The Hows and Whys of SEL feedback for LLRF

Larry Doolittle, LBNL, February 2014

At its simplest, LLRF control should be thought of as a negative feedback loop, just like an op-amp except with complex numbers. For SRF cavities, consideration of setup and off-normal conditions always lead designs to be based on self-excited loop (SEL) topology. Merging the two, without specifying excessive FPGA resources or computational delays, is an art. This paper will explain and develop one particular implementation, targeted at LCLS-2. At the conclusion, synthesizable HDL code will be shown to work in a system simulation.

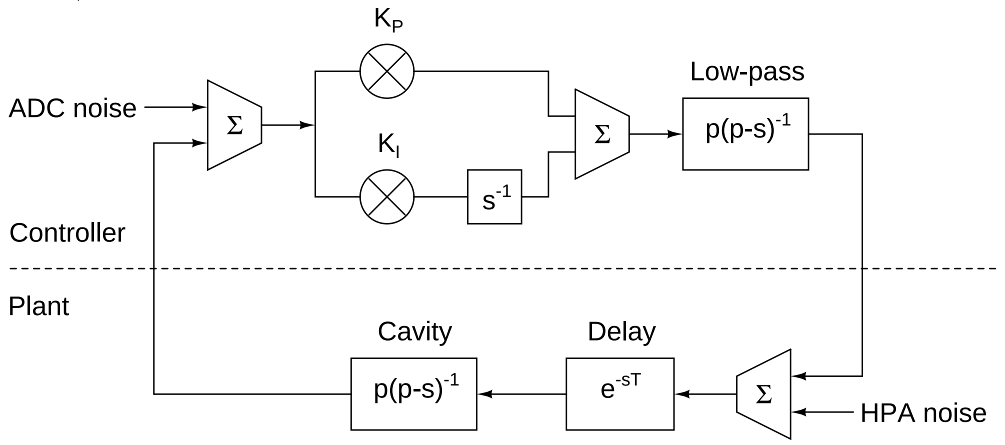
The only topic here is the computation in the baseband complex plane. Downconversion, upconversion, and any linearity corrections are not considered. The plant is simplified to include primarily the cavity, with its pole at *circa* 20 Hz, negative real, plus an imaginary component representing detuning. The system delay, on the order of 800 ns, is also an essential part of the plant. I will also assume that ADCs on the order of 100 MS/s are used, which after processing will produce at most one complex number measurement per two clock cycles. Thus the data rate is at most 50 MS/s of complex numbers. The single-sample SNR of these converters is further assumed to be around 70 dB.

The net effect of negative feedback is to extend the bandwidth of the cavity, from the open loop value of ∼20 Hz, to a closed loop value up in the ∼60 kHz range. That implies a proportional gain term of ∼3000. Without band-limiting, such a high gain transfers far too much noise from the input channel (ADC and related terms) to the power amplifier. Indeed, the simple combination of -70 dB input SNR and 70 dB of gain gives as much noise as signal in the output drive.

Drive at frequencies much beyond the closed-loop cavity bandwidth does nothing useful, and can be eliminated by a first-order low-pass filter. The choice of cutoff frequency can be optimized later. For now, list it as 300 kHz, a factor of 5 higher than the 0 dB loop crossing. While this filter does nothing to the close-in noise power density delivered to the output, at least now the total noise power is only 2% of the nominal drive. This filter introduces an additional 530 ns delay to the feedback loop, but the total (now 1330 ns) is still consistent with a 0 dB crossing at 60 kHz.

Proportional-only feedback has the well-known defect of leaving residual static error. An additional integral feedback term can fix that problem. System stability requires that the integral feedback term not significantly perturb the system at the 0 dB loop crossing frequency, but only contribute at much lower frequencies.

The controller/plant abstraction as developed so far is shown here:



What is not shown explicitly in this diagram is that the imaginary part of the cavity pole varies with time. Its motion is driven by microphonics, helium pressure fluctuation, Lorentz forces, and the tuner subsystem.

The noise suppression performance of this system is easily analyzed in the frequency domain with Laplace techniques. In terms of the plant and controller gains

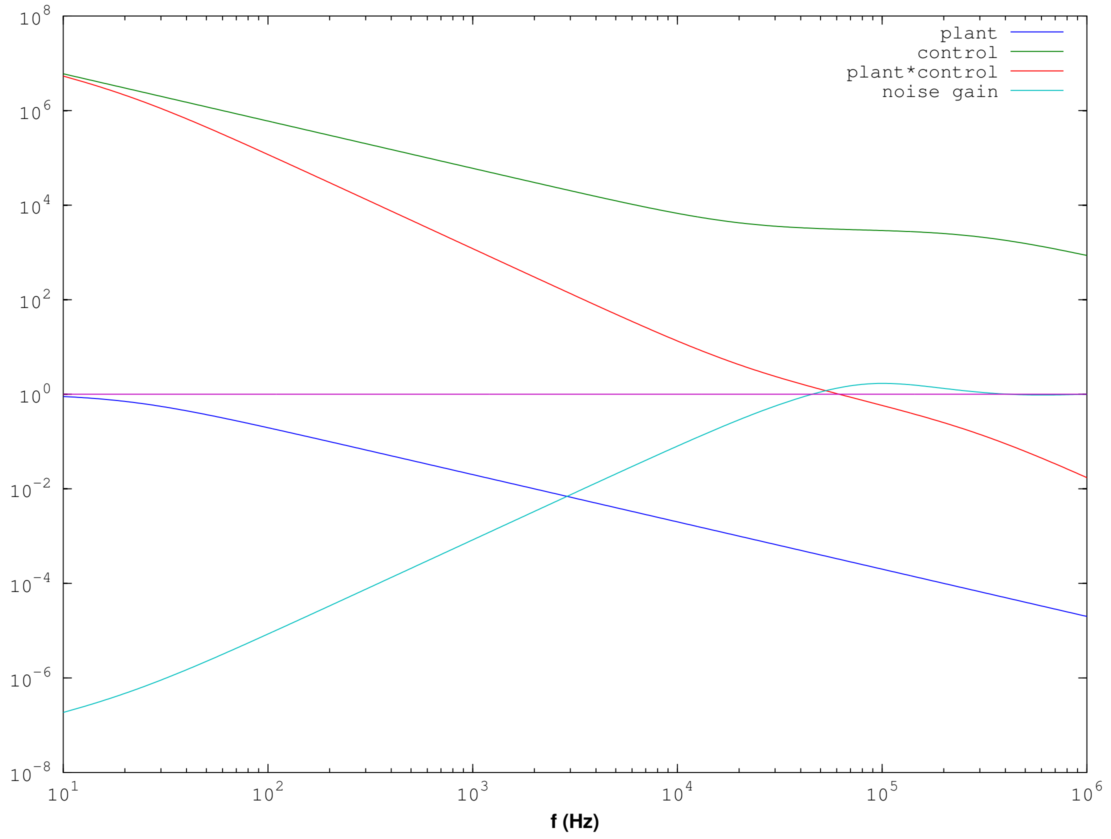
$$A = e^{-sT} \frac{p_C}{p_C - s}$$

$$K = (K_P + K_I/s) \frac{p_L}{p_L - s}$$

the noise gain is simply

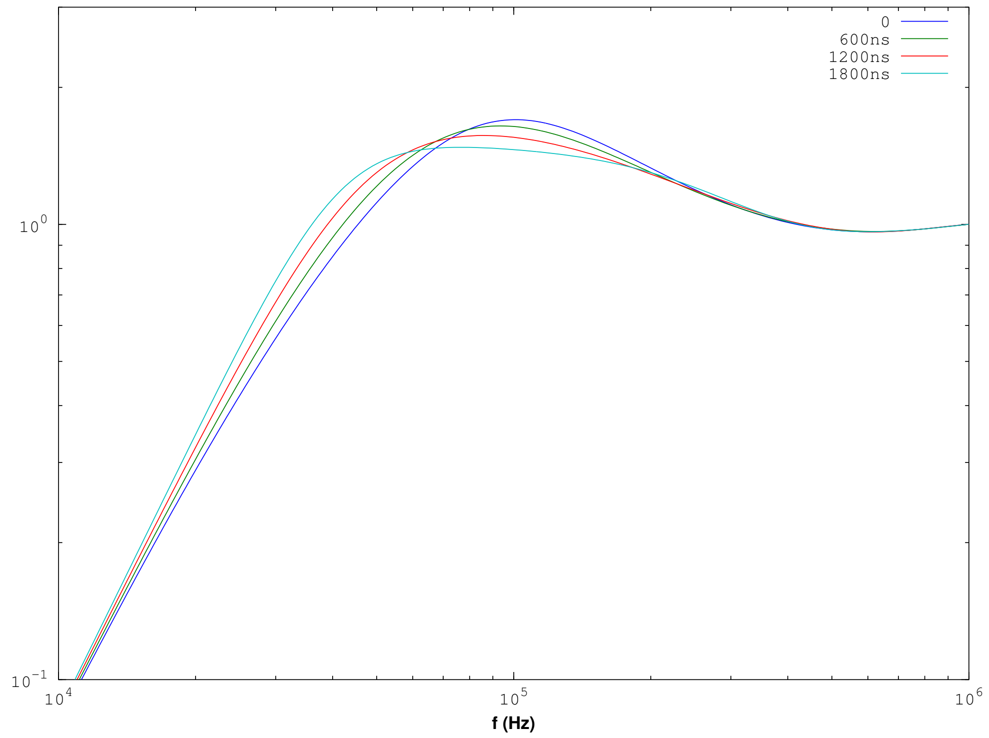
$$\frac{1}{1 + AK}$$

Punching these equations into Octave/Matlab, and using $K_I = 2\pi \cdot 20\,\mathrm{kHz} \cdot K_P$, gives these frequency response curves:



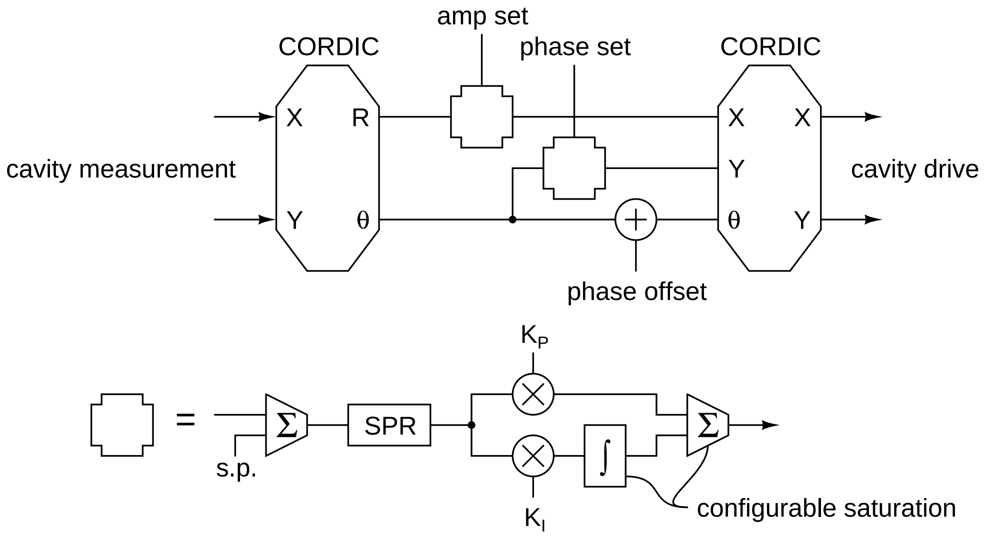
Now the complications begin. Adding features, in particular SEL mode for turn-on and handling large detuning excursions, involves lengthy calculations. In particular, it is considered normal to implement a SEL by converting from rectangular to polar with a CORDIC, computing PI feedback, and converting back to rectangular with another CORDIC. Retuning the feedback loops to account for these computational delays would reduce overall feedback performance.

There are two key observations. The first is that, in the small signal, that intricate calculation must get exactly the same result as the K_P construction in the simple diagram. The differences arise in the large signal cases, as can be generated by the K_I term. The second observation is that the delay sensitivity of that K_I term is much weaker than that of the K_P term, since the former only becomes important at lower frequencies. That conclusion can be quickly borne out by computing the noise gain for various *additional* delays in the K_I term, shown here:



So we have huge freedom – easily 100 cycles – to do all the arithmetic we want in the FPGA, as long as it is limited to the K_I path.

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 here:



Where SPR stands for stateful phase resolver, and it is only used in the phase channel. More on that later.

This diagram shows both proportional and integral terms. The stability limit for the proportional term will be substantially smaller than the limit for the direct, non-CORDIC-based path. It's included here primarily so that the amplitude loop can be closed during free-running SEL operation. For highest performance, in final operation this proportional path will be disabled, and the direct path will take over.

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.

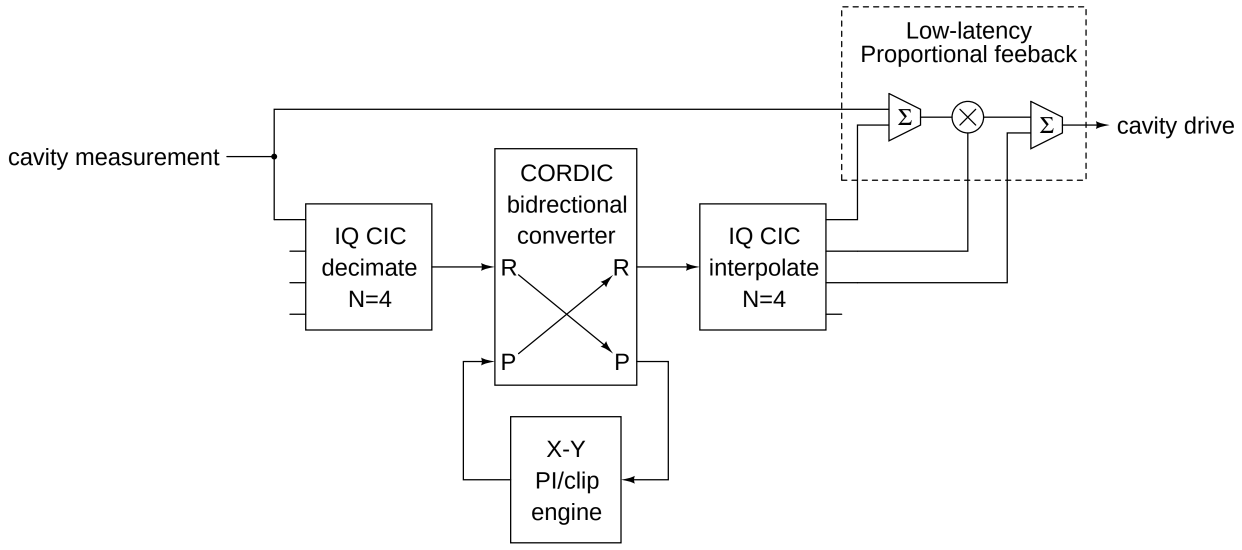
With the amplitude stabilized, it's just as easy to close the phase loop. Here the stateful phase detector comes into play, forcing its output to $\pm\pi/2$ when the phase is spinning counterclockwise or clockwise. Only when the frequency error approaches zero does the output follow the input. Using 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 $X_{\rm NOM}$ is the on-resonance drive amplitude, the peak amplitude used for phase lock (including off-frequency operation) becomes $\sqrt{X^2_{\rm NOM} + Y^2_{\rm LIM}}$.

Show an Octave-based simulation of the process here?

Once closed loop operation is established, the direct (low-latency) feedback path should be configured to replace the proportional gain terms used in the CORDIC path. This can give higher gains, and therefore better control of disturbances. The direct path will also tend to have less DSP noise.

Of course, the implementation of this logic in the FPGA doesn't look much like the abstract data path above. In particular, when the final direct proportional loop is in use, its setpoint and gain need to be set consistently with the CORDIC path. I therefore choose to use the module containing the CORDIC to generate those values, time-multiplexing the computational units (a key concept from DSP Tutorial III, LLRF13).

The block diagram of the feedback path, not including the low-pass filter, looks like:



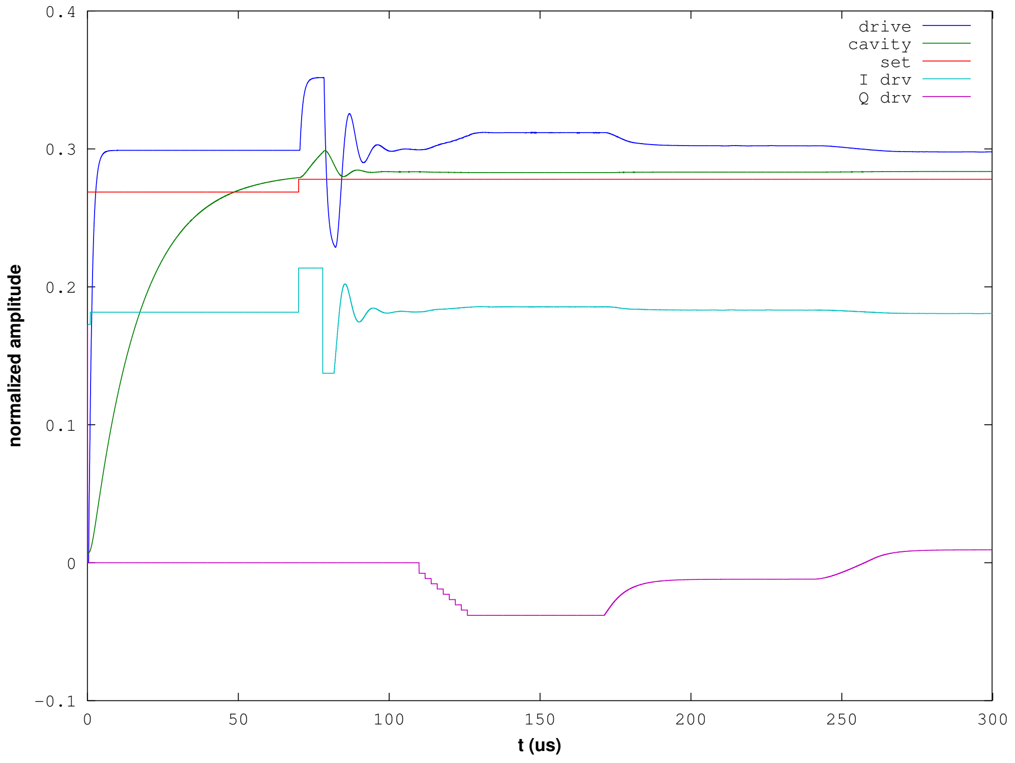
Where the X-Y PI/clip engine evaluates both the amplitude and phase terms of the feedback path shown in the previous figure.

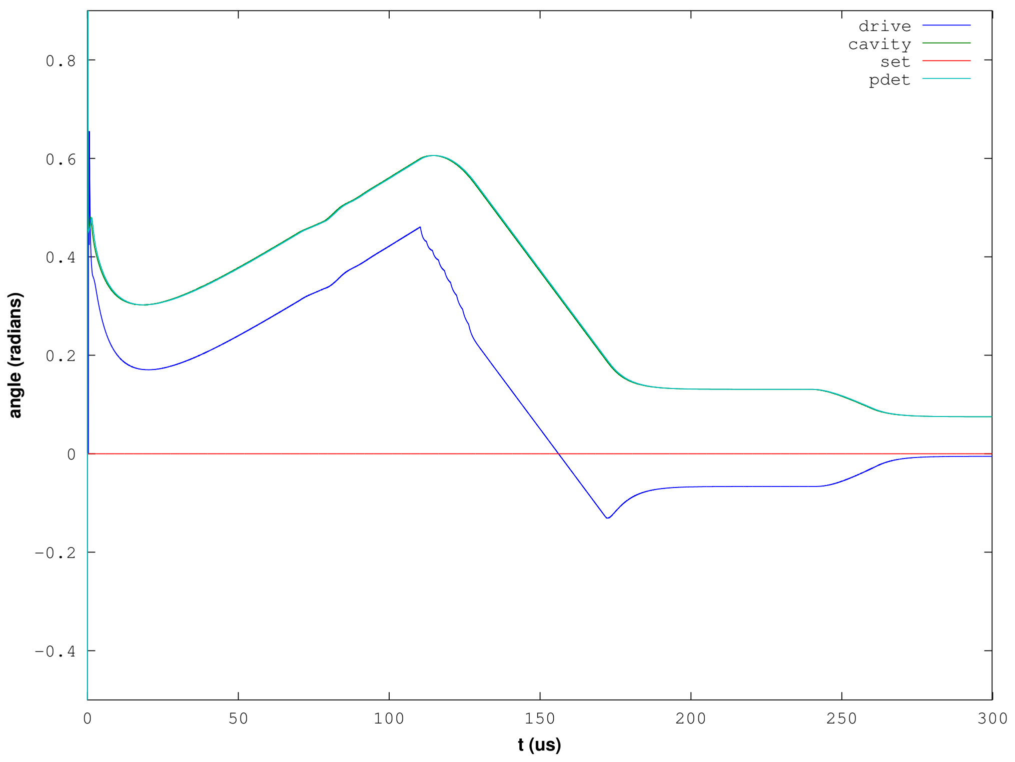
Not counting the output low-pass filter, this construction takes less than 10% of a small FPGA (XC6SLX45). It represents 618 lines (non-blank, non-comment) of synthesizable Verilog HDL, much of which was re-used from previous projects.

Elaborate on combining the low pass filter with an $8\pi/9$ notch filter at about -800 kHz.

These simulations all use a time step of exactly 10 ns. While there are good reasons (non-IQ reference) for thinking the ADC sample rate will not be exactly 100 MHz, there are also good reasons for thinking this estimate is not wrong by more than 20%.

While there are still a few bugs in the system, I can show a system simulation of the Verilog turning on a virtual cavity. This cavity has an open-loop bandwidth of 10 kHz and is detuned by 2 kHz. The phase loop is turned on by gradually opening up the quadrature drive limits between $t = 110\,\mu$s and $t = 126\,\mu$s.





While this work (when complete) should be viewed as a significant down payment on the required gateware, the following items are deliberately not considered here:

- ∘ Down- and up-conversion
- ∘ Phase calibration
- ∘ Input channel distortion correction (optional)
- ∘ Output channel distortion correction
- ∘ Determination of SEL phase offset
- ∘ Waveform feedforward (triggered by an event receiver)
- ∘ Piezoelectric actuator control
- ∘ *in-situ* calibration of $8\pi/9$ notch filter
- ∘ Input from the Beam-Based Feedback subsystem
- ∘ Interlocks

These topics are all over the map in terms of available laboratory code base and field experience, and associated technical risk. Barring surprises, most of them should be solvable with a process of integrating and testing existing code. "High Level Applications" support is often a sizable fraction of that effort.

References:

Phase and Amplitude Stabilization for Superconducting Resonators, Jean R. Delayen Ph.D. Thesis, 1978

A Digital Self Excited Loop for Accelerating Cavity Field Control, T. Allison *et al.*, Proceedings of PAC07, Albuquerque, New Mexico, USA

DSP Tutorial III, Avoiding Resource Overutilization in FPGAs, L. Doolittle, LLRF13, Lake Tahoe, October 2013

Digital Low-Level RF Control Using Non-IQ Sampling, Lawrence Doolittle *et al.*, LINAC 2006