

# On FM Demodulators in Software Defined Radios Using FPGAs

Michael Rice, Marc Padilla, Brent Nelson  
 NSF Center for High-Performance Reconfigurable Computing (CHREC)  
 Department of Electrical & Computer Engineering  
 Brigham Young University, Provo, Utah, USA

**Abstract**—The use FPGAs to implement sampled-data FM demodulators for software-defined radios that must support “legacy waveforms” is explored and analyzed. Feed-forward and feedback structures are examined. The best feed-forward structure, in terms of the time/area trade-off, is the arctangent-differentiator structure. The arctangent-differentiator and PLL demodulators have approximately the same time/area product and approximately the same SNR performance. However, the two occupy very different locations in the time/area trade-off space. Relative to the PLL demodulator, the feed-forward demodulator can achieve a much higher clock rate, but requires more area.

## I. INTRODUCTION

The software defined radio (SDR) is playing an increasingly important role in military communications. Inevitably, the SDR will have to possess the capability to process legacy “analog” waveforms such as frequency modulation (FM).

The basic structure of all SDRs is illustrated in Figure 1. The RF signals picked up by the antenna are conditioned prior to sampling. Ideally, this conditioning is little more than amplification by a low-noise amplifier (LNA). Given the current state of technology, the conditioning usually consists of additional tasks such as filtering and frequency translation to an intermediate frequency (IF). After conversion to the discrete-time domain, the desired frequency band is isolated using by a channelizer. The desired frequency band is translated to complex (or I/Q) baseband and resampled to a lower, more manageable sample rate. The most efficient SDR designs do not perform the functions of channelization, downconversion, and resampling separately, but rather perform these functions jointly by exploiting the properties of multirate processing of bandpass signals [1].

When the desired signal is a frequency modulated carrier, the complex baseband signal output by the channelizer/downconversion/resampler process must be demodulated using a discrete-time FM demodulator. At this point, the system designer is faced with an interesting design challenge: Is it best to mimic the continuous-time FM demodulator or to do something else? As Prof. fred harris pointed out, a DSP-based radio is **not** a digitized analog radio [2]. With this in mind, this paper explores the options available to a system

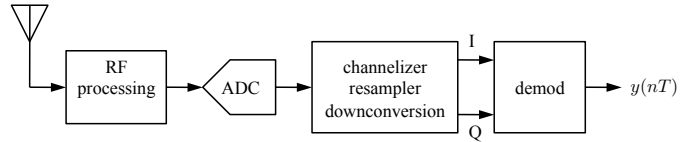


Fig. 1. Block diagram of a typical software-defined radio.

designer when the target platform is a field programmable gate array (FPGA).

We explore the performance of three options for demodulating a frequency modulated signal in discrete-time processing. For convenience, a sinusoidal modulating signal is used as the input to the FM modulator. The performance of these demodulators is quantified both as a signal processing system and as a digital system. As a signal processor, the performance is measured using the output signal-to-noise ratio (SNR) as a function of the input carrier-to-noise ratio (CNR). As a digital system, the performance is measured using FPGA area and maximum achievable clock speed. We show that efficient feed-forward and feedback discrete-time algorithms exist and can be implemented on an FPGA.

## II. DISCRETE-TIME FM

In general, the complex-baseband representation for a frequency modulated carrier is

$$s(t) = e^{j\phi(t)} \quad (1)$$

where  $\phi(t)$  is the instantaneous excess phase that is usually expressed as

$$\phi(t) = 2\pi f_d \int_0^t m(x) dx \quad (2)$$

where  $f_d$  is the frequency deviation with units cycles/s per unit amplitude and  $m(t)$  is the modulating signal. For sinusoidally modulated FM

$$m(t) = A_m \cos(2\pi f_m t) \quad (3)$$

so that the instantaneous excess phase is

$$\phi(t) = \beta \sin(2\pi f_m t) \quad (4)$$

where  $\beta = A_m f_d / f_m$  is the modulation index. The 90% (one-sided) bandwidth is given by the well-known Carson’s rule [3]

$$B_{90} = (\beta + 1) f_m. \quad (5)$$

This work was supported by the IUCRC Program of the National Science Foundation under Grant No. 0801876.

There are two approaches usually taken to demodulate FM: the limiter-discriminator and the phase lock loop (PLL) [3]. The limiter discriminator is based on a derivative operation followed by an envelope detector. These operations are preceded by a band-pass limiter to remove amplitude fluctuations. The PLL uses an FM modulator (voltage controlled oscillator) in a feedback arrangement. Both methods exhibit a threshold effect:<sup>1</sup> in general, the PLL demodulator has a lower threshold than the limiter-discriminator.

A discrete-time version of  $s(t)$  is formed by sampling (1) at  $T$ -spaced intervals. (The sample rate is  $1/T$ .) The  $n$ -th sample is

$$s(nT) = e^{j\phi(nT)} \quad (6)$$

where

$$\begin{aligned} \phi(nT) &= 2\pi f_d \int_0^{nT} m(x) dx \\ &\approx 2\pi f_d T \sum_{k=0}^{n-1} m(kT). \end{aligned} \quad (7)$$

Note that the product  $f_d T$  plays the role of the discrete-time frequency deviation with units cycles/sample per unit amplitude. Using  $m(kT) = A_m \cos(2\pi f_m T k)$  produces

$$\phi(nT) \approx 2\pi f_d T \sum_{k=0}^{n-1} A_m \cos(2\pi f_m T k) \quad (8)$$

$$\approx \frac{2\pi f_d T A_m}{2\pi f_m T} \sin(2\pi f_m T n) \quad (9)$$

where the second approximation is valid for  $2\pi f_m T \ll 1$  rads/sample. Retaining the definition for the modulation index  $\beta$ , the discrete-time version of complex-baseband FM signal is

$$s(nT) = e^{j\beta \sin(2\pi f_m T n)}. \quad (10)$$

Carson's rule for the 90% bandwidth still applies:

$$B_{90}T = (\beta + 1)f_m T \text{ cycles/sample.} \quad (11)$$

Discrete-time demodulators can be based on feed-forward processing or on feed-back processing as described below.

#### A. Feed-Forward FM Demodulator

The feed-forward demodulator is based on the definitions. Let the demodulator input be

$$r(nT) = e^{j\phi(nT)} + w(n) = I(nT) + jQ(nT) \quad (12)$$

where  $w(n)$  is a discrete-time additive noise sequence. If the variance of the additive noise is small relative to the power of the FM signal, then a good approximation of the instantaneous excess phase is

$$\hat{\phi}(nT) = \tan^{-1} \left( \frac{Q(nT)}{I(nT)} \right). \quad (13)$$

<sup>1</sup>The FM threshold is the input carrier-to-noise ratio below which the output signal-to-noise ratio is much worse. This effect can be observed in the SNR performance of the discrete-time PLL in Figure 9.

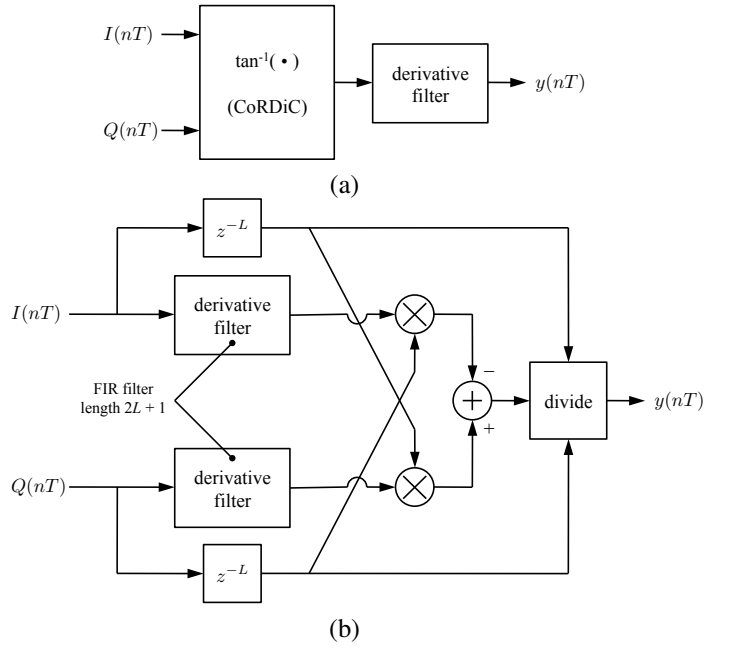


Fig. 2. Two feed-forward FM demodulator structures: (a) the arctangent/derivative process suggested by (14); (b) the derivative/divide suggested by (15).

The desired signal is the time-derivative of the instantaneous excess phase

$$y(nT) = \frac{d}{dt} \tan^{-1} \left( \frac{Q(nT)}{I(nT)} \right) \quad (14)$$

$$= \frac{I(nT)\dot{Q}(nT) - \dot{I}(nT)Q(nT)}{I^2(nT) + Q^2(nT)} \quad (15)$$

where  $\dot{I}(nT)$  means  $dI(t)/dt$  evaluated at  $t = nT$ . The same interpretation applies to  $\dot{Q}(nT)$ .

Equations (14) and (15) suggest the two demodulator structures illustrated in Figure 2. The system illustrated in Figure 2 (a) is based on a four-quadrant arctangent operation. In discrete-time processing, the arctangent is computed using the CoRDic operation [4], [5]. As a practical matter, the four quadrant arctangent operation must be followed by a phase “unwrap” operation (not shown) to remove phase discontinuities. The phase unwrap function,  $g(\cdot)$  may be expressed as

$$g(x) = [x + \pi \text{sign}(x)] \text{mod}(2\pi) - \pi \text{sign}(x) \quad (16)$$

when  $[x + \pi \text{sign}(x)] \text{mod}(2\pi) \neq 0$ . Note that  $g(0) = 0$  and  $g(x) = \pi$  when  $[x + \pi \text{sign}(x)] \text{mod}(2\pi) = 0$ . The derivative may be computed using an FIR filter as described in Chapter 3 of [6]. The system illustrated in Figure 2 (b) is based on the derivative and divide operations. Again, the derivative operations may be computed using a pair of identical FIR filters. The divide operation can be implemented with a dedicated hardware divider or using CoRDic. The relative performance merits of these two approaches is summarized in Section III.

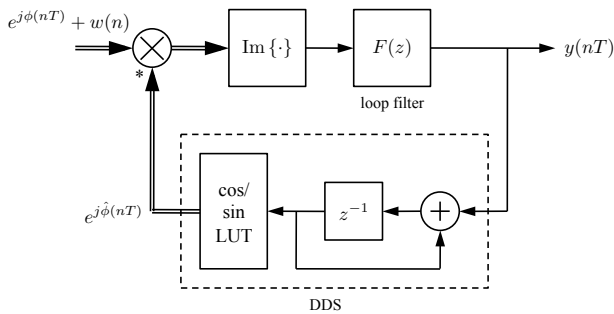


Fig. 3. The discrete-time PLL used as an FM demodulator.

### B. Feedback FM Demodulator: The Discrete-Time PLL

A discrete-time PLL, suitable for use as an FM demodulator with a complex-baseband input is illustrated in Figure 3. The system described in the next section uses a “proportional-plus-integrator” loop filter whose transfer function is

$$F(z) = K_1 + \frac{K_2}{1 - z^{-1}}. \quad (17)$$

This produces a second-order closed-loop system. The loop filter constants,  $K_1$ , and  $K_2$  determine the closed loop bandwidth and the damping constant as described in Appendix C of [6].

From a digital systems perspective, there are two main challenges with this design. First, the direct digital synthesizer (DDS) requires a high-speed look-up table (or ROM) to store samples of the cosine (and sine) function. The size of this table determines the accuracy of the DDS as described in Chapter 9 of [6]. The second challenge is the feedback structure makes achieving a high clock rate difficult.

## III. PERFORMANCE

To compare the resources and clock speed on a real FPGA, the FM demodulator designs were targeted to a Virtex4 FPGA (XC4VSX35-10FF668) on an XtremeDSP board. The designs were made in System Generator and run through synthesis, mapping, and place-and-route to determine the attainable clock rates and required resources. The demodulators were designed with speed in mind. This is not to say that these designs are pipelined to the maximum level (if there is one) but speed was given some preference over area.

### A. Arctangent-Differentiate System

The feed-forward demodulator of Figure 2 (a), here called the arctangent-differentiate system, was based on an “unwrapped” four-quadrant arctangent and a length-31 FIR derivative filter. The arctangent operation was realized by the Xilinx CoRDic Atan block, which is implemented using building blocks from the Xilinx blockset. An 18-stage CoRDic computation was “unrolled” to create a pipelined feed-forward processing unit. The filter realization was based on the Xilinx LogiCORE FIR Compiler V4.0. (The coefficients of the length-31 derivative filter were computed using the Blackman window following the technique described in Chapter 3 of

[6].) The phase unwrap function was implemented using basic logic blocks. In this design, the inputs are represented by 16-bit signed fixed point signals, with 14 bits to the right of the radix point. As the signals propagate through the design, the expected bit growth was observed. The multipliers were pipelined to achieve maximum speed. The required resources and clock rate performance are summarized in the second row of Table I.

### B. Differentiate-Divide System 1

The feed-forward demodulator of Figure 2 (b), here called the differentiate-divide system 1, was based on the same derivative filters described in Section III-A and a divide operation based on CoRDic. The CoRDic divider was implemented using the Xilinx CoRDic block which is based on building blocks from the Xilinx blockset. A 40-stage CoRDic computation was “unrolled” to create a pipelined feed-forward processing unit. The input words were 16-bit fixed point values with 14 bits to the right of the radix point. As before, the multipliers were pipelined to achieve maximum speed. The required resources and clock rate performance are summarized in the third row of Table I.

### C. Differentiate-Divide System 2

The differentiate-divide system 2 is an alternate implementation of the feed-forward demodulator of Figure 2 (b) where the divide operation is based on the Divider Generator 2.0 block. The divide operation was implemented through the Xilinx LogiCORE Divider v2.0. The derivative filters are identical to those described in Section III-B. The same finite precision arithmetic was also used. The required resources and clock rate performance are summarized in the fourth row of Table I.

### D. Feedback (PLL) System

The feedback demodulator based on the PLL of Figure 3 was based on a straight-forward use of addition and multiplication blocks. The DDS was based on two lookup tables (one each for the cosine and sine) consisting of 4096 12-bit words implemented in the on-chip block RAMs. The System Generator DDS block was not used so that loop delay could be carefully controlled. None of the usual precision-enhancing tricks (cf., Chapter 9 of [6]) were implemented. Consequently, the SNR performance (described below) suffered somewhat. The input words were 16-bit fixed point words with 14 bits to the right of the radix point. The loop filter coefficients and registers were 44-bit fixed point values with 40 bits to the right of the radix point. The required resources and clock rate performance are summarized in the fifth row of Table I.

### E. Comparison

The data presented in Table I demonstrate that the four designs considered present a variety of time/area trade-offs. The place each design occupies in this trade-off space is illustrated in Figure 4. Area is quantified using slices and time is quantified using the period of the equivalent *sample* period.

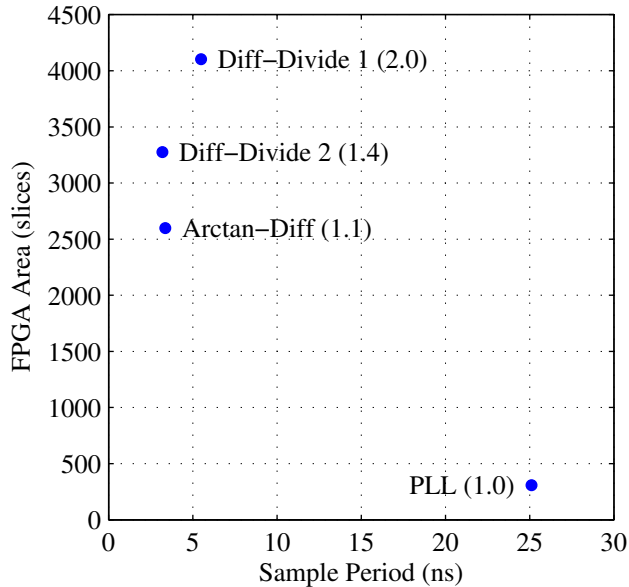


Fig. 4. Resource comparison for the four FM demodulators.

Sample period was used to remove any ambiguity regarding the relationship between clock rate and sample rate when pipelining is used. Also indicated are the time-area products with units slices-ns normalized to the lowest value (that of the PLL).

As expected, the feed-forward demodulators exhibit high throughput (small clock period) and moderate area usage. In contrast, the feedback demodulator requires very little area but, because of the feedback loop, cannot achieve as high a clock rate as the feed-forward options. The surprising result here is that from the signal processing perspective, conventional wisdom predicts the differentiate-divide 1 or differentiate-divide 2 as the “best” option. This would very likely be the case if the target platform were a programmable device such as a DSP. However, in custom hardware, the designer has the option to “unroll” the iterations associated with CoRDic to produce a pipelined feed-forward structure with excellent clock rate performance. It is simply too difficult (if not impossible) to achieve the same pipelining advantage in programmable processors. In the end, the area of the CoRDic arctangent is on the order of the area of a single multiplier.

In all cases, the area resources are quite small. This is a result of including only the basic demodulator functions in the comparison. In a real system, support for channelization and input/output must also be considered. In most SDR applications, the FM radio personality will be one of many radio instantiations on an FPGA of any practically usable size.

#### F. Signal Processing Considerations

The last dimension in the performance space is the signal-to-noise ratio performance of the demodulators. A test signal was used to perform the SNR tests. The test signal was

$$m(nT) = \cos(2\pi f_m T n).$$

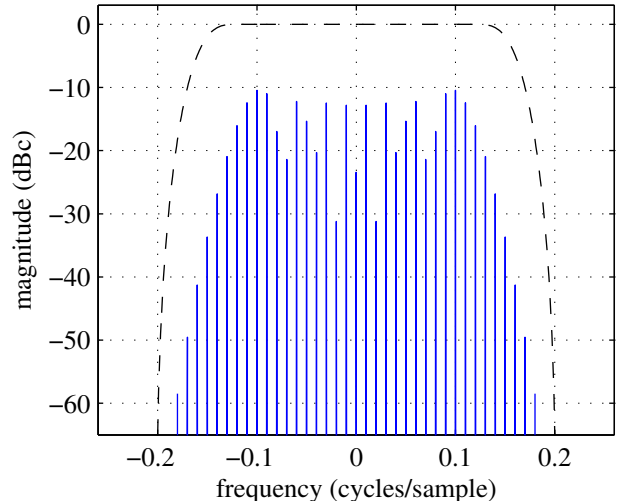


Fig. 5. The spectral representation of the discrete-time FM modulated signal (solid line) and the channelizing filter (dashed line) for  $f_d T = 0.01$  and  $f_m T = 0.115$ .

The modulation index was set to  $\beta = 11.5$  by using  $f_d = \beta f_m = 11.5 f_m$ . The motivation for using a large modulation index is to explore the performance of wideband FM, which is more challenging than narrowband FM. We also explored the performance relative to sample rate. This experiment showed that PLL performance improves as sample rate increases, whereas the performance of the feedforward FM demodulators is less dependent on sample rate, as long as the derivative filters are properly designed.

First, the case  $f_m T = 0.01$  cycles/sample was considered. In this case  $f_d T = 0.115$  cycles/sample. The discrete-time Fourier transform (DTFT) of the resulting FM signal is shown in Figure 5. Note the presence of the spectral lines whose heights are proportional to Bessel functions  $J_k(\beta)$  [3]. The bandwidth given by Carson’s Rule is

$$B_{90} T = (\beta + 1) f_m T = 0.125 \text{ cycles/sample} \quad (18)$$

which corresponds to the frequency at which the spectral lines are about 35 dB below the unmodulated signal. Also shown in Figure 5 is the DTFT of the filter applied at the modulator input. A length-51 FIR filter was used to represent the performance of polyphase channelizer that precedes the demodulator in most SDR applications — see Figure 1 and references [6, Chap. 9], [1], [7].

An example of the output of the arctangent-differentiate demodulator is illustrated in Figure 6 (a) for an input carrier-to-noise ratio (measured before the IF filter) of 10 dB. Observe the presence of large “spikes” caused by abrupt phase changes in the noisy signal. These spikes are the primary cause of signal-to-noise ratio (SNR) performance degradation in feed-forward FM demodulators. Motivated by this phenomenon, the output of the arithmetic processors in the FPGA were designed to saturate at a level approximately 1.5 times the amplitude of the noise-free output. An example of the output of the PLL demodulator is illustrated in Figure 6 (b). The dominant cause

TABLE I

A SUMMARY OF THE REQUIRED RESOURCES AND CLOCK RATE PERFORMANCE OF FOUR FM DEMODULATOR DESIGNS.

Design	Max. Clock Rate (MHz)	Slices/Total	Flip-Flops/Total	BRAMs/Total	DSP48s/Total
Arctan-Derivative	297.8	2,598/15,360 (16%)	3,492/30,720 (11%)	0/192 (0%)	29/192 (15%)
Derivative-Divide 1	182.1	4,103/15,360 (26%)	6,291/30,720 (11%)	0/192 (0%)	37/192 (19%)
Derivative-Divide 2	314.0	3,275/15,360 (21%)	4,402/30,720 (14%)	0/192 (0%)	34/192 (17%)
PLL	39.8	307/15,360 (1%)	117/30,720 (1%)	6/192 (3%)	2/192 (1%)

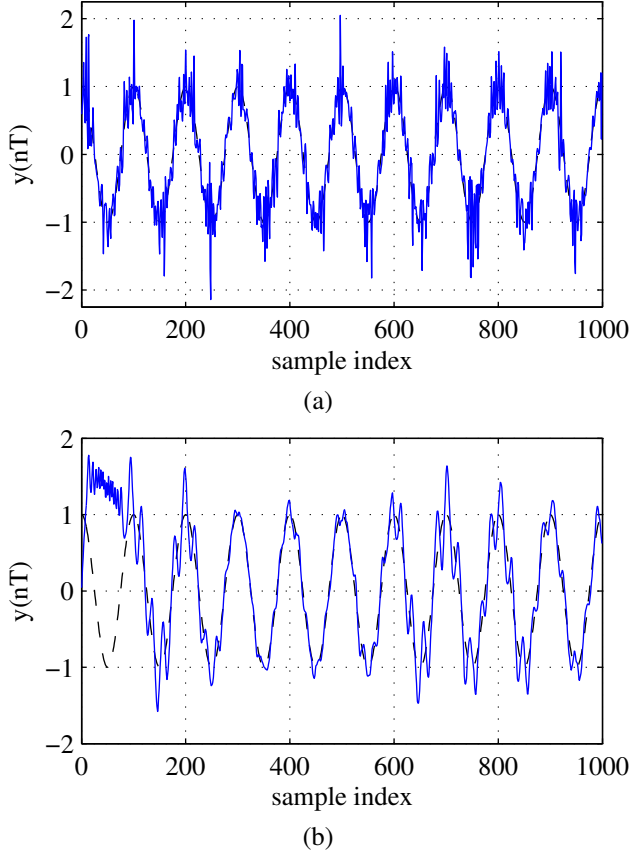


Fig. 6. Examples of distortion due to noise in the two types of demodulators: (a) “FM click” or “spike” distortion in the feed-forward FM demodulator; (b) “Cycle slips” in the PLL demodulator.

of SNR performance degradation in PLL-based demodulators is the phenomenon of “cycle slips” as shown.

To explore the influence of sample rate on performance, the sample rate was increased by 4 while keeping the modulation index  $\beta$  fixed at 11.5. This was accomplished using  $f_m T = 0.0025$  cycles/sample and  $f_d T = 0.02875$  cycles/sample. The 90% bandwidth using Carson’s rule is

$$B_{90} T = (\beta + 1) f_m T = 0.03125 \text{ cycles/sample} \quad (19)$$

An illustration of the resulting FM modulated signal and the length-101 channelizing filter are illustrated in Figure 7.

The SNR experiments were conducted using a combination of Matlab/Simulink and System Generator as illustrated in Figure 8. In Matlab/Simulink, the following steps were performed:

- 1) The test signal was generated and frequency modulated.

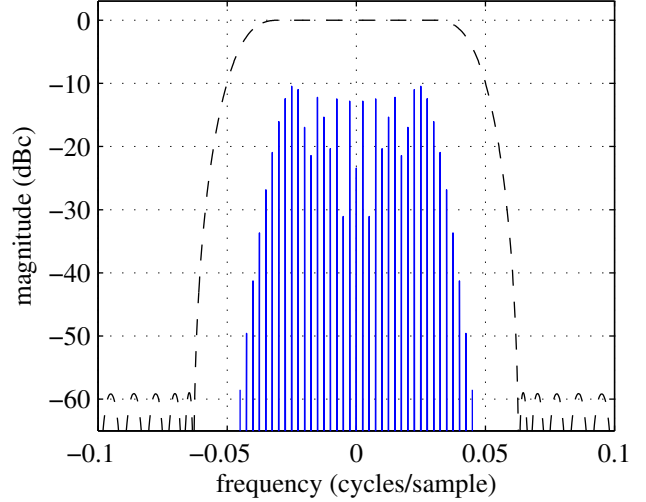


Fig. 7. The spectral representation of the discrete-time FM modulated signal (solid line) and the channelizing filter (dashed line) for  $f_d T = 0.0025$  and  $f_m T = 0.03125$ .

- 2) Noise samples were added to the FM signal. The noise was a sequence of uncorrelated zero-mean Gaussian random variables.
- 3) The noisy FM signal was filtered by the IF filter.

In System Generator, the noisy, filtered, FM signal was demodulated using the four demodulator designs described previously. The resulting demodulator output was transferred back to Matlab/Simulink for calculation of the output signal-to-noise ratio. The performance of the FM demodulators was simulated in System Generator to capture all the effects of finite precision and signal routing associated with the FPGA implementation.

The SNR performance of the four FM demodulators is for  $f_m T = 0.01$  and  $f_m T = 0.0025$  are plotted in Figures 9 and 10, respectively. The three feedforward demodulators used a length-31 derivative filter (although this was overkill for the  $f_m T = 0.0025$  case). The arctangent operation was implemented using an 18-stage CoRDic algorithm. The CoRDic-based divide operation used a 40-stage algorithm. For  $f_m T = 0.01$ , the PLL-based demodulator had a closed-loop bandwidth of 0.25 cycles/sample and a damping constant of 1. For  $f_m T = 0.0025$ , the PLL-based demodulator had a closed-loop bandwidth of 0.2 cycles/sample and a damping constant of 1.

Some general observations are in order. First, the SNR performance of the three feed-forward options is essentially the same. This implies that the improvements in FPGA

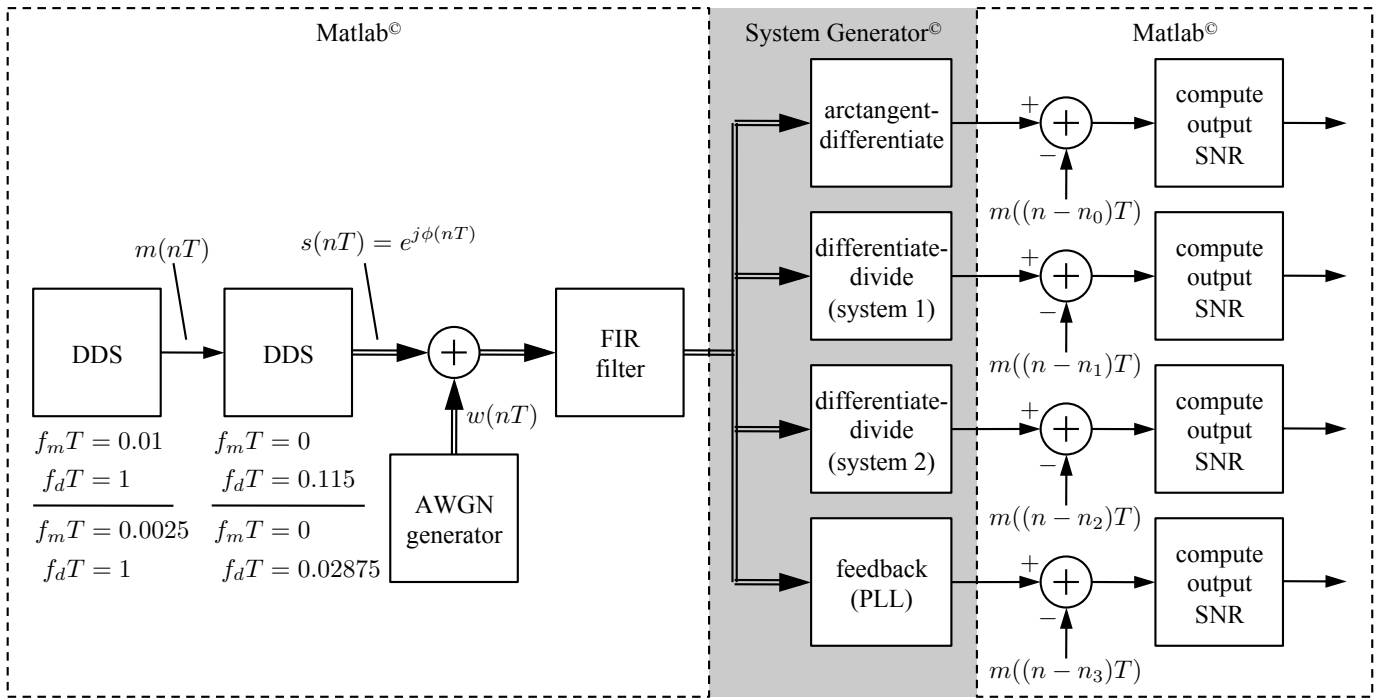


Fig. 8. A block diagram illustrating the simulations used to generate the performance results.

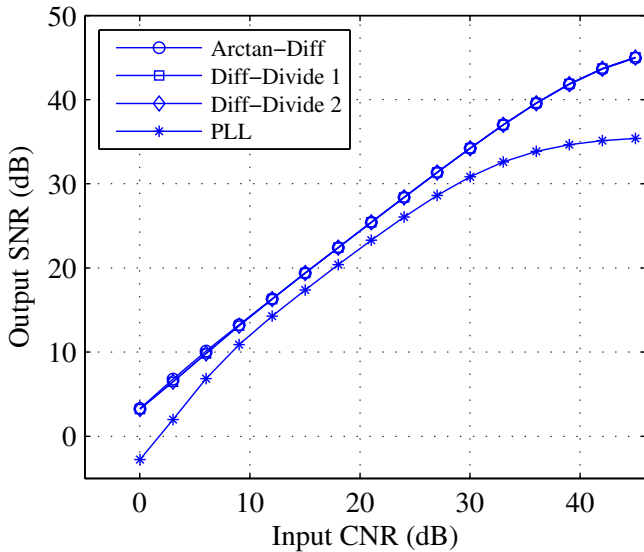


Fig. 9. The SNR performance of the four FM demodulators described in Section II: the differential/divide feed-forward demodulator of Figure 2 (b), the arctangent/differential (or CoRDIC/differential) feed-forward demodulator of Figure 2 (a), and the PLL-based feedback demodulator of Figure 3.

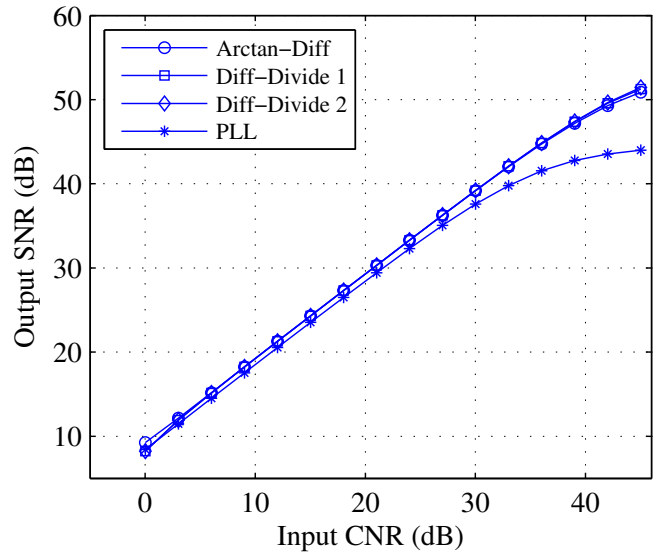


Fig. 10. The SNR performance of the four FM demodulators described in Section II: the differential/divide feed-forward demodulator of Figure 2 (b), the arctangent/differential (or CoRDIC/differential) feed-forward demodulator of Figure 2 (a), and the PLL-based feedback demodulator of Figure 3.

time/area (see Figure 4) are not achieved at the expense of SNR performance. Second, the SNR performance of the PLL FM demodulator is about 2 to 3 dB inferior to that of the feedforward demodulators for  $f_m T = 0.01$ . The performance gap closes to approximately 1 dB for  $f_m T = 0.0025$ . This behavior confirms the notion that the SNR performance of the PLL demodulator improves as the oversample factor increases.

(The differences between the SNR performance of the feed-forward demodulators in Figures 9 and 10 are due to the different IF filters used.) The SNR performance of the PLL demodulator “flattens” at high input CNR. This is due to quantization effects resulting from how the DDS look-up tables were implemented. At high CNR, the quantization effects dominate the SNR performance. Hence improving the

input CNR does improve output SNR. The point at which this phenomenon occurs improves with the use of more sophisticated DDS architectures.

#### IV. CONCLUSIONS

This paper explored the use of FPGAs to implement sampled-data FM demodulators for software-defined radios that must support “legacy waveforms.” Feed-forward and feedback structures were examined. The performance of these structures, both as a digital system and as a signal processor were quantified. The best feed-forward structure, in terms of the time/area trade-off was, surprisingly, the arctangent-differentiator structure. Simulation results showed that the hardware advantages, relative to the other feed-forward demodulators, were not achieved at the expense of SNR performance. The arctangent-differentiator and PLL demodulators have approximately the same time/area product and approximately the same SNR performance. However, the two occupy very different locations in the time/area trade-off space. In applications that need to maximize clock rate (minimize sam-

ple period), the arctangent-differentiator is the best choice. In applications that need to minimize area, the PLL demodulator is the best choice.

#### REFERENCES

- [1] f. harris, *Multirate Signal Processing for Communication Systems*. Prentice-Hall, 2004.
- [2] —, “A trap to avoid: A DSP based radio is NOT a digitized analog radio,” in *Proceedings of the International Symposium on Advanced Radio Technologies*. Boulder, CO: Institute for Telecommunication Sciences, 1998.
- [3] R. Ziemer and W. Tranter, *Principles of Communications*. Hoboken, NJ: John Wiley & Sons, 2009.
- [4] J. Volder, “The CORDIC trigonometric computing technique,” *IRE Transactions on Electronic Computers*, vol. 8, no. 3, pp. 330–334, September 1959.
- [5] J. Walther, “A unified algorithm for elementary functions,” in *Proceedings of the AFIS Spring Joint Computer Conference*, vol. 38. American Federation of Information Processing Societies, Inc., 1971, pp. 279–385.
- [6] M. Rice, *Digital Communications: A Discrete-Time Approach*. Upper Saddle River, NJ: Pearson Prentice-Hall, 2009.
- [7] f. harris, C. Dick, and M. Rice, “Digital receivers and transmitters using polyphase filter banks for wireless communications,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4, pp. 1395–1412, April 2003.