introduced. Convert the latter equation to Laplace form and solve to get

$$\begin{pmatrix} x_\nu \\ y_\nu \end{pmatrix} = \begin{pmatrix} a_\nu - s & -b_\nu \\ b_\nu & a_\nu - s \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 \\ F_\nu \end{pmatrix} \quad (20)$$

Analytically invert that 2×2 matrix, and multiply out to get

$$x_\nu = \frac{-b_\nu c_\nu F_\nu}{(a_\nu - s)^2 + b_\nu^2} \quad (21)$$

Equate coefficients with the earlier low-pass filter form, in the case $Q > \frac{1}{2}$, to get

$$a_\nu \pm j b_\nu = \omega_\nu \left(\frac{-1}{2Q_\nu} \pm j \sqrt{1 - \frac{1}{4Q_\nu^2}} \right) \quad (22)$$

$$c_\nu = -\frac{1}{k_\nu} \cdot \frac{a_\nu^2 + b_\nu^2}{b_\nu} = -\frac{\omega_\nu^2}{k_\nu b_\nu} \quad (23)$$

A deeper understanding of the forces and responses of a single electrical eigenmode μ of the cavity comes from Slater's perturbation theory. For an eigenmode solution $\vec{H}_\mu(\vec{r}) \sin(\omega_{0\mu} t)$, $\vec{E}_\mu(\vec{r}) \cos(\omega_{0\mu} t)$ to Maxwell's equations in a closed conducting cavity (volume V), the stored energy U_μ is given by

$$U_\mu = \int_V \left[\frac{\mu_0}{4} H_\mu^2(\vec{r}) + \frac{\epsilon_0}{4} E_\mu^2(\vec{r}) \right] dv \quad (24)$$

Suppose a mechanical eigenmode ν involves small deflections $x_\nu \cdot \vec{\xi}(\vec{r})$, where x_ν gives the amount of deflection, and the dimensionless quantity $\xi(\vec{r})$ represents the mode shape. Both the force on the mode and the response to a deflection x_ν are given in terms of the Slater integral

$$F_\mu = \int_S \left[\frac{\mu_0}{4} H^2(\vec{r}) - \frac{\epsilon_0}{4} E^2(\vec{r}) \right] \vec{n}(\vec{r}) \cdot \vec{\xi}(\vec{r}) dS \quad (25)$$

where $\vec{n}(\vec{r})$ is the normal vector to the cavity surface S , and F_μ directly gives the force. Note in particular the subtraction of E and H terms, contrasted with the addition in the energy integral. Also notice the dot product of the deflection shape with the surface normal. Then

$$\Delta\omega_\mu = -x_\nu \omega_{0\mu} \left(\frac{F}{U} \right)_\mu \quad (26)$$

and

$$F_\mu = \left(\frac{F}{U} \right)_\mu \cdot \frac{1}{(R/Q)_\mu \omega_{0\mu}} V_\mu^2 \quad (27)$$

where $(F/U)_\mu$ is a property of the electrical eigenmode, independent of amplitude, with units of m^{-1} . Thus

$$A_{\nu\mu} = \left(\frac{F}{U} \right)_\mu \cdot \frac{1}{(R/Q)_\mu \omega_{0\mu}} \quad (28)$$

and

$$C_{\mu\nu} = -\omega_{0\mu} \left(\frac{F}{U} \right)_\mu \quad (29)$$

Slater's analysis above lets us express the static Lorentz response as

$$\left(\frac{\Delta\omega}{V^2} \right)_{\nu\mu} = \frac{C_{\mu\nu} A_{\nu\mu}}{k_\nu} = -\left(\frac{F}{U} \right)_\mu^2 \cdot \frac{1}{k_\nu (R/Q)_\mu} \quad (30)$$

correctly showing that this constant is always negative: the mode's static resonance frequency gets lower as it is filled. Summing over all mechanical modes ν gives the total DC response, often quoted in units of $\text{Hz}/(\text{MV}/\text{m})^2$.

Using electrical measurements alone, it's not possible to constrain the scaling of x_ν . It is therefore helpful to rescale x_ν and F_ν each by a factor of $\sqrt{k_\nu}$, and eliminate k_ν from the equations. Instead of conventional units (m and N) for x and F , they now both have units of $\sqrt{\text{Joules}}$, so that $x \cdot F$ still represents energy. In this rescaled no- k case,

$$A_{\nu\mu} = \frac{1}{\omega_{0\mu}} \sqrt{-\frac{1}{(R/Q)_\mu} \left(\frac{\Delta\omega}{V^2} \right)_{\nu\mu}} \quad (31)$$

$$C_{\mu\nu} = -\omega_{0\mu} \sqrt{-(R/Q)_\mu \left(\frac{\Delta\omega}{V^2} \right)_{\nu\mu}} \quad (32)$$

It is perhaps an unexpected result that the cross-coupling between cavity modes (e.g., excite the π mode, measure $\Delta\omega$

for the $8\pi/9$ mode) is quantitatively predicted from measurements of each mode individually, with the exception of the choice of sign of the above radicals. All that is required is confidence that mechanical modes are correctly identified and non-degenerate.

HDL IMPLEMENTATION

The overall architecture of CMOC matches that of a real accelerator with a single-cavity single-source type of configuration as shown in figure 1. The Verilog implementation is flexible and makes use of generate loops to replicate instances for each building block to match a particular configuration. The controller implementation shown here is the one use in the real accelerator, hence the interest of running the live simulation using CMOC. The cryomodule physics model architecture is shown in figure 5 and implemented in an FPGA. The cavity electromagnetics simulator contains a 1st order low-pass filter for each Electromagnetic eigenmode, which are driven by the HPA and beam (combined into the drive signal in the block diagram). The mechanical eigenmodes are represented by the mechanical eigenmode propagator. The couplings between electromagnetic and mechanical eigenmodes are computed in the V^2 of each electromagnetic mode is transformed into Lorentz force by the outer product blocks and the mechanical displacements are transformed into frequency shifts by the dot product blocks. The diagram shows an example of a single cavity with two eigenmodes (the fundamental π mode and the $8\pi/9$ mode, of especial interest due to the proximity to the fundamental in TESLA-like cavities), where cavity-to-cavity interactions are propagated in the same fashion. The diagram also shows the virtual piezo interacting with the cavity mechanical modes as well as environmental noise sources, also coupling to the cavity electromagnetic eigenmodes through the mechanical structure.

The mathematical development so far showed the decomposition of the electrical and mechanical fields into eigenmodes leading to their state-space formulation in equations 15 and 19. In the interest of brevity, we show the discretization of the second-order equation from which one can deduce the reduced form for the implementation of the first order low-pass filter for the electrical mode computation.

Physics Model Implementation

The matrix calculations shown in Fig. 4 are applied every time step following equations 16 and 17. The box labeled “Mechanical Mode Dynamics, 2nd-order LPF” takes Lorentz forces for each mechanical eigenmode as an input (F_v) and produces a mechanical displacement (x_v). This corresponds to Eq. 19, the state-space formulation of the 2nd-order low-pass filter. Expanding that equation in matrix form, two expressions appear:

$$\frac{dx_v}{dt} = a_v x_v - b_v y_v, \quad (33)$$

and

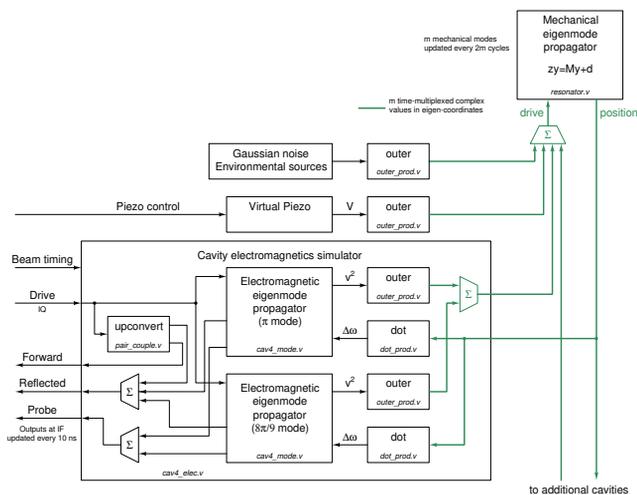


Figure 5: Physics model architecture.

$$\frac{dy_v}{dt} = b_v x_v + a_v y_v + c_v F_v \quad (34)$$

where:

$$a_v = \frac{-\omega_v}{2Q_v} \quad (35)$$

$$b_v = \omega_v \sqrt{1 - \frac{1}{4Q_v^2}} \quad (36)$$

$$c_v = -\frac{\omega_v}{k_v b_v} \quad (37)$$

These displacements influence each electrical mode's instantaneous eigenmode frequency ω_μ as follows:

$$\omega_\mu = \sum_v C_{\mu v} x_v \quad (38)$$

where C is the coupling matrix from mechanics to EM.

In order to discretize this equation, we will use the superscript n to indicate the current time iteration. Hence the following time step is superscripted by $n + 1$. We can thereby approximate a time derivative of a variable A in the following fashion:

$$\frac{dA}{dt} \approx \frac{A^{n+1} - A^n}{T} \quad (39)$$

where T is the time step size (usually expressed as Δt).

Discretizing Equations 33 and 34, we find:

$$x_v^{n+1} = x_v^n + T(a_v x_v^n - b_v y_v^n) \quad (40)$$

$$= x_v^n (1 + T a_v) - T b_v y_v^n \quad (41)$$

$$y_v^{n+1} = y_v^n + T(b_v x_v^n + a_v y_v^n + c_v F_v^n) \quad (42)$$

$$= y_v^n (1 + T a_v) + T(b_v x_v^n + c_v F_v^n) \quad (43)$$

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

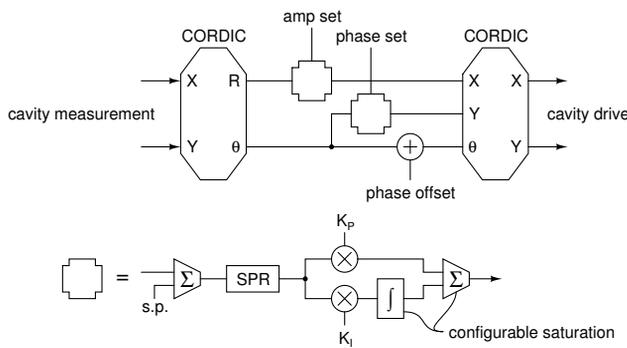


Figure 6: Cavity controller.

Fixed-point Scaling and Resources

Software simulators have the luxury of basing their calculations on numbers kept in normalized or SI units, since floating-point hardware has become standard on workstation-class computers. Current FPGA hardware resources do normally include hardware multipliers, but they are integer units, typically limited to 18×18 or slightly higher.

Scaled fixed-point arithmetic is used in all the steps described here. This requires attention in the problem setup to make sure that the quantities stay in a useful range, without overflowing or wasting precision. A simple example is the register holding cavity field \vec{V}_μ . The UI that configures the system requires selection of the largest representable field, and the register holds the binary fraction of that maximum. A similar argument holds for \vec{E}_{reverse} , except this time choosing a maximum is easier since it corresponds to a (modeled) ADC channel. Now the scaling of the coefficient connecting the two, $1/\sqrt{Q_{g\mu}(R/Q)_\mu}$, is well-determined and can be set by the UI.

Most of the data paths are based on complex numbers. We interleave I and Q on a single data path, saving nearly a factor of two in resources, with only minor complication of the control logic. For instance, a full-speed complex number multiplier takes two (scalar, fixed-point) hardware multipliers instead of four to provide a result at the maximum system data rate. To compensate for this interleaving, we run the simulation at twice the clock rate of the (simulated) ADCs.

With these techniques, the simulator for a single electrical mode consumes eight hardware multipliers. A typical configuration simulating three modes ($7\pi/9, 8\pi/9, \pi$) and a serialized DSP engine for the the mechanical modes consumes 27 multipliers total for a cavity system.

Controller Implementation

The core of the controller design is a Self-Excited-Loop (SEL), which has well understood [10] advantages for operating very narrow-band SRF cavities. Our flexible digital implementation, see figure 6, allows conceptually simple selection of amplitude-locked or not, and phase-locked or not, by adjusting clip limits on the PI controllers for amplitude and phase. The actuator for the phase loop is the reactive

(imaginary) component of the drive signal, such that the amplitude loop has no response to cavity detuning.

The important features of an SEL data path were clearly laid out in 1978, although using analog terminology. A modern rendition of the block diagram in the digital domain, loosely following JLab's experience, is shown in figure 6. Where SPR stands for stateful phase resolver, and it is only used in the phase channel.

Pure cavity-resonance-following SEL mode is easily set up with the above hardware by setting amplitude and phase K_I to zero, and using the amplitude loop saturation values to set the drive level. The Y input to the output CORDIC is set to zero at this point, and the phase offset needs to be given the right value for SEL operation.

Closing the amplitude loop is easily accomplished by putting in moderate values of K_P and K_I , and providing some separation between the lower and upper limits for drive amplitude.

With the amplitude stabilized, it's just as easy to close the phase loop. Here the stateful phase detector comes into play, forcing its output to $\pm\pi/2$ when the phase is spinning counterclockwise or clockwise. Only when the frequency error approaches zero does the output follow the input. Using the Y output CORDIC on the output of the phase loop gives natural amplitude compensation for detuning, as discussed by Delayen. In combination with the stateful phase detector and moderate values for the phase K_P and K_I , phase lock is naturally obtained by opening up the saturation limits on the phase channel. If X_{NOM} is the on-resonance drive amplitude, the peak amplitude used for phase lock (including off-frequency operation) becomes $\sqrt{X_{\text{NOM}}^2 + Y_{\text{LIM}}^2}$.

The real and imaginary clip limits will, in the end, be set to align with the power capabilities of the high-power amplifier, the fundamental power coupler's Q , beam current, and the allocated peak microphonic detuning.

In final phase-lock mode, there is a possibility of computing the small-signal proportional gain term separately, bypassing the CORDIC blocks and therefore achieving lower latency. It has been shown that system stability is primarily sensitive to the proportional term latency, not the integral term latency.

Noise Generator

ADC noise is modeled as Additive White Gaussian Noise (AWGN). A set of n uniform noise bits are added together to approximate the Gaussian distribution. Currently $n = 26$, giving an rms noise of 2.55 bits and a peak/rms ratio of 5.10. After dividing by two, this gives a white noise component that approximately matches measurements on a 14-bit LTC2175. The uniform noise bits are generated using the TT800 LFSR [11]. No attempt is made to model $1/f$ noise.

Communications

FPGA registers in the CMOC simulation engine and the cavity controller are accessible from software through the

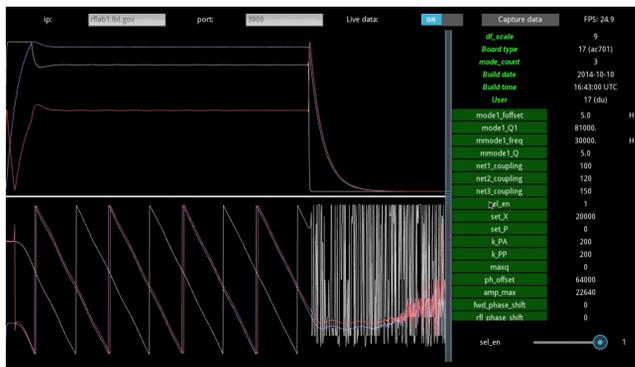


Figure 7: CMOC user interface.

network via GbE. This access is made possible by an in-fabric implementation of the GbE protocol developed at LBNL [12], which encodes and decodes UDP packets. Encapsulated inside the UDP payload there are register read and write commands to set configuration registers, read status registers, and to extract waveforms.

SOFTWARE IMPLEMENTATION

The software stack to support CMOC implements the drivers to communicate with the FPGA over UDP, a middle layer for conversion between Physics units and FPGA register values, and a user interface. The current implementation is based on Python, and optionally communicates with the FPGA directly over Ethernet without the intermediary of a control system [12].

CMOC serves as a platform for the development of the LLRF EPICS IOC. In addition to the user interface, CMOC has been the test bed used for the development of automation scripts used to tune the LCLS-II cavities [13], which have then been directly transferred for use in real cavities at the LCLS-II cryomodule test-stand at FNAL [14].

Python-based tools are used to parse the CMOC Verilog code at compile time and automatically generate a register map in JSON format. The same register map is then used on the software side and configuration registers are presented to the user. The JSON file is also stored (in compressed form) inside the FPGA, so one can retrieve the configuration from a running FPGA. Other information, such as the Git commit SHA hash, is also stored in the FPGA logic at compile time so as to be available at run time. There is a conversion layer between the Physics (SI) units presented to the user and the FPGA register values. In some case this translation is more involved than scaling and some assumptions are made on register names to establish the link between the register and their semantics.

The user interface (shown in figure 7) is based on Kivy [15], an Open Source Python library for user interface development. It allows the user to view waveforms generated by CMOC at the same rate as the implementation with real cavities. Waveform display is available in different formats: I-Q waveforms, amplitude and phase, 3D live display of the waveform vectors and FFTs. The user is then presented with

sliders for the configuration registers, for which the user interface loads the register map when it's launched. The user interface application also runs a configuration script to load default values onto CMOC at start-up and provides the possibility to run Python scripts to interact with CMOC from the user interface. In order to have access to these scripts from the user interface, the user stores Python scripts in a user directory and the user interface automatically generates buttons at start-up for the user to launch them.

OPEN SOURCE

The implementation of the CMOC simulation engine has been entirely written in various programming languages without the use of black boxes or proprietary software. The authors have requested the University of California to release the code to the public along with the documentation on the implementation and the physics involved and is being reviewed by LBNL's Technology Transfer Department at the time of this writing. As soon as it is approved for release it will be available on the LBNL (*a.k.a.* Berkeley Lab) GitHub page [16].

CONCLUSION

CMOC takes the FPGA-based LLRF system development cycle one step closer to the final implementation before it is tested with real cavities. It provides a framework to verify cavity controller implementations in HDL as well as the communications logic, drivers and higher level software tools. It is implemented in Verilog without the use of proprietary IP cores and is therefore portable for use under different FPGA-based platforms. It generates physically meaningful waveforms presented through a user interface the same way real cavity signals would, which can be used as a training platform for operators, users or developers as well as a testing framework for both HDL and software development of LLRF applications. It is currently in the process of being reviewed by LBNL's Technology Transfer Department for release to the public.

REFERENCES

- [1] L. Doolittle, A. Ratti, *et al.* "Design of RF Controls for Precision CW SRF Light Sources," in *Low Level RF workshop 2013*, Lake Tahoe, CA, October 2013.
- [2] P. Emma, *et al.*, "Linear Accelerator Design for the LCLS-II FEL Facility" in *Proceedings of FEL2014*, Basel, Switzerland
- [3] P. Emma, "LCLS-II Physics Requirement Document: Linac Requirements", LCLSII-2.4-PR-0041-R2.
- [4] C. Serrano, L. Doolittle, *et al.*, "End-to-end FEL beam stability simulation engine", in *Proceedings of IPAC2016*, Busan, Korea, 2016.
- [5] C. G. Montgomery, R. H. Dicke. E. M Purcell, "Principles of Microwave Circuits", MIT Radiation Lab Series V8, 1947.
- [6] L. R. Doolittle, "Understanding 5-cell mode structures", JLab tech note CEBAF-TN-0120, May 1989.

- Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.
- [7] Slowly varying envelope approximation (SVEA), http://en.wikipedia.org/wiki/Slowly_varying_envelope_approximation.
 - [8] T. Schilcher, “Vector Sum Control of Pulsed Accelerating Fields in Lorentz Force Detuned Superconducting Cavities”, Hamburg 1998.
 - [9] J. R. Delayen, “Ponderomotive Instabilities and Microphonics – A Tutorial”, in *Proc. SRF’05*, Ithaca, NY, July 2005, paper SUP01, pp. 35–39.
 - [10] J. R. Delayen, “Phase and Amplitude Stabilization of Superconducting Resonators,” Ph.D. Thesis, Caltech (1978).
 - [11] M. Matsumoto and Y. Kurita, “Twisted GFSR Generators II,” in *ACM Trans. on Modeling and Computer Simulation*, 4(1994), 254–266 DOI:10.1145/189443.189445.
 - [12] C. Serrano, L. Doolittle, “FPGA Communications based on Gigabit Ethernet”, in *ICALEPCS 2011*, Grenoble, France, October 2011.
 - [13] C. Serrano, L. Doolittle, *et al.* “LLRF Control of High Q_L Cavities for the LCLS-II”, in *Proceedings of IPAC2016*, Busan, Korea, 2016.
 - [14] L. Doolittle *et al.*, “High-precision RF Control for SRF Cavities in LCLS-II”, in *SRF Workshop 2017*, Lanzhou, China, July 2017.
 - [15] Kivy, Open Source Python library for user interface development, <https://kivy.org>
 - [16] LBNL GitHub Page, <https://github.com/BerkeleyLab>