Analog Cavity Emulators to Support LLRF Development

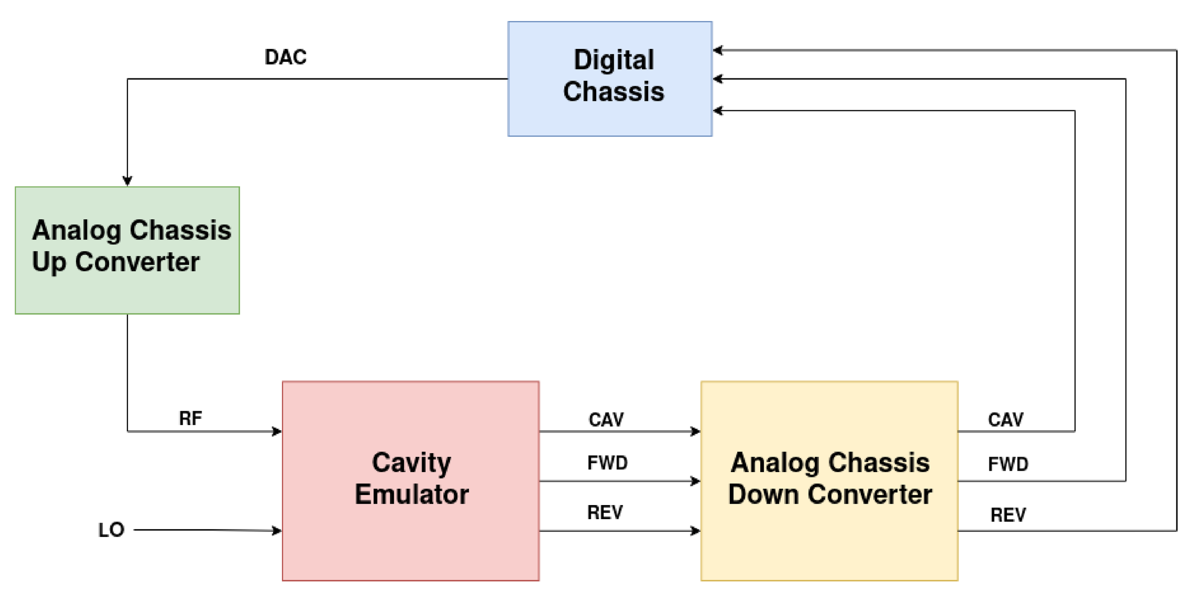
S.D. Murthy*, L. Doolittle, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
A. Benwell, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA
sdmurthy@lbl.gov*

Abstract

The goal of a LLRF system is to control an actual RF cavity with beam. While digital simulations have a place, having an analog circuit to stand in for the cavity can be tremendously helpful in validating hardware+firmware+software under development. A wide range of cavity emulators have been developed in collaboration with SLAC, and LBNL. Cavity emulators are typically based on quartz crystals and frequency conversion hardware. The choice of crystal frequency and coupling mechanism depends in part on the bandwidth and coupling of the cavity it's intended to emulate. Examples of bandwidth range from 800 Hz (SLAC) as a stand-in for a SRF cavity, to 31 kHz (LBNL) for a room-temperature accumulator-ring cavity. An external LO is used to tune the emulated cavity frequency. The coupling properties are also of interest if the scope includes emulating reverse power waveforms. LLRF system checks such as closed-loop bandwidth, and determining cavity detuning can be performed interactively and as part of a Continuous Integration (CI) process. This paper describes the design, implementation, and performance of the cavity emulators.

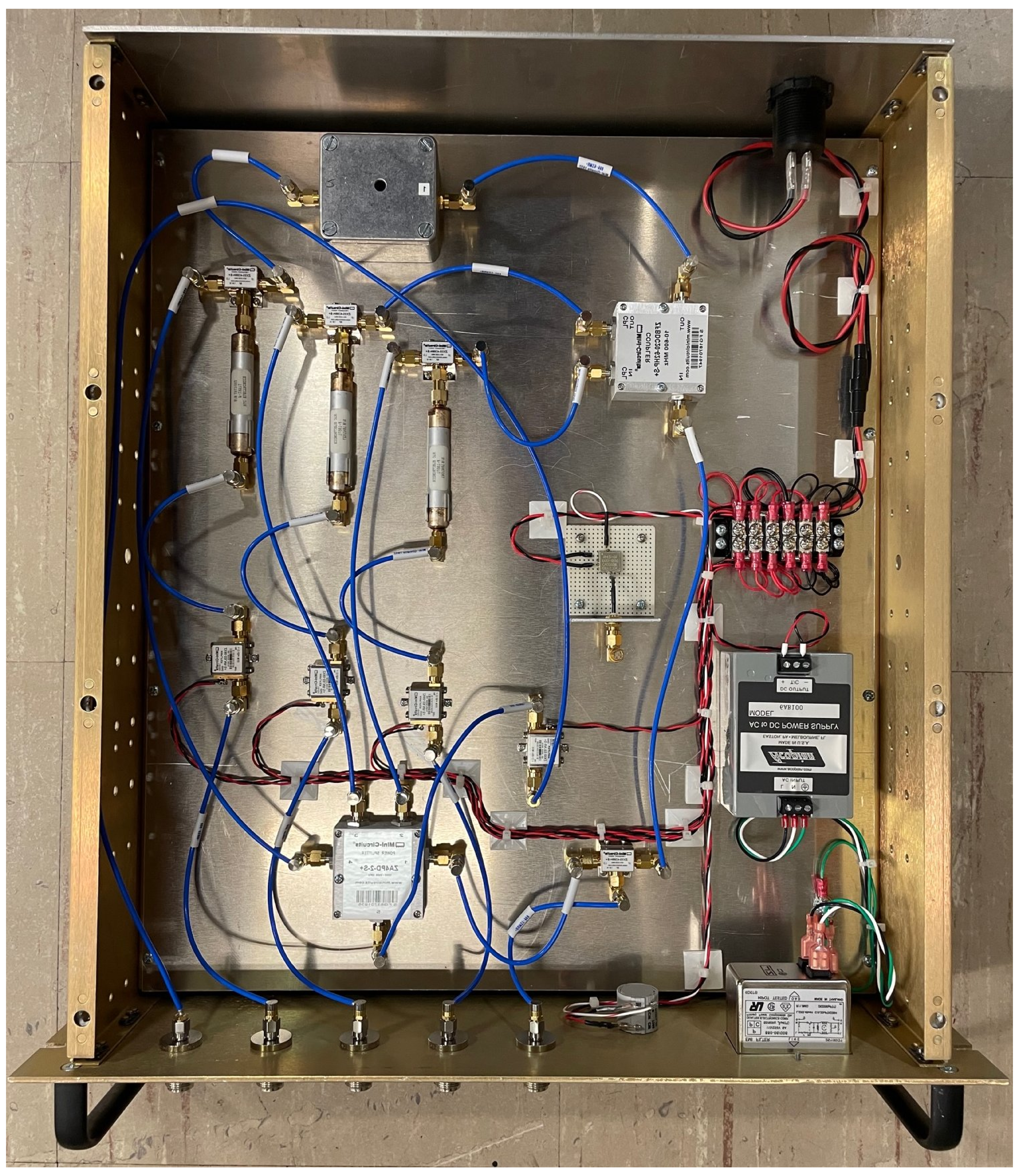
Integration & Performance

Cavity Emulator integrated with the LLRF system



- **Deployed for Continuous Integration (CI)**
- **Automated tests to test changes to hardware/firmware/software**
- **LLRF code-base is always tested and ready for deployment**
- **Includes bringing up the emulator and closing RF feedback loop**
- **Measure closed-loop BW and total group delay**
- **Characterize cavity detuning function by changing LO**
- **In-loop stability analysis met for SLAC and LBNL LLRF systems**

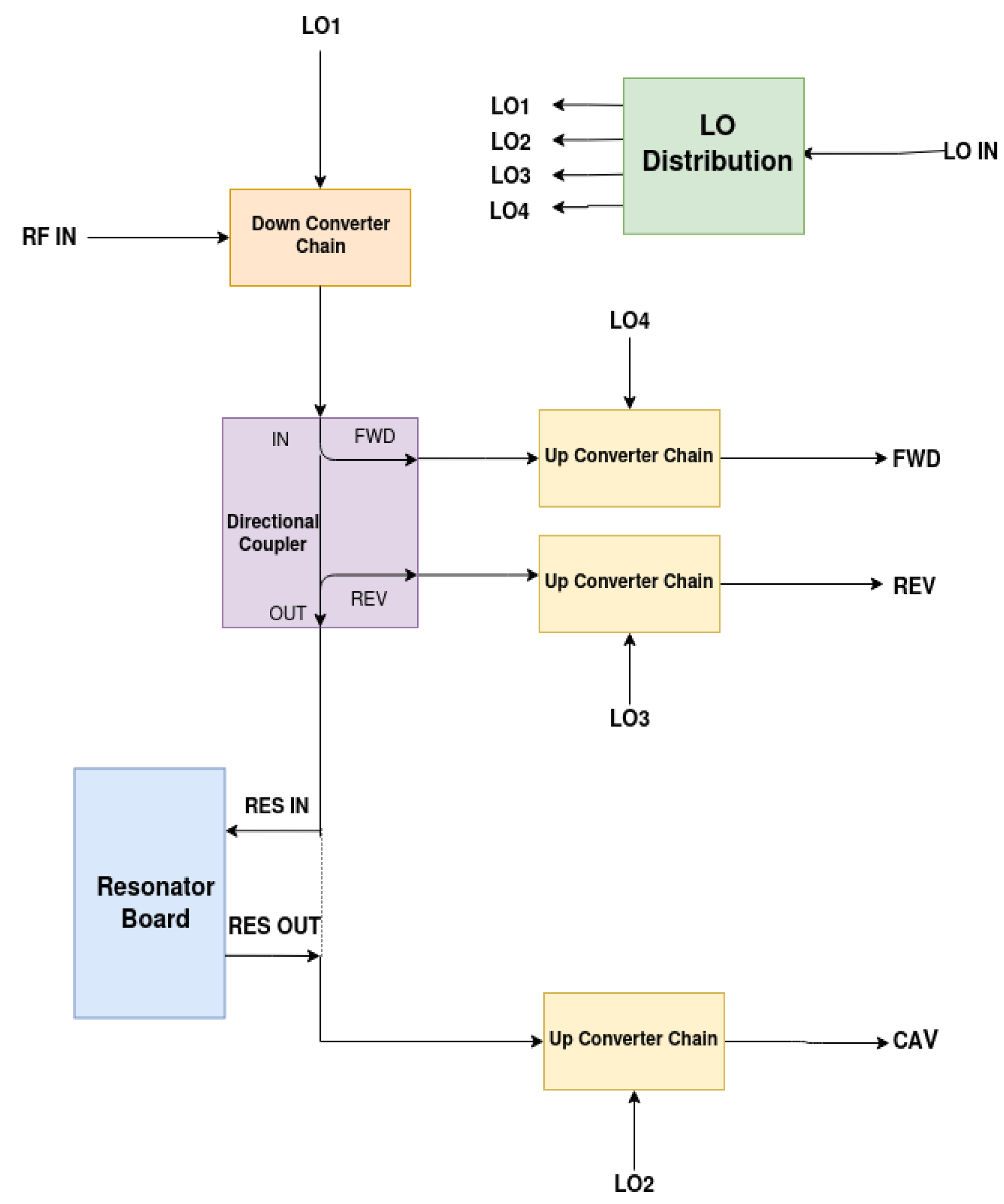
Componentized cavity emulator at SLAC



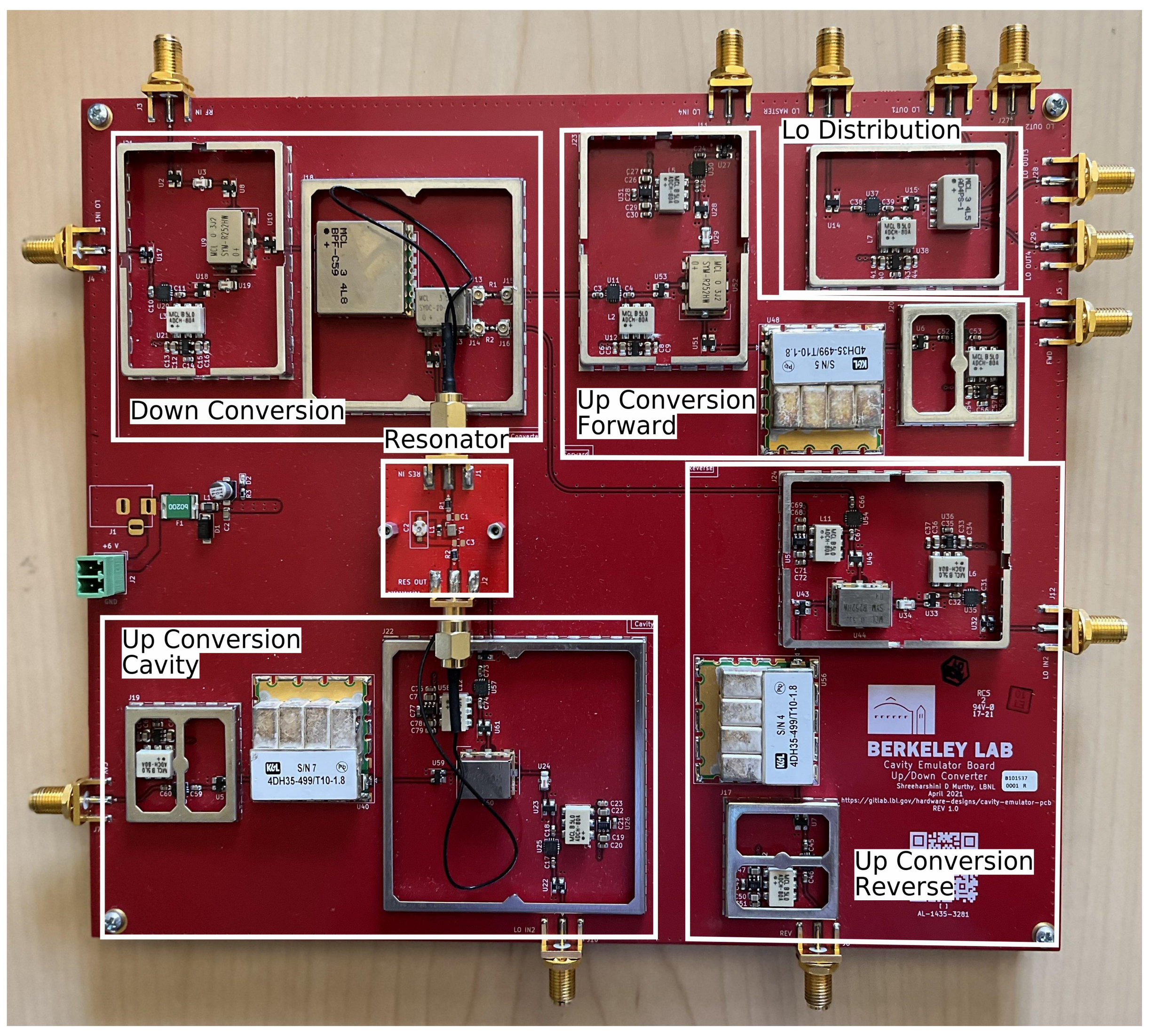
Conclusion

- Cavity emulators designed and built in collaboration with SLAC, and LBNL.
- Offer a good benchtop evaluation for testing LLRF systems.
- Train system engineers to understand the loop parameters better.
- Emulation of different RF cavities can also be achieved with the existing system by using a different crystal, coupling mechanism, and up conversion bandpass filters.

Cavity Emulator Design



Cavity Emulator PCB at LBNL



Simple linear regression model with Coupling and Rotation

$$y_i = \beta_0 + \beta_1 + \beta_2 \cdot a + \beta_3 \cdot a + \epsilon_i$$

1. $y_i = S_{21}$ data of the crystal in complex form or [real, img] form array
2. a = Lorentzian function

```
In [5]: def basis_cpx(params, f):
            f0, Q = params
            constant = np.array(len(f)*[1+0j])
            lorentz = lorentzian_complx(f, f0, Q)
            basis = np.vstack((constant, constant*1j, lorentz, lorentz*1j))
            return basis
```

```
In [6]: def residuals(params, f, s21_data, ret='beta'):
            y  = np.concatenate((s21_data.real, s21_data.imag))
            basis = basis_cpx(params, f)
            a = np.vstack([np.concatenate((b.real, b.imag)) for b in basis]).T
            beta, res, rank, sigulars = np.linalg.lstsq(a, y, rcond=None)
            if (ret == 'beta'):
                return beta
            elif (ret == 'residuals'):
                return res
            return None
```

Non-linear Fit

```
In [7]: f, y = c.freq_data, c.s21_complex
        fit_nls = least_squares(residuals, guess, args=(f, y, 'residuals'))
        fit = lorentzian_complx(f, *fit_nls.x)
        print("Initial guess values = ", guess)
        print("Final guess values = ", fit_nls.x)
        print("Least Squares status = %d, iterations= %d, cost = %d"% (fit_nls.status, fit_nls.nfev, fit_nls.cost))

        fit_beta = residuals(guess, f, y, ret='beta')
        print("Beta values = ", fit_beta)
        fit_f = np.sum(np.vstack(fit_beta[:]) * basis_cpx(fit_nls.x, f), axis=0)

        Initial guess values =  (39.998928, 1341.2312074626063)
        Final guess values =  [  39.99878304 1341.23120746]
        Least Squares status = 3, iterations= 19, cost = 0
        Beta values =  [ 0.01317145 -0.00841011  0.04053251 -0.03413317]
```

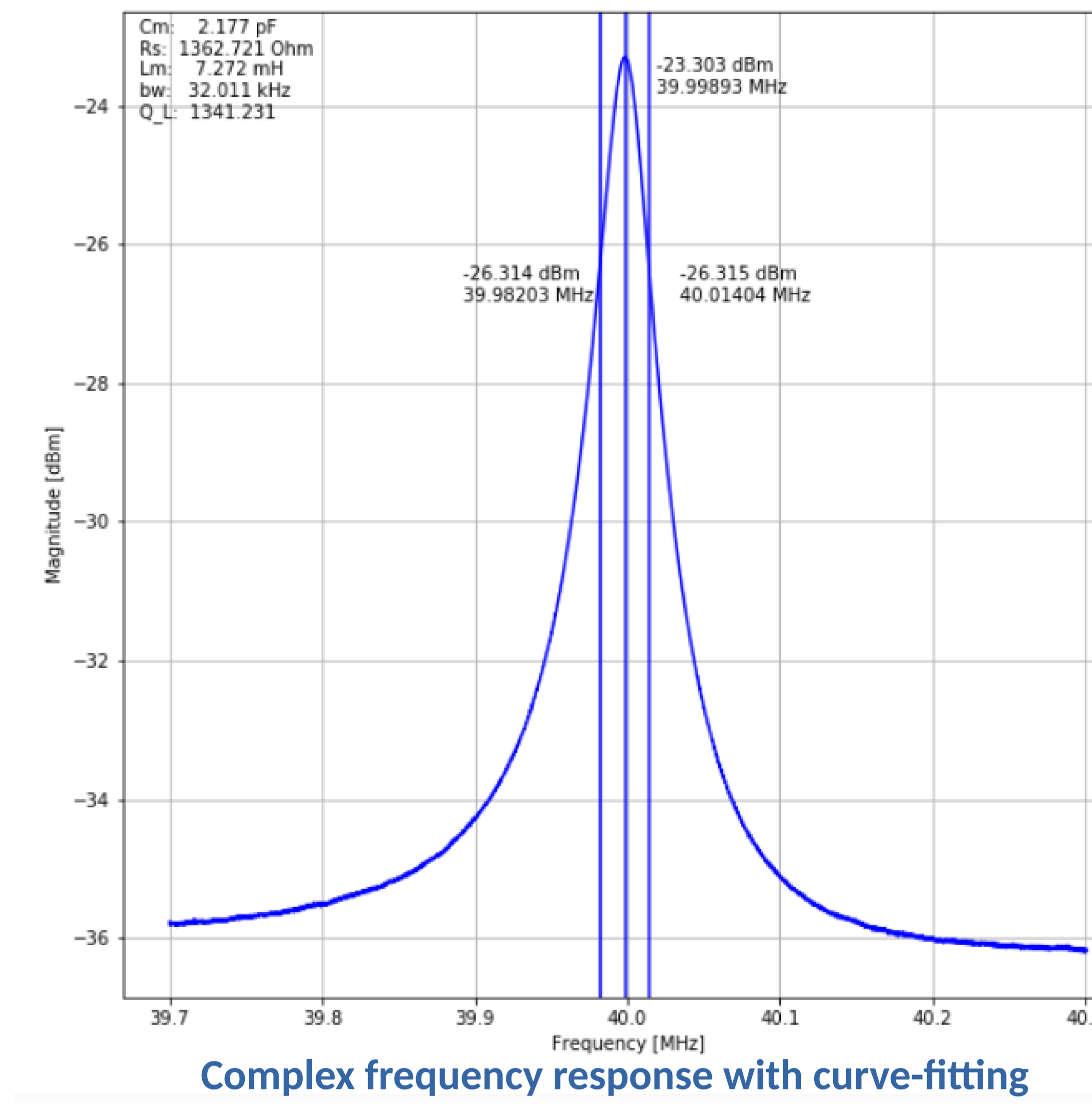
```
In [8]: mse = np.square(abs(np.subtract(y, fit_f))).mean()
        rmse = np.sqrt(mse)
        print("Root Mean Square Error :%4f" % rmse)
        print("RMSE Normalised        :%4f" % (rmse/abs(y).mean()))

        Root Mean Square Error :0.000215
        RMSE Normalised        :0.010033
```

```
In [13]: plt.figure()
         fnom = 40e6
         fp = f
         plt.plot(fp, fit_f.real, label='Real fit', color='y',linewidth=3)
         plt.plot(fp, y.real, label='Real data', linestyle='dotted', color='k', linewidth=4)
         plt.plot(fp, fit_f.imag, label='Imag fit', color='c',linewidth=3)
         plt.plot(fp, y.imag, label='Imag data', linestyle='dotted', color='r', linewidth=4)
         plt.grid(True)
         plt.legend(loc='best')
         plt.xlabel("Frequency [MHz]")
         plt.ylabel("S21")
         plt.savefig("iq.png")
         #plt.show()
```

- An analog cavity emulator consists of two components:
  - Crystal resonator
  - Frequency conversion chains
- Resonator is a commercial quartz crystal, with additional passive components. Crystals are modeled as a series RLC circuit (mechanical vibrations) with a parallel Capacitor (electrodes attached to the crystal).
- All crystal parameters, such as Equivalent Series Resistance (ESR), Shunt Capacitance (C_0), Motion Inductance (L_m), Motion Capacitance (C_m), Series Resonance Frequency (f_s), and Quality factor (Q_L) characterized using the measured transfer function (S_{21}) to achieve the desired bandwidth and minimize broadband coupling.
- Broadband coupling is reduced: adding compensating inductance in parallel with the crystal (cancels the crystal's parasitic shunt capacitance).
- For trimming : use a variable inductor (SLAC), or use a fixed inductor with an additional variable capacitor (LBNL).
- Bandwidth adjustment - a series resistance. If it increases, the signal strength decreases and the bandwidth increases.
- Some loss in the signal strength is acceptable, because the up-conversion chains need to be designed to handle forward and reverse signals attenuated by the directional coupler.
- The frequency conversion hardware:
  - Single down-conversion chain - convert the incoming RF cavity drive signal to the crystal resonance frequency
  - Three up conversion chains
- The emulated cavity output is then up converted to RF. Forward and reverse signals are also be produced with a directional coupler at IF, and then up converted.
- Addition of well-defined bandpass filter to remove any spurs/leakage after the up-conversion chains.
- An external signal source (LO) to tune the emulated system's resonance frequency.

Crystal resonance frequency (LBNL) (span = 600 kHz)

Measured loaded Q_L: 1341
Full-bandwidth: 31 kHz
Insertion loss: 23 dB



Complex frequency response with curve-fitting