412TW-PA-19420

APSK SYMBOL TIMING AND CARRIER PHASE SYNCHRONIZATION ON AN FPGA IN A C-BAND TELEMETRY RECEIVER

4
1
2
T
W

JASON BAXTER
ERIK PERRINS

AIR FORCE TEST CENTER
EDWARDS AFB, CA

8/5/2019

Approved for public release; distribution is unlimited.
412TW-PA-19420

412TH TEST WING
EDWARDS AIR FORCE BASE, CALIFORNIA
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

APSK SYMBOL TIMING AND CARRIER PHASE SYNCHRONIZATION ON AN FPGA IN A C-BAND TELEMETRY RECEIVER

Jason Baxter
Electrical Engineering & Computer Science Department
University of Kansas
Lawrence, KS
j.baxter@ku.edu

Faculty Advisor:
Erik Perrins

ABSTRACT

This paper presents the implementation of a standard PLL-based timing and phase synchronization system on hardware using an FPGA. The synchronization system is shown to successfully recover a 16-APSK signal despite offsets in phase and frequency between the transmitter and receiver local oscillators. Furthermore, it is shown that system performance, in terms of symbol times required to achieve lock, is comparable to double-precision floating point simulations despite using fixed point numbers with as few as 5 fractional bits for most computations.

INTRODUCTION

The job of the receiver in a digital communication system is to extract the original information signal from the modulated (i.e. upsampled and filtered) signal. Most discussions of demodulation techniques assume perfect synchronization in timing and phase between the transmitter and receiver. Under this assumption, the received signal can be demodulated by simply filtering and downsampling, and any errors would be due to channel conditions. However, in real systems, the transmitter and receiver each have their own local oscillator for conversion between baseband and passband, and it is highly unlikely that the oscillators will be perfectly aligned in phase and frequency. In addition to any doppler shifts caused by the channel, this introduces two offsets between the transmitter and receiver: A timing offset, which means that the matched filter outputs are not being sampled at the correct instances, and a phase offset, which means that the received constellation is a rotated version of the original constellation used for modulation.

Because of these timing and phase offsets, a synchronization system that will recover the original timing and phase information is needed at the receiver. For this paper, the PLL-based synchronization techniques described in [1] are implemented on an FPGA. The next sections summarize these synchronization techniques. Then, a description of the hardware implementation is given, and hardware considerations are discussed. The results of the FPGA implementation are compared with Matlab simulations to confirm successful synchronization of a 16-APSK signal.

NOTATION

The notation used in this paper refers to signals within the synchronization system. The I and Q matched filter outputs at the k^{th} symbol time T_s are denoted $x(kT_s)$ and $y(kT_s)$ respectively. The I and Q components of a decision are denoted $a_I(k)$ and $a_Q(k)$. The upsampling factor of the transmitted data is denoted N.

SYSTEM DESCRIPTION

A. CARRIER PHASESYNCHRONIZATION

A carrier phase offset means that the oscillator at the receiver is not aligned in phase and/or frequency with the oscillator at the transmitter. This results in the matched filter output being out of phase with its nearest constellation point (i.e. the decision) as shown in Figure 1, even in a noiseless environment. The carrier phase synchronization system can be seen as a rotation of matched filter outputs in order to properly align them into a constellation. This rotation is achieved using the counter-clockwise rotation matrix shown in (1), where $\hat{\theta}$ is the estimated carrier phase offset.

$$\begin{bmatrix} x(kT_s) \\ y(kT_s) \end{bmatrix} = \begin{bmatrix} \cos(\hat{\theta}(k)) & \sin(\hat{\theta}(k)) \\ -\sin(\hat{\theta}(k)) & \cos(\hat{\theta}(k)) \end{bmatrix} \begin{bmatrix} x(kT_s) \\ y(kT_s) \end{bmatrix} \tag{1}$$

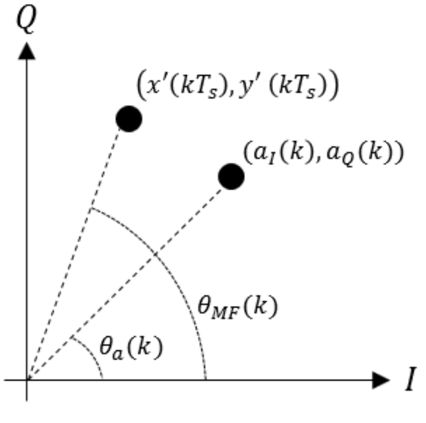

Figure 1: Graphical Interpretation of Phase Error

In order to estimate the carrier phase offset, the phase synchronization system used in this paper employs the phase-locked loop (PLL) as derived in [1]. Specifically, the second-order PLL shown in Figure 2 is used, where K_0, K_1, and K_2 are PLL gain constants. These constants depend on the desired loop bandwidth and loop damping factor. The direct digital synthesizer (DDS) in the discrete-time PLL replaces the VCO of an analog PLL, and the loop filter shown is the discrete-time proportional-plus-integrator (PPI) filter. The error signal input to the PLL can be seen geometrically as the difference between $\theta_{MF}(k)$ and $\theta_a(k)$ in Figure 1. If $\theta_{MF}(k)$ and $\theta_a(k)$ are the phases of the k^{th} rotated matched filter output and the k^{th} decision constellation point, then [1] shows that the maximum likelihood phase error detector (PED) is

$$\begin{aligned} e(k) &= \sin(\theta_{MF}(k) - \theta_a(k)) \\ &\propto y(kT_s)a_I(k) - x(kT_s)a_Q(k) \end{aligned} \tag{2}$$

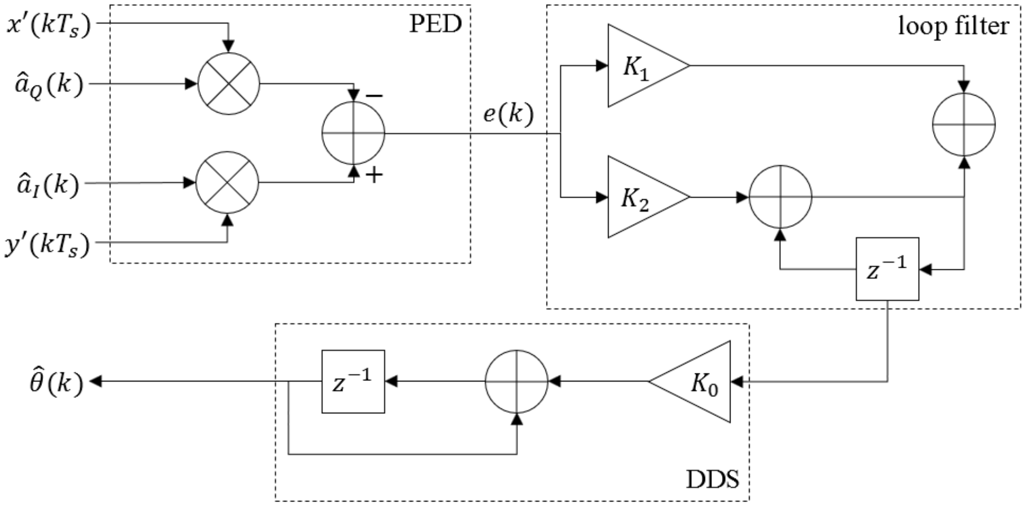

Figure 2: Second-Order PLL with PPI Filter

A block diagram of the entire carrier phase synchronization system is shown in Figure 3.

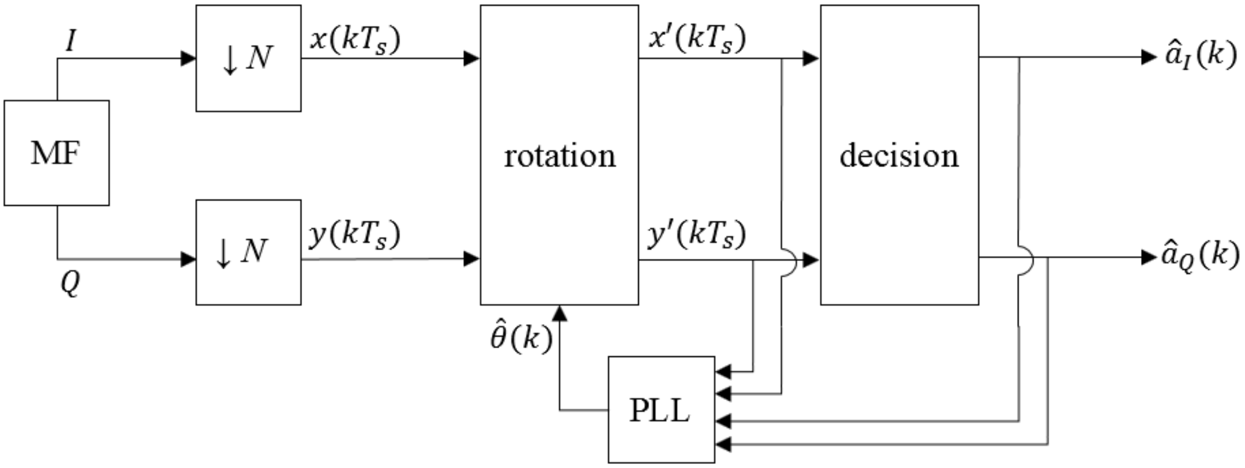

Figure 3: Phase Synchronization System

B. SYMBOL TIMING SYNCHRONIZATION

A symbol timing error occurs when the received waveform is sampled in between ideal sampling instants, i.e. when the eye diagram is not at its maximum opening as shown in Figure 4. The goal of a symbol timing synchronization system is to ensure that during each symbol time, there is a sample aligned with the eye diagram's maximum opening. One way of achieving this is to sample the received signal at a rate higher than the symbol rate and interpolate these samples. Because the timing offset is unknown, the interpolator must be dynamic to allow for the system to lock. In this paper, the Farrow piecewise parabolic interpolator is used with $\alpha = 1/2$, chosen for its divide-by-twos which can easily be implemented on hardware. Interpolation is controlled by the interpolator fractional interval, denoted μ.

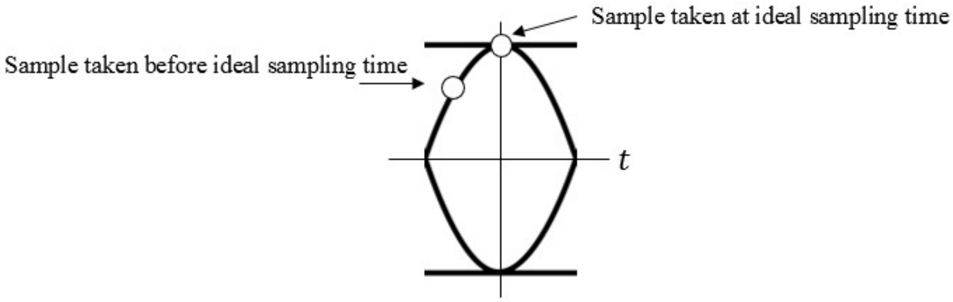
Sample taken at ideal sampling time
Sample taken before ideal sampling time
t

Figure 4: Graphical Representation of Timing Error

The value of μ is updated using a PLL, which consists of a timing error detector (TED) and a loop filter similar to the PLL described in the phase recovery section. The same second order PPI loop filter is used here as in the phase synchronization system. There are many different kinds of TEDs described in [1]; the one used in this paper is the zero-crossing TED (ZCTED). The ZCTED operates at 2 samples per symbol, and its output is zero when every other sample is time-aligned with the eye diagram's zero crossings, implying that the other samples are aligned with the eye diagram's maximum opening. The ZCTED error signal is shown in (3), where $\hat{\tau}$ is the timing offset.

$$e(k) = x((k-1/2)T_s + \hat{\tau})[a_I(k-1) - a_I(k)] + y((k-1/2)T_s + \hat{\tau})[a_Q(k-1) - a_Q(k)] \quad (3)$$

A block diagram of this symbol timing synchronization system is shown in Figure 5. The stars indicate the blocks which are enabled by a "strobe" signal. In this system, the strobe is set high when the modulo-1 decrementing counter underflows. Because the ZCTED operates at 2 samples per symbol, the counter is designed to underflow every 2 samples by decrementing by an average of $1/2$ every iteration. This is shown in (4), where η is the counter output and υ is the loop filter output.

$$\eta(k+1) = \left(\eta(k) - \upsilon(k) - \frac{1}{2}\right) \bmod 1 \quad (4)$$

The output of the loop filter adjusts the amount by which the counter decrements. When the strobe is high, η and υ lead to the computation of μ for the interpolator as

$$\mu(k) = \frac{\eta(k)}{\upsilon(k) + \frac{1}{2}} \quad (5)$$

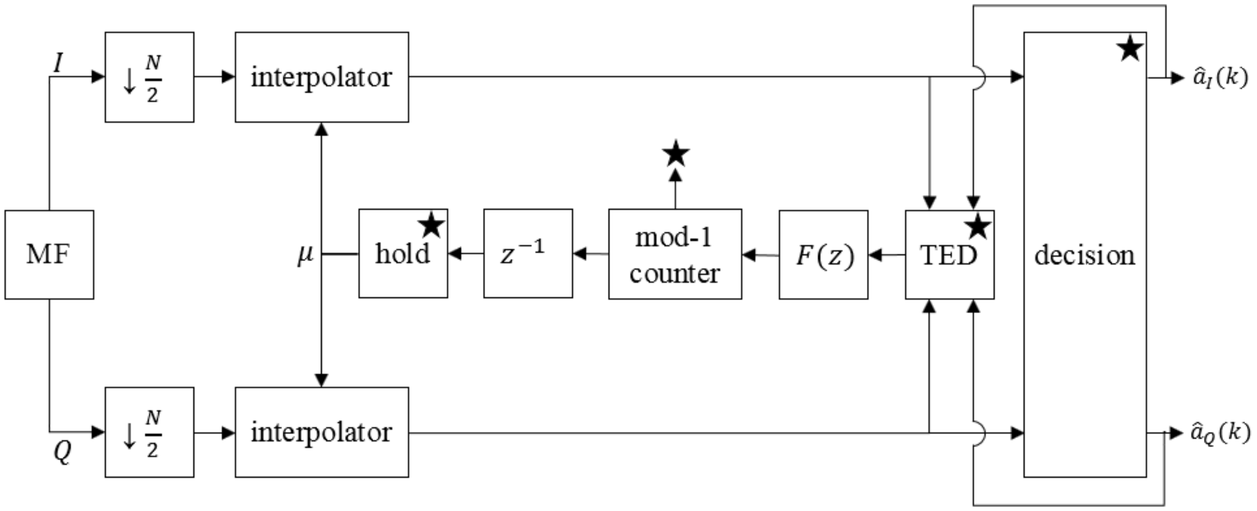

Figure 5: Timing Synchronization System

C. COMPLETE SYNCHRONIZATIONSYSTEM

A complete synchronization system consists of both the phase and timing synchronization systems described in the previous sections. Figure 6 shows the complete synchronization system, where the star indicates the strobe-enabled blocks as before. Dashed lines indicate signals of the phase synchronization system.

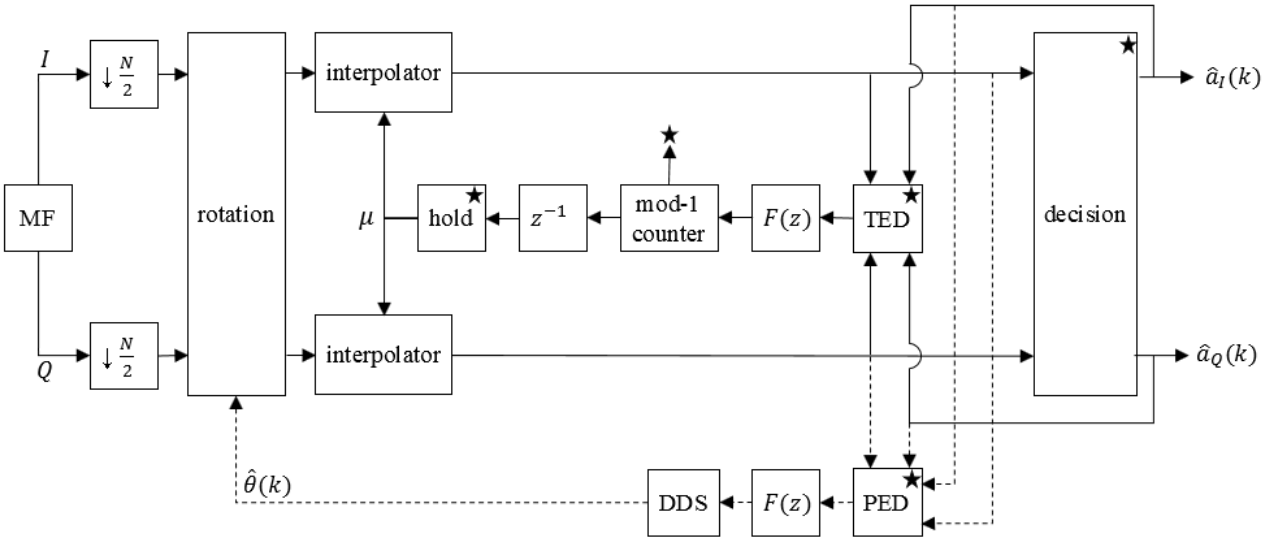

Figure 6: Complete Synchronization System

IMPLEMENTATION

A. SYSTEM PARAMETERS

As discussed in the previous sections, PLL design requires the selection of several constants depending on the desired system behavior. Tables 1 and 2 show the constants used for this paper. In addition to the constants K_0 and K_p, design also includes selection of the loop bandwidth B_nT and the damping factor ζ. The values of K_1 and K_2 are functions of these four values. For both the phase and timing systems, B_nT was set to 0.01 (normalized to symbol time) and ζ was set to $1/\sqrt{2}$. The values of K_0 and K_p and the equations for K_1 and K_2 were obtained from [1].

Table 1: Parameters for Phase Synchronization PLL

K_0	1
K_p	1
K_1	$2.667E-2$
K_2	$3.556E-4$

Table 2: Parameters for Timing Synchronization PLL

K_0	-1
K_p	2.68
K_1	$-9.950E-3$
K_2	$-1.327E-4$

Another design parameter was the pulse used for the matched filter in the receiver. The pulse is described in Table 3. This pulse was optimized for minimal peak to average power ratio.

Table 3: Matched Filter Description

shape	raised cosine
span	16 symbols
rolloff	0.4051
upsampling factor	32
window	Kaiser
window shape factor	2.8299

The data was randomly generated and mapped to the 16-APSK constellation shown in Figure 7 as per the DVB-S2 standard for rate 4/5 LDPC codes [2]. The average symbol energy was normalized to unity. A phase offset of $\pi/10$ radians and a frequency offset of $1E-5$ cycles per symbol were introduced in the oscillator at the receiver. Five zeros were padded at the beginning of the received data sequence to simulate a symbol timing offset. No noise was added to the data. All modulation operations and the baseband conversion at the receiver were performed in Matlab. The baseband received data was then loaded onto the FPGA memory, and processing began at the matched filter input. The data was loaded onto the FPGA in fixed point Q13 format. This was chosen to match

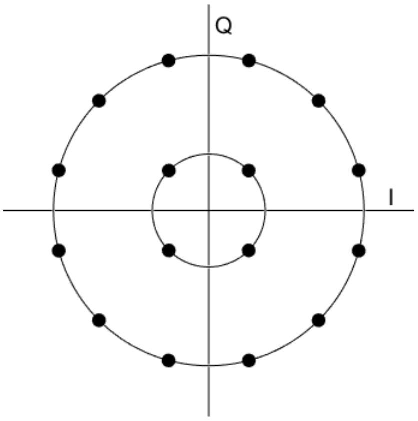

Figure 7: 16-APSK Constellation from [2]

the 14-bit ADC which will be used in the future to quantize the data used by the FPGA. The matched filter outputs and most subsequent computations were represented using 18 fractional bits. This would be an unnecessary amount of precision in a noisy environment, but it simplified the performance comparison against Matlab. The FPGA sampled the data at 1.25 Msps.

B. HARDWARE CONSIDERATIONS

Migrating the synchronization system from Matlab to the FPGA required attention to timing, bit allocation, and computational complexity. The FPGA uses a 200 MHz system clock, meaning that each clock cycle is 5 ns. For some parts of the system with a high bit precision requirement, such as the PLLs with small constant values, one clock cycle was not long enough to complete the computations, and delays had to be introduced. Another consideration was bit growth that resulted from certain computations. For example, the constellation itself could be represented using one integer bit because the I or Q value with the highest magnitude is±1.0916. However, the ZCTED error signal contains a subtraction of two I and Q constellation values. Therefore, it is possible for this subtraction to result in a number that would require two integer bits. A less obvious issue presented by bit growth involved the CORDIC used for the phase synchronizer's rotation operation. Because of the CORDIC rotation algorithm's inherent scale factor, it is possible for an iteration to result in a value outside the input's fixed point range.

Furthermore, it is well known that the division operation should be avoided on FPGAs when possible. However, it can be seen in (5) that the calculation of the interpolator fractional interval contains a division. This division was avoided by considering the timing PLL output υ to be much smaller than 1/2, allowing the quotient to be estimated with a simple multiply-by-two bit shift.

RESULTS

Before FPGA implementation, the system was scripted in Matlab to serve as the benchmark for synchronization performance on the FPGA. The results presented here show the two main signals used to observe performance: Figure 8 shows the output of the phase error detectors from Matlab simulation and FPGA implementation, and Figure 9 shows the value of μ from the timing synchronization loop in Matlab simulation and FPGA implementation. In addition to the PED output and μ, a 0 bit error rate was observed after the systems had locked.

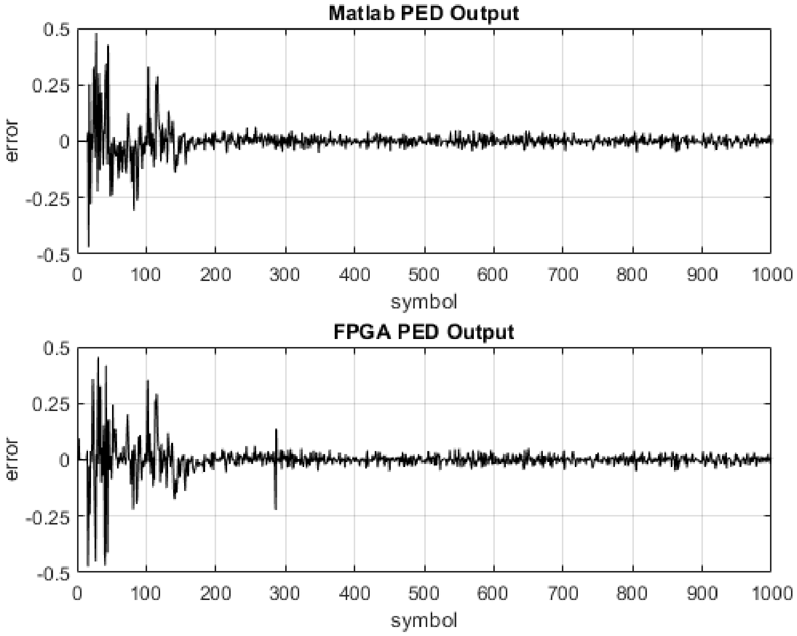

Figure 8: Phase Synchronization System Performance

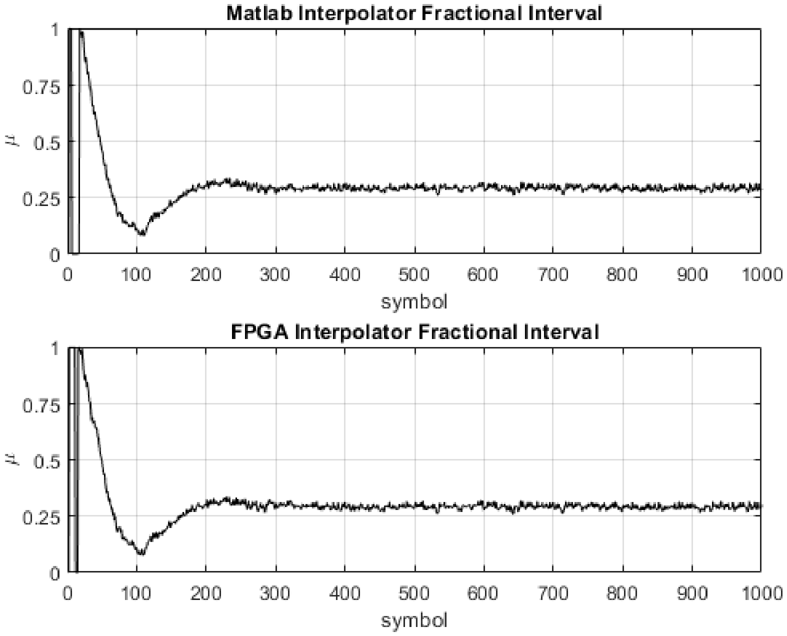

Figure 9: Timing Synchronization System Performance

ANALYSIS

The most important result in a synchronization system is the amount of time it takes to lock. These results indicate that the FPGA implementation of the system was able to recover the carrier phase and symbol timing information from the offset data in the same number of symbol times as the Matlab simulations. This is the best possible outcome in this scenario. The FPGA could not be expected to outperform the Matlab simulations due to the fixed point operations and limited bit precision. Additionally, it can be seen that the expected underdamped response was achieved given the damping factor.

It should be noted that although the results shown in Figure 8 and Figure 9 were obtained using 18 fractional bits for most computations, similar performance was observed using as few as 5 fractional bits. Using fewer fractional bits allows for faster computation times. This could eliminate the need for the extra delays inserted throughout the system to allow the hardware to complete lengthy multiplications.

CONCLUSION

Without proper synchronization between the transmitter and receiver, a communication system will not operate reliably. This paper described a system which processes the received data to align the samples in phase and frequency with the original data so the information can be recovered. The system was implemented on an FPGA and shown to successfully demodulate a received 16-APSK signal that had a carrier phase offset, an oscillator frequency offset, and a symbol timing offset.

FUTURE WORK

As described in [1], a phase synchronization system could potentially lock onto a constellation with the incorrect phase correction, resulting in a constant phase offset of the decision constellation. This phase ambiguity can be resolved with the unique word approach, where the receiver searches for a known sequence by comparing the decisions and each potential offset of the decisions to the unique word. The system implemented for this paper does not currently have this phase ambiguity resolution capability. Another advancement will be configuring the FPGA to accept real-time data through a host-to-card DMA transfer rather than reading from its ROM. The system will also be made to handle a dynamic scaling of the constellation to account for the automatic gain control of the telemetry receiver.

REFERENCES

[1] M. Rice, *Digital Communications: A Discrete Time Approach*. Pearson, 2009.

[2] European Telecommunications Standards Institute (ETSI), *Digital Video Broadcasting (DVB), Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2)*, 2006.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 09-08-2019	**2. REPORT TYPE** Technical Paper	**3. DATES COVERED** *(From - To)* October 21-24, 2019

4. TITLE AND SUBTITLE
Apsk Symbol Timing And Carrier Phase Synchronization On An Fpga In A C-Band Telemetry Receiver

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jason Baxter, Erik Perrins

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)
University of Kansas
1450 Jayhawk Blvd
Lawrence, KS 66045

8. PERFORMING ORGANIZATION REPORT NUMBER
412TW-PA-19420

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
412th Test Wing
195 E Popson Ave
Edwards AFB CA 93524

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release A: distribution is unlimited.

13. SUPPLEMENTARY NOTES
International Telemetering Conference (ITC), Las Vegas, NV

14. ABSTRACT
This paper presents the implementation of a standard PLL-based timing and phase synchronization system on hardware using an FPGA. The synchronization system is shown to successfully recover a 16-APSK signal despite offsets in phase and frequency between the transmitter and receiver local oscillators. Furthermore, it is shown that system performance, in terms of symbol times required to achieve lock, is comparable to double-precision floating point simulations despite using fixed point numbers with as few as 5 fractional bits for most computations.

15. SUBJECT TERMS
Synchronization, Field Programmable Gate Array(FPGA)

16. SECURITY CLASSIFICATION OF: Unclassified			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** 412 TENG/EN (Tech Pubs)
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	None	11	**19b. TELEPHONE NUMBER** *(include area code)* 661-277-8615