

FPGA Implementation of Burst-Mode Synchronization for SOQSPK-TG

June 2014

Final Report

Tom Young
SET Executing Agent
412 TENG/ENI
(661) 277-1071
Email: tommy.young.1@us.af.mil

Approved for public release; distribution is unlimited.

Test Resource Management Center (TRMC)
Test & Evaluation/ Science & Technology (T&E/S&T)
Spectrum Efficient Technology (SET)

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 04-06-2014		2. REPORT TYPE Technical Paper		3. DATES COVERED (From - To) 9/11 -- 8/14	
4. TITLE AND SUBTITLE FPGA Implementation of Burst-Mode Synchronization for SOQSPK-TG				5a. CONTRACT NUMBER: W900KK-11-C-0032	
				5b. GRANT NUMBER: N/A	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Erik Perrins				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University Of Kansas Center For Research 2385 Irving Hill RD, Lawrence KS 66045-7563				8. PERFORMING ORGANIZATION REPORT NUMBER 412TW-PA-14298	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Test Resource Management Center Test and Evaluation/ Science and Technology 4800 Mark Center Drive, Suite 07J22 Alexandria, VA 22350				10. SPONSOR/MONITOR'S ACRONYM(S) N/A	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) SET 2014-0043	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release A: distribution is unlimited.					
13. SUPPLEMENTARY NOTES CA: Air Force Flight Test Center Edwards AFB CA CC: 012100					
14. ABSTRACT In this paper, we present an FPGA implementation for synchronization of SOQPSK-TG in burst-mode transmissions. The system first detects arrival of new bursts, after which it estimates carrier frequency, carrier phase, and symbol timing offsets. Additionally, it is designed based on the synchronization algorithms developed for the iNET preamble. Here, we introduce some complexity reduction techniques in order to save chip area and to minimize latency. The implementation results are shown to be very close to the computer simulations in terms of estimation error variances and the overall bit-error rate (BER).					
15. SUBJECT TERMS Spectrum, Aeronautical telemetry, algorithm, bandwidth, Integrated Networked Enhanced Telemetry (iNET), Shaped Offset Quadrature Phase Shift Keying (SOQPSK), bit error rate (BER), burst-mode receiver synchronization					
16. SECURITY CLASSIFICATION OF: Unclassified			17. LIMITATION OF ABSTRACT None	18. NUMBER OF PAGES 11	19a. NAME OF RESPONSIBLE PERSON 412 TENG/EN (Tech Pubs)
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code) 661-277-8615

DISTRIBUTION LIST

Onsite Distribution

Number of Copies

E-Mail Digital Paper

Attn. Tom Young, SET Executing Agent
412 TENG/ENI
61 N. Wolfe Ave. Bldg 1632
Edwards, AFB, CA 93524

0 2 2

Attn: Erik Perrins
University Of Kansas Center For Research
2385 Irving Hill RD
Lawrence KS 66045-7563

1 0 0

Edwards AFB Technical Research Libaray
Attn: Darrell Shiplett
307 East Popson Ave, Bldg 1400
Edwards AFB CA 93524

0 2 2

Offsite Distribution

Defense Technical Information Center
DTIC/O
8725 John J Kingman Road, Suite 0944
Ft Belvoir, VA 22060-6218

1 0 0

U.S. ARMY PEO STRI Acquisition Center
Email to: kaitlin.lockett@us.army.mil
Attn: Kaitlin F. Lockett
12350 Research Parkway
Orlando, FL 32826

1 0 0

FPGA IMPLEMENTATION OF BURST-MODE SYNCHRONIZATION FOR SOQPSK-TG

Erik Perrins

Department of Electrical Engineering & Computer Science

University of Kansas

Lawrence, KS 66045

esp@ieee.org

ABSTRACT

In this paper, we present an FPGA implementation for synchronization of SOQPSK-TG in burst-mode transmissions. The system first detects arrival of new bursts, after which it estimates carrier frequency, carrier phase, and symbol timing offsets. Additionally, it is designed based on the synchronization algorithms developed for the iNET preamble. Here, we introduce some complexity reduction techniques in order to save chip area and to minimize latency. The implementation results are shown to be very close to the computer simulations in terms of estimation error variances and the overall bit-error rate (BER).

INTRODUCTION

The migration toward the integrated network enhanced telemetry (iNET) system introduces a fundamental physical layer challenge: how to acquire and lock onto—i.e., *synchronize with*—brief signal bursts at a low signal-to-noise ratio (SNR). In this paper, we begin by formulating this important synchronization problem. We then show how to detect the so-called start-of-signal (SoS) condition and how to roughly estimate its exact beginning. Once the location of the SoS is roughly known, we then formulate the remaining task as a three-way joint problem of estimating carrier frequency, carrier phase, and symbol timing. We provide synchronization algorithms that solve this problem, which are based on the data preamble for the iNET system. We take care in formulating these algorithms for use in FPGA implementations. We conclude by presenting a performance characterization of hardware prototypes of the synchronization algorithms.

SOQPSK-TG SIGNAL MODEL

In our model, we assume each burst starts with a preamble, which is comprised of L_0 symbols having a duration of $T_0 = L_0 T_s$ seconds where T_s is the symbol duration. It is immediately followed by the payload carrying the information symbols. The complex baseband SOQPSK signal during transmission of the preamble can be expressed as

$$s(t) = \sqrt{\frac{E_s}{T_s}} \exp\{j\phi(t; \alpha)\} \quad (1)$$

where E_s is energy per transmitted symbol. The phase of the signal $\phi(t; \alpha)$ is defined as

$$\phi(t; \alpha) = 2\pi h \sum_{i=0}^{L_0-1} \alpha_i q(t - iT_s) \quad (2)$$

where α_i is the transmitted ternary symbol, i.e. $\alpha_i \in \{-1, 0, 1\}$, and $h = 1/2$ is the modulation index. The waveform $q(t)$ is the *phase response* of SOQPSK and in general is represented as the integral of the *frequency pulse* $g(t)$ with a duration of LT_s . There are currently two different versions of SOQPSK defined by their own frequency pulses. The first one known as the SOQPSK-MIL [1] is a full-response ($L = 1$) scheme with a rectangular-shaped frequency pulse. The second form is the telemetry group version [2], i.e. SOQPSK-TG, which is partial-response ($L = 8$) with a custom frequency pulse. According to the CPM definition, $q(t)$ is zero for $t < 0$ and is $1/2$ for $t > LT_s$.

The SOQPSK modulator can be characterized as a precoder connected to a CPM modulator. The precoder converts information bits $a_i \in \{0, 1\}$ to ternary symbols by means of

$$\alpha_i = (-1)^{i+1} (2a_{i-1} - 1)(a_i - a_{i-2}) \quad (3)$$

in order to impose OQPSK-like characteristics on the CPM signal. In the following, we will perform our analysis based on $\{\alpha_i\}$ because the preamble is fixed and known in terms of the ternary symbols. Note that the precoder does not change the rate of input bits, and hence, $T_b = T_s$.

Assuming transmission over an AWGN channel, the complex baseband representation of the received signal is

$$r(t) = \sqrt{\frac{E_s}{T_s}} e^{j(2\pi f_d t + \theta)} e^{j\phi(t-\tau; \alpha)} + w(t) \quad (4)$$

where θ is the unknown carrier phase, f_d is the frequency offset, τ is the timing offset, and $w(t)$ is complex baseband AWGN with zero mean and power spectral density N_0 . The transmitted data symbols are denoted by $\alpha = [\alpha_0, \alpha_1, \dots, \alpha_{L_0-1}]$. Our known preamble is implicit in the definition of $s(t)$. Prior to signal detection, we need to estimate the synchronization parameters. The first step in synchronization is called frame synchronization in which the location of the start-of-signal (SoS) is determined within one symbol duration. Next, we jointly estimate frequency offset, phase offset and symbol timing. Thus, we assume $-T_s/2 < \tau < T_s/2$ when we are dealing with symbol timing estimation.

The proposed preamble for iNET has a length of $L_0 = 128$. This preamble is periodic and it consists of repeating a sequence of 16 ternary symbols 8 times as follows.

$$\left. \begin{aligned} \alpha_k &= 1, & 1, & 1, & 1, & 1, & 1, & 0 \\ \alpha_{k+8} &= -1, & -1, & -1, & -1, & -1, & -1, & 0 \end{aligned} \right\} \text{ for } k = 0, \dots, 7. \quad (5)$$

SYNCHRONIZATION ALGORITHM

As mentioned earlier, the first stage is frame synchronization. The frame synchronization algorithm in this work is based on the data-aided synchronization algorithm of [3], which is for general CPM signals. We only need to replace h , $q(t)$ and the data symbols with their SOQPSK-TG counterparts.

The first step in frame synchronization is called the SoS detection, where the algorithm decides on the presence of the preamble in the received signal using the following test:

$$L_D[n] \triangleq \sum_{d=1}^D \left| \sum_{k=n}^{n+Np-d-1} r[k] r^*[k+d] s^*[k-n] s[k+d-n] \right| \geq \gamma_D \quad (6)$$

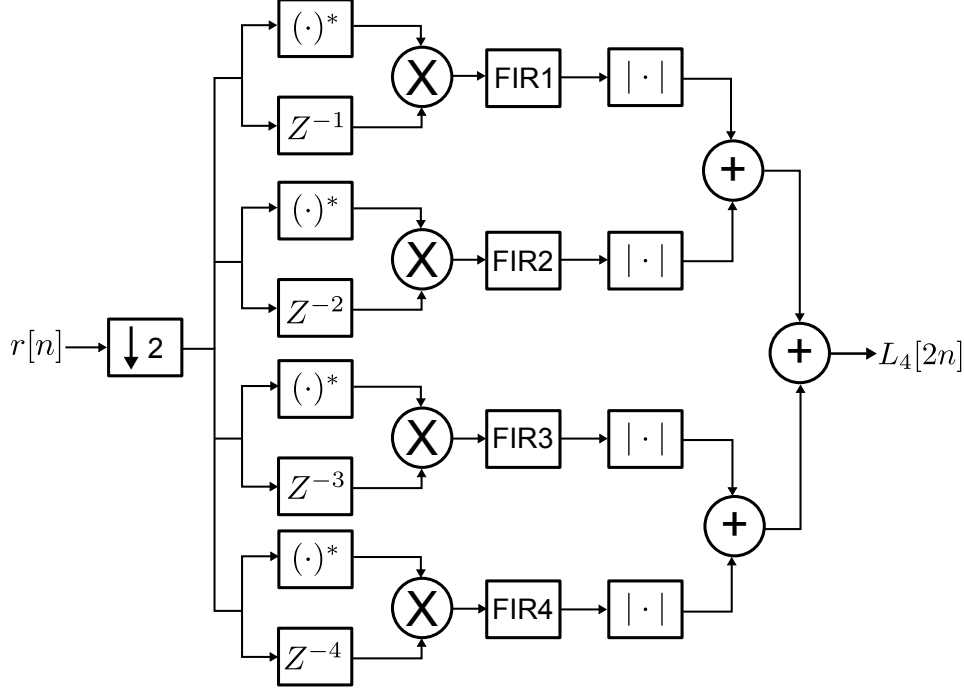


Figure 1: The frame synchronization data path.

where $N_p = 128N$ is the number of samples in the preamble when $r(t)$ is sampled at N samples per symbol. Additionally, $s[n]$, for $0 \leq n < 128N - 1$, are the known samples of our preamble. In the above, $1 \leq D' < N_p$ is a design parameter to control the complexity. The above test is performed on a sliding window over the received samples, and the SoS is detected when it becomes greater than the test threshold γ_D . The second step in frame synchronization is called the SoS estimation, where the exact location of the SoS is estimated by finding the maximum of (6) over a window of N_w samples after it crosses the threshold.

A high level block diagram of (6) is depicted in Figure 1. Based on our MATLAB simulations, we determined that $D = 4$ performs close to $D = N_p$. Consequently, the frame synchronization algorithm requires us to implement four complex finite impulse response (FIR) filters. We denote these filters by $\text{FIR}d$ for $1 \leq d \leq 4$. The input to $\text{FIR}d$ filter is $r[n]r^*[n + d]$, or equivalently $r[n]r^*[n - d]$. Additionally, the coefficients of the d -th filter are $C_i^d = s^*[i]s[i + d]$ for $0 \leq i < 128N - d$. We are able to generate the left hand side of (6) by adding the absolute value of the FIR filters' outputs. As (6) suggests, we have a new value for $L_4[n]$ with every incoming sample. This value is compared to γ for the SoS detection.

The SoS estimator observes $L_4[n]$ after it crosses the threshold for a duration equivalent to the preamble length, i.e. $N_w = N_p$. This is the uncertainty window in which we expect the SoS. The SoS estimator simply returns the index of the peak in the uncertainty window. Finally, the input samples are buffered to accommodate the latency introduced by the frame synchronization circuit and the search window for the SoS.

The hardware complexity of the frame synchronization block is proportional to the number of samples per preamble, and is dominated by the FIR filters. Therefore, the complexity can be reduced by decimating $r[n]$. In our implementation in Figure 1, we found that a decimation factor of 2 when $N = 2$ results in satisfactory performance at E_s/N_0 as low as 1 dB. More importantly, this rate of the input signal allows

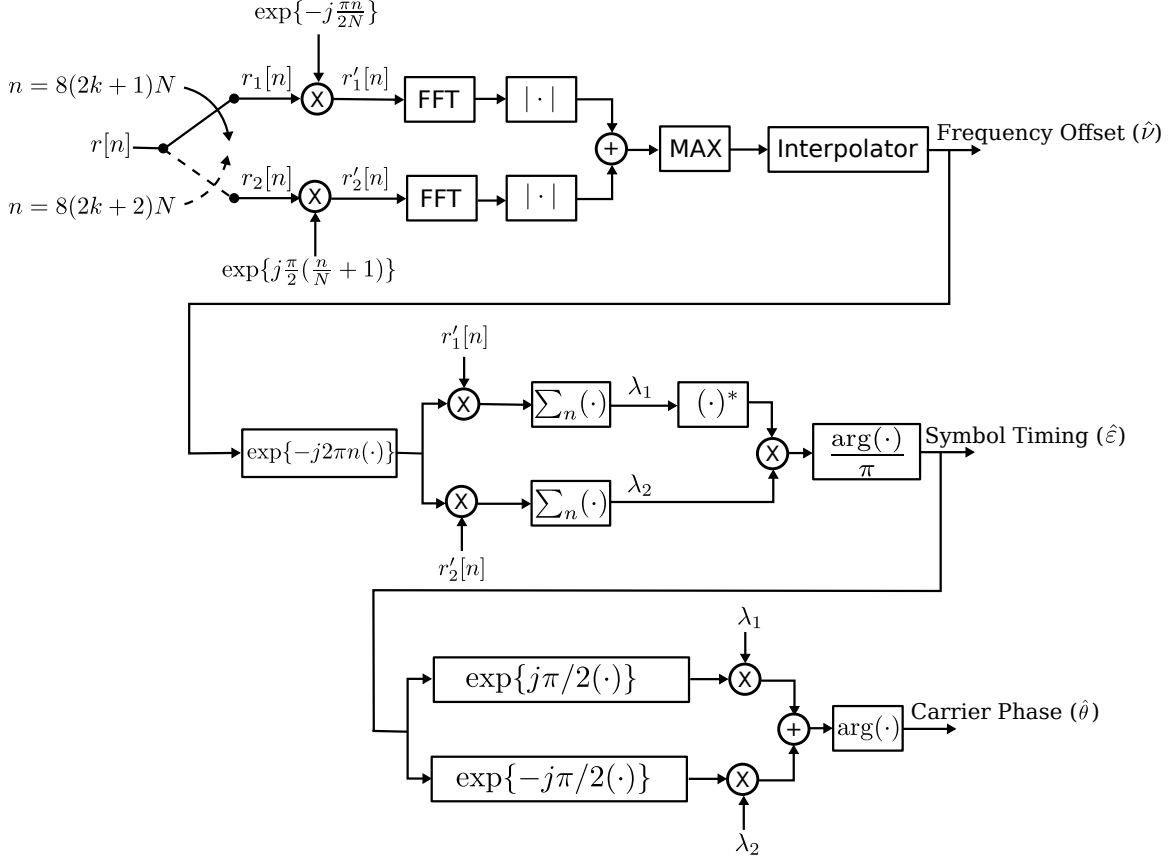


Figure 2: The joint frequency, timing and phase estimator.

the FIR filter coefficients to be selected from $\{-1, 1, -j, j\}$, which reduces the filter multipliers into multiplexers. These coefficients are exact for the majority of the preamble except for the transition points where we truncate the filter coefficients to the above values.

The next stage in our synchronization algorithm is carrier and symbol timing recovery based on the joint ML estimator of [4]. A high level block diagram of this algorithm is presented in Figure 2.

The first step in the joint estimator is frequency estimation. First, $r[n]$ is demultiplexed at the time instants shown in Figure 2 where $0 \leq k < 8$. This results in $r_1[n]$ and $r_2[n]$. It is assumed that $r[n]$ is initially connected to $r_1[n]$. Moreover, $r_2[n]$ is fed with zero samples when $r[n]$ is connected to $r_1[n]$ and vice versa. The frequency estimator employs two fast Fourier transform (FFT) blocks. Prior to the computation of the FFTs, these blocks zero pad $r_1'[n]$ and $r_2'[n]$ by a factor of K_f in order to improve the frequency resolution. Each of the FFT blocks generates $128NK_f$ samples, which correspond to discrete frequencies separated by $\frac{1}{128K_f}$ of the sampling frequency. Our investigations show that $K_f = 2$ delivers satisfactory frequency resolution. The maximum of the magnitude of the frequency domain samples correspond to the frequency offset normalized to the sampling frequency. The frequency offset estimator is further improved by using a Gaussian interpolator, i.e.,

$$\hat{\nu} = \hat{\nu}_0 + \frac{1}{2K_fNL_0} \frac{\log X(\hat{\nu}_{-1}) - \log X(\hat{\nu}_1)}{\log X(\hat{\nu}_{-1}) + \log X(\hat{\nu}_1) - 2\log X(\hat{\nu}_0)} \quad (7)$$

where $\hat{\nu}_0$ represents the maximizing frequency resulting from FFT operations, and $X(\cdot)$ is the output of the

Resource Type	Number	Utilization Ratio
Slices	5,783	33%
Block RAMs	26	17%
DSP48	21	32%

Table 1: FPGA utilization results

adder. $\hat{\nu}_{-1}$ and $\hat{\nu}_1$ denote the discrete frequency components immediately before and after $\hat{\nu}_0$ respectively.

The next stage is the symbol timing estimation where $\hat{\nu}$ is used to remove the frequency offset from $r'_1[n]$ and $r'_2[n]$ for $0 \leq n < 128N$. Two accumulators are then employed to compute λ_1 and λ_2 from $r'_1[n]$ and $r'_2[n]$ respectively. Finally, the angle of $\lambda_1^* \lambda_2$ divided by π results in the symbol timing estimate normalized by the symbol duration, which is denoted by $\hat{\epsilon}$. Therefore, the symbol timing estimator's range is limited within $\pm T_s$.

The last stage is computation of the carrier phase offset. This stage requires λ_1 , λ_2 , and $\hat{\epsilon}$ from the symbol timing estimator. Similarly, the argument function has to be realized.

Similar to the frame synchronization block, we are able to reduce the hardware size by decimating $r[n]$ by a factor of 2. In particular, it allows us to perform smaller FFTs, which have smaller latency. The side effect of this operation is the reduction in the frequency estimation range by 2. Additionally, we need to utilize a low pass filter (LPF) before the decimator in order to avoid aliasing. In our work, we implement the LPF by averaging three consequent samples at the rate of $N = 2$. This configuration is not an ideal LPF and will cause distortion if the frequency offset is high. However, we assume that the frequency offset is reasonably smaller than the bit rate in our application. Finally, it is noted that this LPF filter has a delay of T_s , and hence, the estimated symbol timing can be directly applied to the un-decimated signal for timing recovery and demodulation.

FPGA IMPLEMENTATION

In order to develop an exact representation of the hardware, we developed a bit-precise MATLAB model to determine the bit-widths in our design. Eight bits of precision are used to represent the received signal. This consists of one sign bit, three integer bits and four fractional bits, when the SOQPSK signal has an amplitude equal to one. Based on this assumption and our MATLAB model, we identified suitable bit-widths for the internal signals. The frequency estimate is represented using 14 bits to cover the range of $[-0.5, 0.5)$ when it is normalized to the symbol rate. Both symbol timing and phase estimates are signed values represented using 8 bits including 6 fractional bits. The symbol timing is normalized to T_s and the phase estimate is normalized to π .

The proposed burst-mode architecture is written in VHDL and verified using Modelsim. The VHDL design is implemented on a Xilinx Virtex 5 110xt FPGA with a speed grade of -1. The implementation results are presented in Table I. It is seen that our implementation consumes about of third of the FPGA. A great amount of savings in terms of FPGA area is due to the decimation of the received signal, which leaves us enough room for future implementation of the demodulator and the decoder. Furthermore, the FPGA is capable of running at up to 100.878 MHz.

We study the synchronization performance of our FPGA implementation using test vectors generated by MATLAB. First, we investigate the performance of the joint frequency, symbol timing, and phase estimator by providing the FPGA with preambles that have known frequency, timing, and phase offsets, in addition to the additive white Gaussian noise (AWGN). The estimated values from the FPGA are sent

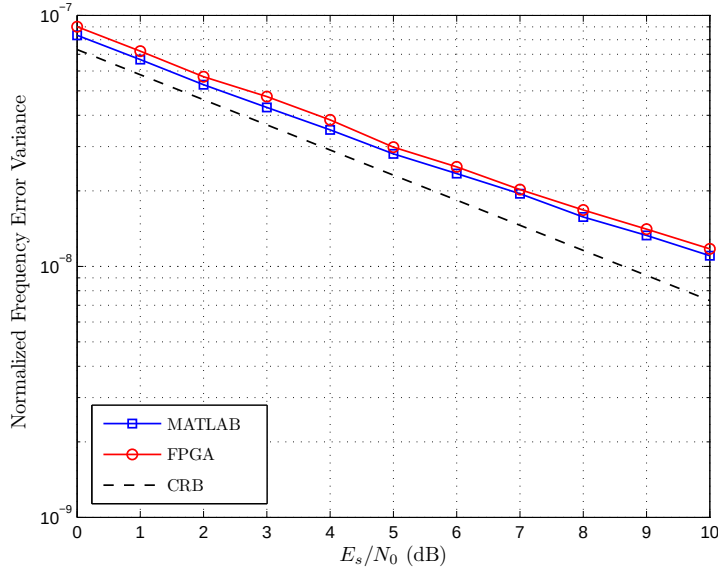


Figure 3: The normalized frequency error variance of the joint ML estimator. The frequency is normalized to the symbol rate.

back to the host PC and are compared with their original values. In this setup, we bypass the frame synchronization step in order to solely observe the performance of the joint estimator. The error variances corresponding to the frequency, timing, and phase are plotted in Figures 3, 4, and 5 respectively. In this set of plots, we run the simulations for 20,000 bursts at each signal-to-noise ratio (SNR) value. Moreover, we provide estimation results from MATLAB where no simplification or quantization is present. The Cramér-Rao bound (CRB) is also included, which represents the lower bound on the estimation error variance. It is observed that the FPGA implementation performs very close to the MATLAB results as far as frequency and phase estimations are concerned. However, we see some degradation in the symbol timing estimate. This is mainly due to the down-sampling and our non-ideal LFP, which cause signal distortion. Despite this, our simulations show that SOQPSK-TG's demodulation is almost intact for timing errors as large as $T_s/4$, which is quite a bit larger than the timing error standard deviation delivered by our design.

The overall performance of our system is tested by simulating a burst-mode receiver where the FPGA performs the synchronization task and the demodulation is carried out in MATLAB on the host PC. Each burst consists of the 128-bit known preamble and 6204 random bits as the payload. In order to emulate a burst-mode scenario, we add a guard time of 200 bit intervals between bursts in which only AWGN is present. However, the duration of this guard interval is unknown to the receiver. Moreover, we introduce a random frequency offset (up to 0.2 times symbol rate), a random carrier phase offset, and a random symbol timing offset. The generated signal is sent to the FPGA and the detected bursts along with the estimated values are sent back to the host PC for demodulation and bit error rate (BER) computation. The BER results for this setup are plotted in Figure 6. The results are also compared against two other scenarios where the synchronization is performed in MATLAB as well as a perfect synchronization scenario, i.e. where only AWGN is present. It is observed that the FPGA synchronization has almost identical performance to the MATLAB results. This validates our simplifications and choice of proper bit-widths in VHDL. Furthermore, we observe an SNR loss of less than 0.5 dB in all regions. This small SNR loss is

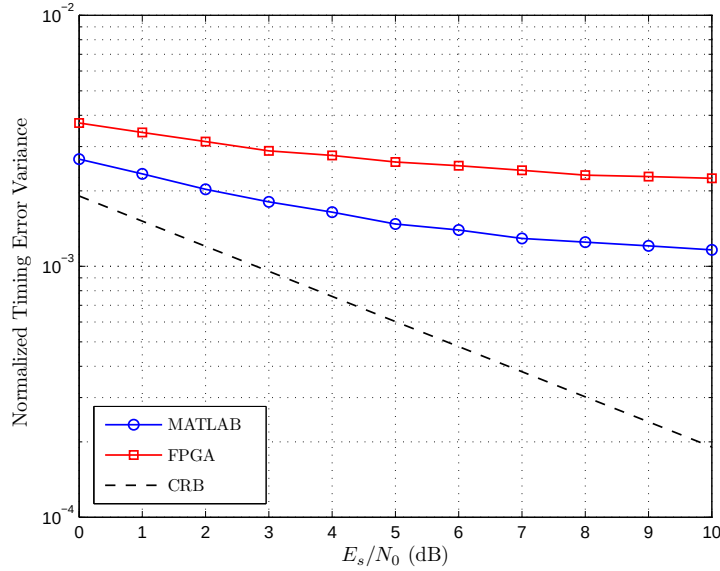


Figure 4: The normalized timing error variance of the joint ML estimator. The timing error is normalized to the symbol duration.

of great importance in the low SNR region because synchronization does not become the limiting factor for modern error correction schemes, which are capable of correcting error bits at $E_s/N_0 = 1$ dB.

CONCLUSIONS

In this paper, we have described an FPGA implementation for burst-mode synchronization of SOQPSK-TG. The proposed architecture has a feedforward structure and is designed based on the synchronization preamble of iNET. It detects the arrival of a new burst after which it estimates frequency offset, symbol timing, and carrier phase. Our test results demonstrate that the error variances of the synchronization parameters generated by the hardware are very close to that of our MATLAB implementation. Additionally, we simulated a burst-mode SOQPSK-TG receiver in which the synchronization is executed on the FPGA. The BER results of such a system shows an SNR loss of less than 0.5 dB compared to a receiver with perfect synchronization.

REFERENCES

- [1] M. J. Dapper and T. J. Hill, "SBPSK: a robust bandwidth-efficient modulation for hard-limited channels," in *IEEE Military Communications Conference Proceedings*, (Los Angeles, CA), pp. 458–463, 1984.
- [2] Range Commanders Council Telemetry Group, Range Commanders Council, White Sands Missile Range, New Mexico, *IRIG Standard 106-00: Telemetry Standards*, [online] ed. (Available: <http://www.ntia.doc.gov/osmhome/106.pdf>).
- [3] E. Hosseini and E. Perrins, "Timing, carrier, and frame synchronization of burst-mode CPM," *Communications, IEEE Transactions on*, vol. 61, pp. 5125–5138, Dec 2013.
- [4] E. Hosseini and E. Perrins, "Synchronization of SOQPSK-TG in burst-mode transmissions," in *International Telemetry Conference Proceedings*, Oct. 2013.

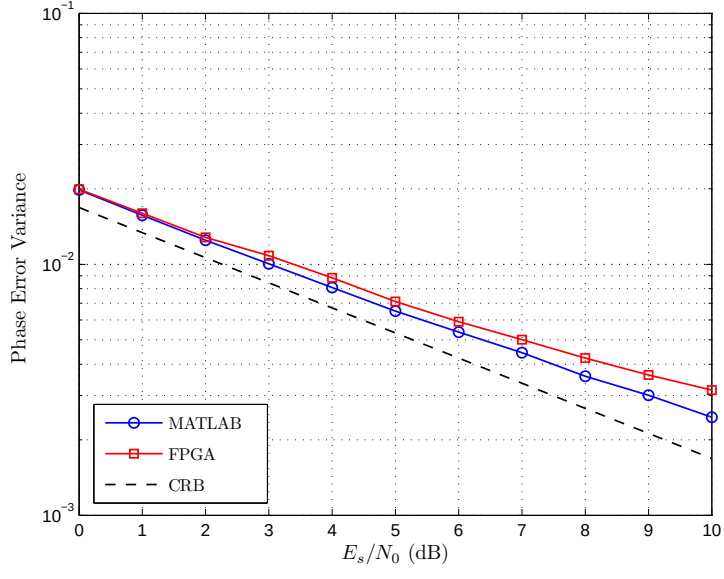


Figure 5: The joint frequency, timing and phase estimator

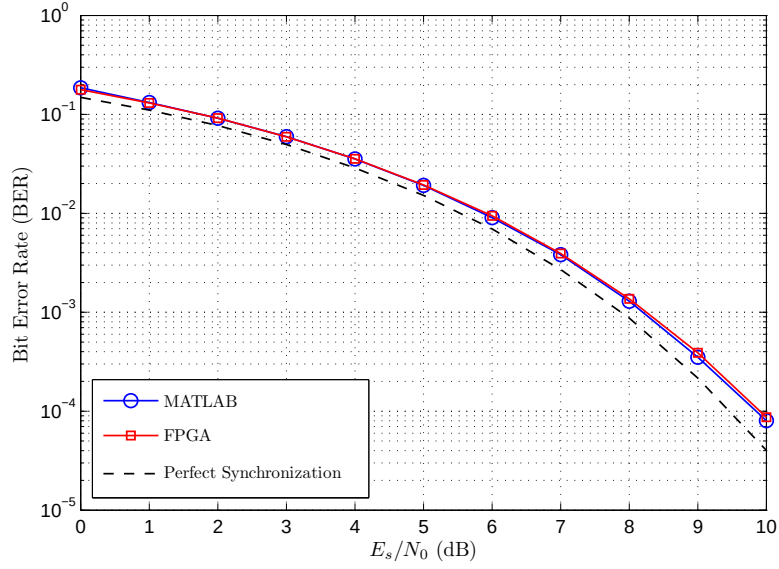


Figure 6: The BER performance of a burst-mode SOQPSK-TG receiver that employs our synchronization system.