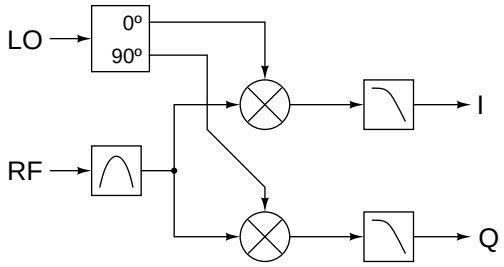
Digital IQ Reconstruction from a non-IQ Sampled Waveform

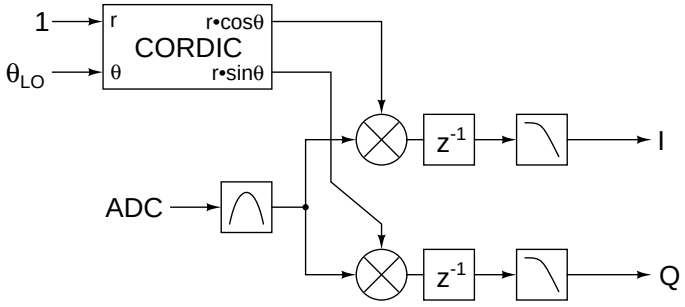
Larry Doolittle, LBNL, November 2008

In a digital RF system, there are sound reasons for wanting to represent a waveform in the form of a complex vector (phasor), commonly described with its I and Q components. If the conversion from an oscillating waveform to I and Q signals is performed with analog electronics, the block diagram necessarily looks like this.



When the signal processing path is used in a feedback loop, choosing the output filter ends up as an awkward compromise between group delay and second harmonic rejection.

This same circuit can be implemented in the digital realm, and comes out looking roughly like this.



That design still leaves open the question of the output low-pass filter, with its attendant compromises. While there are times when this design makes sense, the pressures for low latency around a feedback loop motivate a search for something better. Fortunately, the range of capabilities in the digital realm is richer than that in the analog realm. The following construction shows no spurs with a steady-state sine wave input, and has an intrinsic delay of less than one cycle.

Suppose we digitize an IF carrier with θ rotation between samples, and any modulation of the carrier is slow compared to the sample rate. Then two successive samples are

$$y_n = I\cos n\theta + Q\sin n\theta$$

$$y_{n+1} = I\cos(n+1)\theta + Q\sin(n+1)\theta \quad ,$$

better represented by

$$\begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} \cos n\theta & \sin n\theta \\ \cos(n+1)\theta & \sin(n+1)\theta \end{pmatrix} \begin{pmatrix} I \\ Q \end{pmatrix} \quad .$$

To reconstruct I and Q, take the these two samples and multiply by the inverse matrix:

$$\begin{pmatrix} I \\ Q \end{pmatrix} = \frac{1}{D} \begin{pmatrix} \sin(n+1)\theta & -\sin n\theta \\ -\cos(n+1)\theta & \cos n\theta \end{pmatrix} \begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix} \quad ,$$

where the determinant D simplifies to $\sin\theta$.

If $\theta = \pi/2$, and we only consider even n, the equation simplifies to

$$\begin{pmatrix} I \\ Q \end{pmatrix} = (-1)^{n/2} \begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix} \quad ,$$

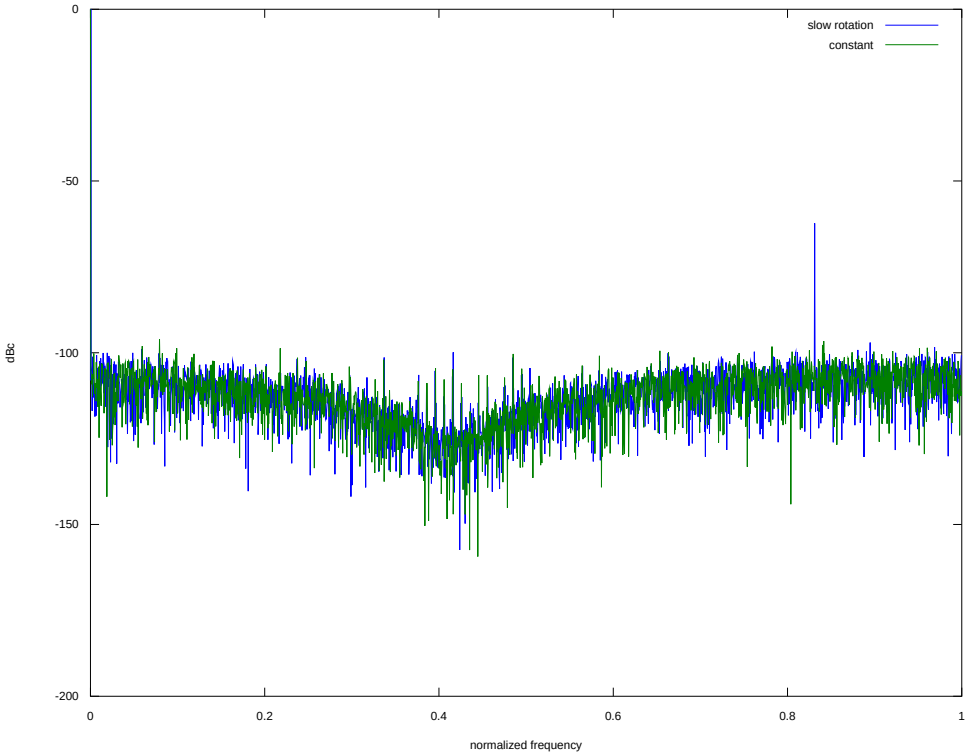
which describes the IQ sampling process popular at the turn of the century.

Considering the degenerate IQ input case makes it clear that the output data rate as originally drawn is double the sensible rate. The output I and Q data streams should be decimated by a factor of two to avoid double-referencing the input data. To maintain data flow through later processing steps, it is natural to also multiplex I and Q data streams, so the one-datum-per-clock-cycle paradigm is continued.

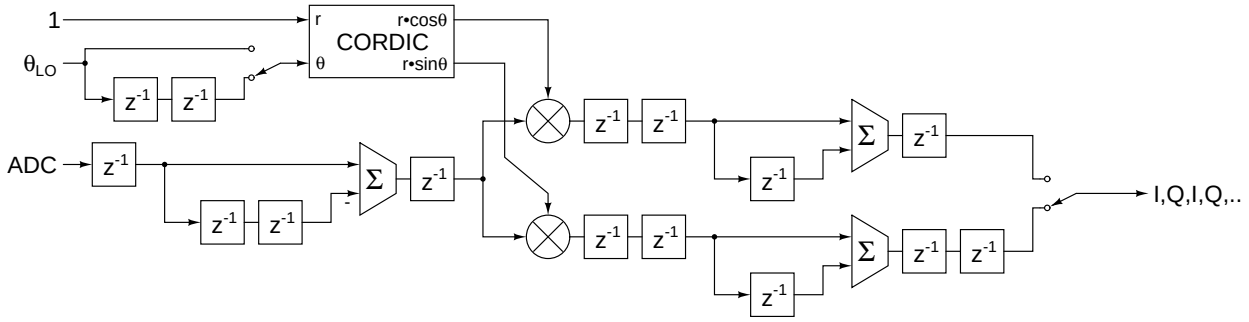
If measurements of y_i have random error σ, the resulting I, Q vector has a random error $\sqrt{2}\sigma/D$. For $\sin\theta \neq 1$, there is a correlation between I and Q errors.

If a DC offset is present on the input, it will generate a spur in the output at the LO frequency. The conventional DC blocking filter to use on near-IQ sampled signals is $\frac{1}{2}(1 - z^{-2})$. If this filter is placed before the above reconstruction, and the $1/D$ term is dropped from the above equation, the signal gain at the center frequency is D^2, and the noise gain is unity. The noise spectrum is no longer white, however, so the effects of post-filtering are painful to compute in general.

The following figure shows the noise spectrum as simulated with a 21/101 ratio of IF frequency to sample rate, and 1 bit rms added to a 7000 bit amplitude sine wave. The slow rotation curve shows a spur (-60 dBc at normalized frequency 84/101) caused by rotating the input sine wave with a period of 2121 samples. The white noise component of both spectra totals 1.18 bits, referred to the input. Since the spectra are clean at low frequencies, additional filtering can predictably trade off noise and bandwidth. The dominant feedback pole, usually the cavity resonance itself, will necessarily provide such filtering.



The final FPGA programming block diagram has only subtle differences from the original analog diagram.



Manipulating the r and θ inputs to the CORDIC allow this circuit to match gain and phase of multiple channel inputs to a vector sum. The output I, Q stream is thus ready to be summed across per-cavity instantiations.

An implementation in Verilog has been thoroughly tested in both simulation and hardware. Besides the 1.5 essential cycles of delay, it adds five pipeline cycles to allow operation at up to 150 MHz in an XC3S chip. It adds less than one bit of noise to the signal from a 14-bit ADC. Resource consumption is about 1000 4-input LUTs (mostly in the CORDIC), plus two multipliers. That is small enough to fit 8 channels in a US$40 XC3S1000, or 32 channels in a US$250 EP2C50.