

# Robust Sampling Clock Recovery Algorithm for Wideband Networking Waveform of SDR

Muhammad Zeeshan<sup>1</sup>, Shoab Ahmed Khan<sup>2</sup>

<sup>1,2</sup>College of Electrical & Mechanical Engineering, National University of Sciences & Technology, Rawalpindi, Pakistan  
<sup>1</sup>ranazeeshan@ceme.nust.edu.pk, <sup>2</sup>shoabak@ceme.nust.edu.pk

**Abstract:** A novel technique for sampling clock recovery in a wideband networking waveform of a software defined radio is proposed. Sampling clock recovery is very important in wideband networking radio operation as it directly affects the Medium Access adaptive time slot switching rate. The proposed Sampling clock recovery algorithm consists of three stages. In the first stage, Sampling Clock Offset (SCO) is estimated at chip level. In the second stage, the SCO estimates are post-filtered to improve the tracking performance. We present a new post-filtering method namely Steady-State State-Space Recursive Least Squares with Adaptive Memory (S4RLSWAM). For the third stage of SCO compensation, a feedforward Lagrange interpolation based algorithm is proposed. Real-time hardware results have been presented to demonstrate the effectiveness of the proposed algorithms and architecture for systems requiring high data throughput. It is shown that both the proposed algorithms achieve better performance as compared to existing algorithms.

**Keywords:** CDMA, Sampling Clock Offset, Estimation, FPGA, SDR, Wideband waveform.

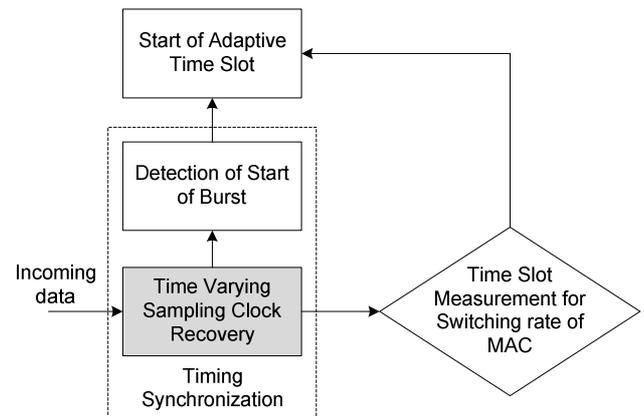
## 1. Introduction

Present wireless communication transceivers based on Software Defined Radio (SDR) technology have been used primarily in high-end applications due to their relative high cost and power consumption [1]. Examples are wireless military communication equipment and base station equipment for cellular mobile communications. The SDR-based networks of the future will have to support a wide variety of data-intensive applications such as situational awareness, biometrics, streaming video, IP data while offering a high degree of mobility, security and survivability. Due to these requirements, the developments for the future networks are moving toward wideband and digital signal based networking [2]. Wideband networking radio waveform is developed to overcome the insufficient capacities of the conventional narrowband wireless channel, so that it provides higher data transmission rate to support multimedia and bulky data traffic.

Direct Sequence Spread Spectrum has been used in the physical layer of wideband waveform. The purpose of Direct Sequence Spread Spectrum scheme is to make these networking radios work under noise floor without affecting primary license users in the used spectrum. Wireless networks suffer from fading, low power transmission, interference and interception. Direct Sequence Spread Spectrum is well suited for these networks due to its anti-jamming, anti-interference and robustness against multipath fading effects [3]. Multiple access in SDR-based wideband networks is provided by Time Division Multiple Access

(TDMA), Adaptive TDMA, Carrier Sense Multiple Access (CSMA) etc.

Although Direct Sequence Spread Spectrum has the advantages of security, low interference and Power Spectral Density (PSD) [4], timing synchronization is one of the major concerns of such systems. The problem of timing synchronization includes; (1) Estimation and Compensation of time varying Sampling Clock Offset (SCO) caused due to sampling clock inaccuracies and (2) Detection of the Start of burst for burst mode of transmission. Both these operations are very important in wideband networking radio operation as they directly affect the adaptive time slot algorithm in Adaptive TDMA based Medium Access Control (MAC) protocol. Figure 1 shows the flow graph of adaptive time slot algorithm in wideband networking operation and its relation with the timing synchronization.



**Figure 1.** Adaptive time slot algorithm and its relation with the timing synchronization (shaded block is the scope of this paper)

The paper is organized as follows. Section 2 presents some research work related to the estimation of sampling clock offset. Section 3 presents the system overview. The problem is formulated in section 4. The proposed sampling clock recovery algorithm is presented in section 5. Simulation results and comparison of the proposed algorithm with the existing techniques is given in section 6 followed by hardware architecture in section 7. Finally, section 8 concludes the paper.

## 2. Related Work

The first stage of timing synchronization is the recovery of time varying sampling clock offset which is the scope of this paper. The Sampling Clock Offset (SCO) is present due to

the inherent inaccuracies of transmit and receive crystal oscillators. Due to thermal drift, this sampling clock frequency offset will also change slowly in time [5]. A time domain-based sampling clock offset estimation and correction algorithm is presented in [6]. This algorithm is more specific to Orthogonal Frequency Division Multiplexing (OFDM) systems. A more generic low complexity algorithm for sampling clock offset estimation is proposed in [7]. Another algorithm for fractional timing estimation using two samples per symbol is proposed in [8]. Both the algorithms proposed in [7] and [8] have two major limitations. Firstly, they are only valid for constant drift between the transmitter and receiver sampling clocks and secondly, they do not incorporate the multipath channel effects while evaluating the estimator's performance. An algorithm based on Sample Point Reordering (SPR) is proposed in [9] but it assumes sampling clock inaccuracies of up to only 12 ppm and the performance is affected if the amount of frequency offset present in the received signal is large. The algorithms given in [6]-[9] provide SCO estimate only and do not explain the method of proper SCO compensation for high data rate systems.

To combat multipath fading effects, especially in fast fading channels, many communication systems use burst mode of transmission [10]. The size of each burst is selected such that the channel behaves time invariant within the duration of each burst [11]. Consequently, timing and frequency synchronization, channel estimation, equalization etc. are performed on each burst independently. In such systems, sampling clock offset needs to be tracked for each burst independently. Moreover, in a multiuser system, sampling clock offset mostly changes because of frequent switching of transmitting or receiving user.

To overcome the limitations of the existing algorithms mentioned above, a complete and robust sampling clock recovery algorithm for burst mode of transmission in wideband networking radios has been proposed. The proposed Sampling clock recovery algorithm consists of three stages. In the first stage, Sampling Clock Offset (SCO) is estimated at chip level. In the second stage, the SCO estimates are post-filtered to improve the tracking performance. We present a new post-filtering method namely Steady-State State-Space Recursive Least Squares with Adaptive Memory (S4RLSWAM). For the third stage of SCO compensation, a feedforward Lagrange interpolation based algorithm is proposed. The proposed algorithm is actually part of complete wideband waveform design that has been designed, implemented and tested on SDR platform.

### 3. System Overview

In wideband SDR waveforms, multiuser support is usually provided by Time Division Multiple Access (TDMA) and direct sequence spreading is used for security purpose only.

Since different applications (e.g. Push To Talk (PTT), position tracking, point-to-point calls, messages, file transfer, video communication etc.) have different Quality of Service (QoS) requirements. This varying QoS requirement is guaranteed by Adaptive TDMA (ATDMA). It provides certain end-to-end delay and reliability guarantees to different applications according to their QoS requirement. However, there is also certain delay constraint for non-real-time applications too, though not tightly bound. The real-time requirement is met in ATDMA based MAC protocol by guaranteeing the allocation of slots within the delay bound, while reliability is ensured by allocating conflict free time slots. The sampling clock recovery algorithm is very important in wideband networking radio operation as it directly affects the adaptive time slot algorithm in Adaptive TDMA based Medium Access Control (MAC) protocol (see Figure 1).

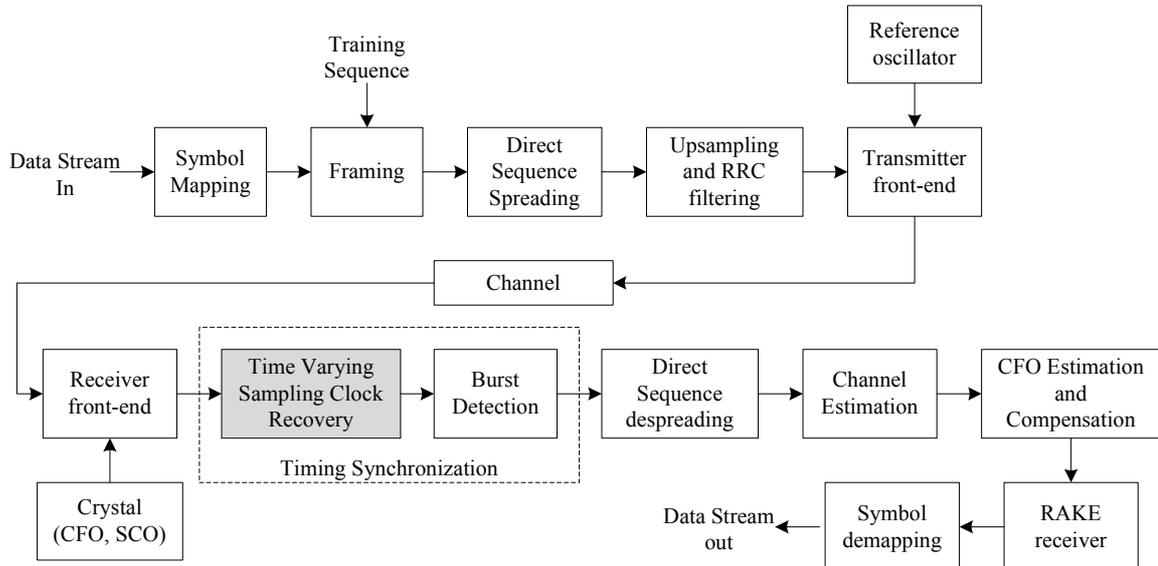
The physical layer of the wideband networking waveform based communication system is shown in Figure 2. At the transmitter side, the data stream is first mapped using QPSK symbol mapping. Bursts of the symbols are formed in which specific training sequence (to be discussed later) is inserted prior to each data burst. After direct sequence spreading, upsampling and Root Raised Cosine (RRC) filtering, the data is modulated with the carrier generated from the reference oscillator. After passing through channel, the data is received at the receiver's front-end. The crystal oscillator of the receiving device generates Carrier Frequency Offset (CFO) and Sampling Clock Offset (SCO). The next block is the scope of this paper which is SCO estimation & compensation. This stage is very important in case of Adaptive Time Division Multiple Access (ATDMA) since it controls the time slot measurement switching rate for Medium Access Control (see Figure 1). Next block is the detection of start of burst. After the detection of each valid burst, despreading operation is performed. The next blocks including channel estimation, CFO estimation & compensation and RAKE receiver are not the scope of this paper. However, efficient algorithms for these operations have also been proposed, implemented and tested by the authors. At the end, symbol demapping is performed to retrieve the data.

### 4. Problem Formulation

The transmitted baseband wideband direct sequence signal samples prior to upsampling and pulse shaping can be expressed as

$$s(t) = \sum_{m=0}^{M-1} g(m)\pi(t - mT_s) \quad (1)$$

where  $M$  is the number of symbols used for synchronization,  $g(m)$  and  $\pi(t)$  are the  $m^{\text{th}}$  symbol and spreading waveform of



**Figure 2.** Physical Layer block diagram of Wideband networking waveform (shaded block is the scope of this paper)

time span  $T_s$ , respectively. Symbol duration is denoted by  $T_s$  such that  $T_s = GT_c$ , where  $G$  is the spreading gain and  $T_c$  is the duration of one chip of spreading waveform. The data from each user is upsampled and filtered through Root Raised Cosine (RRC) filter [12] with impulse response given as

$$g_T(t) = 4R \frac{\cos((1+R)\pi t/T_c) + \frac{\sin((1-R)\pi t/T_c)}{4Rt/T_c}}{\pi\sqrt{T_c}(1-(4Rt/T_c)^2)}. \quad (2)$$

where  $R$  is the roll-off factor,  $T$  is the duration of each sample. Let the upsampled and filtered signal be  $x(t)$ . After passing through multipath fading channel, the signal for each burst of data is given as [13]

$$r(t) = \sum_{l=1}^J \alpha_l(t)x(t - \tau_l - \varepsilon(t))e^{j\Omega(t-\tau_l)} + n(t) \quad (3)$$

where  $J$  is the number of multipath,  $\alpha_l(t)$  is the fading coefficient of  $l^{\text{th}}$  path,  $\tau_l$  is the path delay of  $l^{\text{th}}$  path,  $\Omega$  is the carrier frequency offset which is present due to local oscillator frequency mismatch and/or Doppler spread,  $n(t)$  is the White Gaussian Noise with zero mean and variance  $\sigma^2$  and  $\varepsilon(t)$  is an unknown slowly varying time delay produced due to Sampling Clock Offset (SCO). This slowly varying time delay is produced due to frequency drift between the oscillators of the two communicating devices.

At the receiver, the analog received signal is first sampled by Analog-to-Digital Converter (ADC). The drift caused by the sampling clocks of the radios produces sampling clock errors at the ADC before timing and frequency estimation. Due to the sampling clock errors, ADC starts to sample at an unknown uncertain rate [9]. This uncertain rate is neither synchronous to the chip rate nor its oversampled rate. During the transmission of one burst, this clock error is accumulated. This causes excess and starvation of data samples at the output of ADC for slower and faster receiver sampling clocks, respectively. The situation is depicted in Figure 3.

## 5. Proposed Sampling Clock Recovery Algorithm

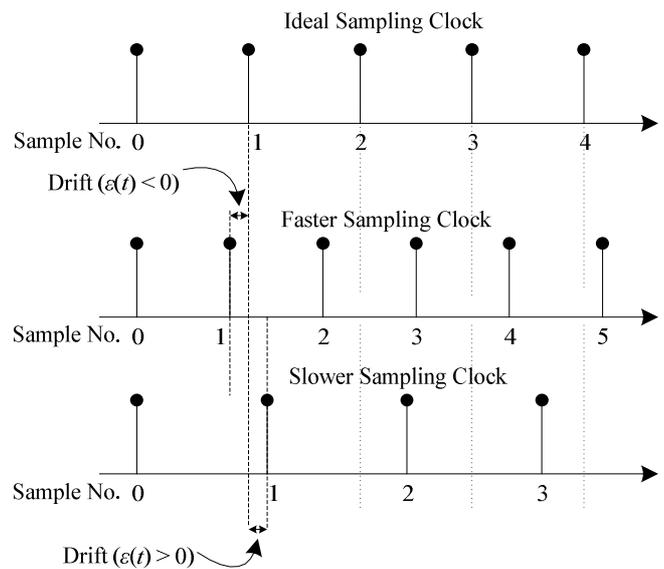
In this section, the proposed algorithm for sampling clock recovery has been presented. A three stage clock recovery algorithm has been proposed, including:

- Estimation of slowly varying time delay
- Post filtering of the time delay estimates
- Clock offset compensation

### 5.1. Stage 1

The first stage of the algorithm finds the estimate of the slowly varying time delay  $\varepsilon(t)$  defined in (3). The received signal is first sampled at a sample rate of  $N/T_c$ , where  $N$  is the upsampling factor and  $T_c$  is the chip duration. The sampled signal is given as

$$r_k = r(kT/N). \quad (4)$$



**Figure 3.** Concept of sampling clock drift

After processing through receiving matched filter (i.e. RRC filter) having impulse response of  $g_{R,k}$ , the filtered signal is given by

$$x_k = r_k * g_{R,k} \quad (5)$$

where  $*$  denotes the convolution sum. Now, the unbiased estimate of sampling clock offset  $\hat{\varepsilon}$  is found by  $KN$  samples of the filtered sequence using [14]

$$\hat{\varepsilon} = -\frac{N}{2\pi} \arg \left( \sum_{k=0}^{KN-1} |x_k|^2 e^{-j2\pi k/N} \right). \quad (6)$$

In (6),  $KN$  is the number of samples used for estimation. The authors of [14] estimate  $\varepsilon$  section by section by assuming very slow variation in time. For each section  $\Delta_m$  (where  $\varepsilon$  is assumed to be constant), an estimate  $\hat{\varepsilon}_m$  is found. This assumption is not practical in the presence of large clock offset. In our proposed estimator, the estimate  $\hat{\varepsilon}_m$  for each incoming chip sample is found by computing the complex Fourier coefficient of the  $KN$  chips samples where  $K$  is a design constant to be discussed later. The proposed sliding window computation of the estimates is given by

$$\hat{\varepsilon}_m = -\frac{N}{2\pi} \arg \left( \sum_{k=m}^{KN+m-1} |x_k|^2 e^{-j2\pi k/N} \right). \quad (7)$$

The sampling rate must be such that the spectral component of the filtered data at  $1/T_c$  can be represented. It means that we must have  $N/T_c > 2/T_c$ . So,  $N = 4$  has been chosen.

Now, for  $N = 4$ , substitute  $m = 4n$  in (7), so that it can further be written as

$$\hat{\varepsilon}_m = -\frac{2}{\pi} \tan^{-1} \left( \frac{\sum_{n=m}^{K+m-1} |x_{4n+3}|^2 - |x_{4n+1}|^2}{\sum_{n=m}^{K+m-1} |x_{4n}|^2 - |x_{4n+2}|^2} \right). \quad (8)$$

## 5.2. Stage 2

The second stage of SCO estimation gives the optimal estimate  $\tilde{\varepsilon}_m$  by post-filtering the estimate  $\hat{\varepsilon}_m$ . The main advantage of post-filtering is the reduction of variance of the estimates. In this paper, a new adaptive filter namely State Space Recursive Least Squares with Adaptive Memory (SSRLSWAM) has been introduced for post-filtering. SSRLSWAM has very good tracking performance especially in time-varying environments [15]. The reason for selecting SSRLSWAM instead of other adaptive filters (e.g. Least Mean Square filter, Kalman filter etc.) is the adaptive tuning of the forgetting factor, which is a key parameter in the SSRLSWAM algorithm. Since SSRLSWAM is computationally extensive, an approximate solution is used which is termed as Steady State SSRLSWAM (or S4RLSWAM). The steady state algorithm is still time varying due to the time varying behavior of the forgetting factor.

Since the SCO estimate from the first stage is bounded by  $\square N/2 \leq \hat{\varepsilon}_m < N/2$ , the post-filtered estimate  $\tilde{\varepsilon}_m$  must also be bounded. Therefore, S4RLSWAM cannot be directly applied for the post-filtering of SCO estimates. A new idea of boundedness has been proposed within S4RLSWAM algorithm. The proposed idea is to apply modulo  $\square N$

operation to the prediction error and a-posteriori states to reduce them to the interval  $[\square N/2, N/2)$ .

Now, the summarized S4RLSWAM algorithm with the proposed modulo  $\square N$  operation ( $[\cdot]_{\square N}$ ) is described which has been used to find the optimal SCO estimate  $\tilde{\varepsilon}_m$  (detailed generic algorithm of S4RLSWAM can be found in [15]). Since it is a recursive algorithm, it needs to be initialized. The method of regularization term has been used for initialization. Following initializations can be taken to simplify the process.

$$\psi[0] = \mathbf{0}, \hat{x}[0] = \mathbf{0}$$

The constant velocity model [16] has been used as the state space model of the signal. This model (unforced) is given as

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, C = [1 \quad 0]. \quad (9)$$

The algorithm then proceeds as follows.

- Predicted (a-priori) states

$$\bar{x}_m = A\hat{x}_{m-1}$$

- Predicted output (Predicted SCO Estimate)

$$\bar{\varepsilon}_m = C\bar{x}_m$$

- Prediction error ( $\hat{\varepsilon}_m$  is the input to filter)

$$\xi_m = [\hat{\varepsilon}_m - \bar{\varepsilon}_m]_{\square N}$$

- Forgetting factor update

$$\lambda_m = \left[ \lambda_{m-1} + \alpha \psi_{m-1} A^T C^T \xi_m \right]_{\lambda^-}^{\lambda^+}$$

- Calculation of matrices  $P_m$  and  $S_m$

$$P_m = P_\lambda = \begin{bmatrix} 1 - \lambda_m^2 & (1 - \lambda_m)^2 \\ (1 - \lambda_m)^2 & (1 - \lambda_m)^3 / \lambda_m \end{bmatrix}$$

$$S_m = S_\lambda = \frac{\partial P_\lambda}{\partial \lambda} = \begin{bmatrix} -2\lambda_m & -2(1 - \lambda_m) \\ -2(1 - \lambda_m) & (1 - \lambda_m)^2 (-1 - 2\lambda_m) / \lambda_m^2 \end{bmatrix}$$

- Calculation of S4RLSWAM gain

$$K_m = \lambda_{m-1}^{-1} A P_m A^T C^T \times \left[ 1 + \lambda_{m-1}^{-1} C A P_m A^T C^T \right]^{-1}$$

- States (a-posteriori) estimates

$$\hat{x}_m = [A\hat{x}_{m-1} + K_m \xi_m]_{\square N}$$

- Output estimate (Optimal SCO estimate  $\tilde{\varepsilon}_m$ )

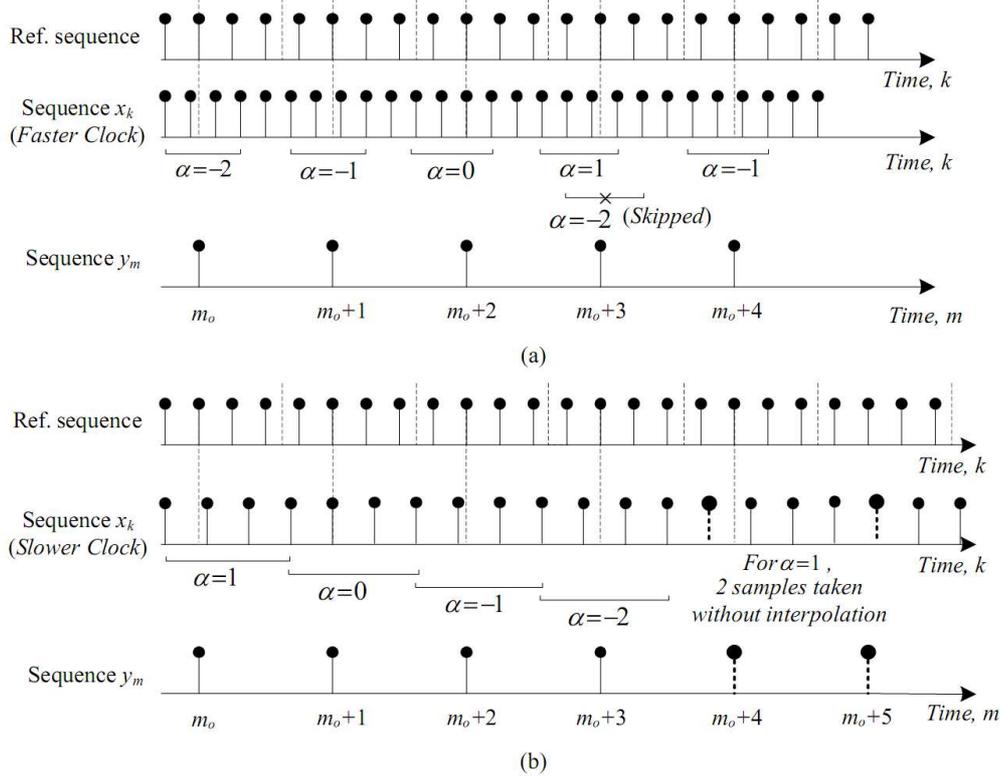
$$\tilde{\varepsilon}_m = C\hat{x}_m$$

- Update  $\psi_m$

$$\psi_m = (A - K_m C A) \psi_{m-1} + S_m C^T \xi_m$$

## 5.3. Stage 3

The third stage of sampling clock recovery consists of compensation of the sampling clock offset using the optimal estimate found in the second stage. A feedforward compensation method based on polynomial-based Lagrange interpolation has been proposed for this stage. It was



**Figure 4.** Proposed concept of SCO compensation using cubic interpolation; (a) Faster sampling clock (b) Slower sampling clock (Rapid changes in the integer part  $\alpha$  are shown to explain all the cases; this variation is slow in practical systems)

mentioned in the previous section that there can be both slower and faster receiver sampling clocks in different devices resulting in positive and negative clock drifts respectively (Figure 3). The situation is further depicted in Figure 4, which shows the method of selection of samples to be interpolated based on the estimated sampling clock offset.

From (8) it can be seen that the possible range of  $\hat{\epsilon}_m$  is  $\square 2 \leq \hat{\epsilon}_m < 2$  (for  $N = 4$ ), which will be the same for  $\tilde{\epsilon}_m$ . Let the integer part and fractional parts of  $\tilde{\epsilon}_m$  be  $\alpha$  and  $\delta$  respectively. If  $i$  is an index incremented by  $N$  for each value of  $m$ , then for  $N = 4$ , the samples for cubic interpolation are given as

$$\begin{aligned} s_1 &= x_{i+\alpha-1} \\ s_2 &= x_{i+\alpha} \\ s_3 &= x_{i+\alpha+1} \\ s_4 &= x_{i+\alpha+2} \end{aligned}$$

After selecting the samples for interpolation, the fractional part  $\delta$  is used to perform the Lagrange polynomial based cubic interpolation. Let  $s_1, s_2, s_3,$  and  $s_4$  be the 4 samples of the filtered signal corresponding to a specific value of  $m$  on which cubic interpolation is to be performed. For each estimate  $\tilde{\epsilon}_m$ , the fractional part  $\delta_m$  is used to interpolate the 4 samples. The compensated and downsampled sample  $y_m$  is given as

$$\begin{aligned} y_m &= \left( -\frac{s_1}{6} + \frac{s_2}{2} - \frac{s_3}{2} + \frac{s_4}{6} \right) \delta_m^3 \\ &+ \left( \frac{s_1}{2} - s_2 + \frac{s_3}{2} \right) \delta_m^2 \\ &+ \left( -\frac{s_1}{3} - \frac{s_2}{2} + s_3 - \frac{s_4}{6} \right) \delta_m + s_2 \end{aligned} \quad (10)$$

Another problem caused by the sampling clock drift is the excess and starvation of samples at the receiver due to faster and slower receiver sampling clocks respectively. In case of faster receiver clock, samples must be discarded whereas in case of slower receiver clock, extra samples must be put to avoid starvation of data samples.

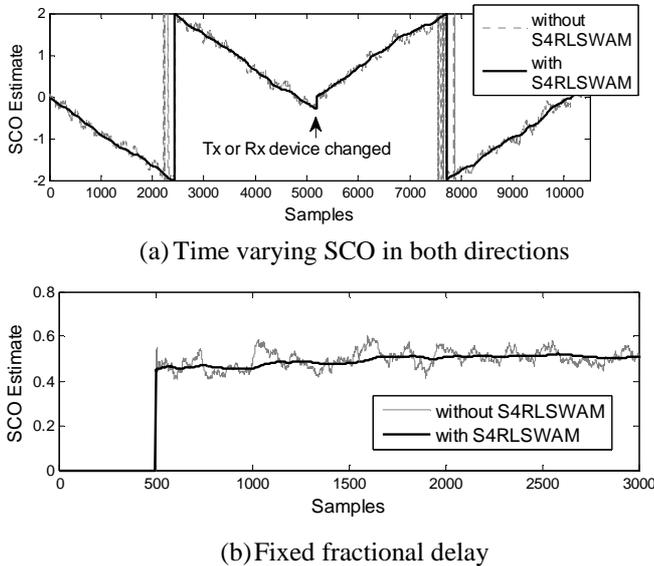
A new technique to avoid starvation or excess of samples is proposed. The proposed technique works as follows (assuming  $N = 4$ ). The integer part  $\alpha$  of the SCO estimate decides whether the downsampler has to put required samples or discard extra samples. This is explained in Figure 4, which shows that if the receiver has faster sampling clock, the integer part  $\alpha$  increases from  $\square 2$  to 1 slowly. Since SCO estimate has an upper bound of  $\tilde{\epsilon}_m < 2$  integer part jumps from 1 to  $\square 2$ . At this point a sample is discarded for the purpose of synchronization. Similarly, if the receiver has slower sampling clock, the integer part  $\alpha$  decreases from 1 to  $\square 2$  slowly. Since the SCO estimate is lower bounded by  $\tilde{\epsilon}_m \geq \square 2$ , integer part jumps from  $\square 2$  to 1. At this point, the valid sample ( $x_{i+\alpha\square 3}$ ) is inserted directly without interpolation and an extra sample ( $x_{i+\alpha+1}$ ) is also inserted directly to avoid starvation of samples.

## 6. Simulation Results

In this section, simulation results of the proposed sampling clock recovery algorithm have been presented. The parameters used in the simulation are as follows.

- Number of modulated symbols in each ( $N_{spr}$ ) = 288
- RRC roll-off factor ( $R$ ) = 0.65
- Number of samples used for SCO estimation ( $K$ ) = 32
- Initial value of forgetting factor ( $\lambda_{ini}$ ) = 0.995
- Constant  $\alpha = 0.000005$

Golay sequences of length 16 have been used for spreading and QPSK has been used as the modulation technique. Figure 5 shows the tracking performance of the proposed estimator with and without S4RLSWAM post-filtering at SNR of 2 dB. S4RLSWAM has been initialized by the method of regularization term. The performance metric used in Figure 5 is the post-filtered SCO estimate  $\tilde{\varepsilon}_m$ . In Figure 5(a), tracking performance is shown for a slowly varying time delay due to clock offsets of  $\pm 200$  ppm and  $\pm 200$  ppm. The sudden change of clock offset is common in communication networks due to the change of transmitting or receiving device. It can be seen that the proposed estimator efficiently tracks the varying time delay even after the sudden change in clock offset. The proposed estimator can cope with large sampling clock offsets in contrast to the algorithm given in [11] which assumes the inaccuracy of oscillators up to only 12 ppm. Figure 5(b) shows the tracking performance of the proposed estimator for a fixed fractional time delay of  $0.5T$ . It can be seen that the proposed estimator performs well for both the fixed and varying time delays.



**Figure 5.** Performance of the proposed SCO estimator for time varying and fixed clock offsets

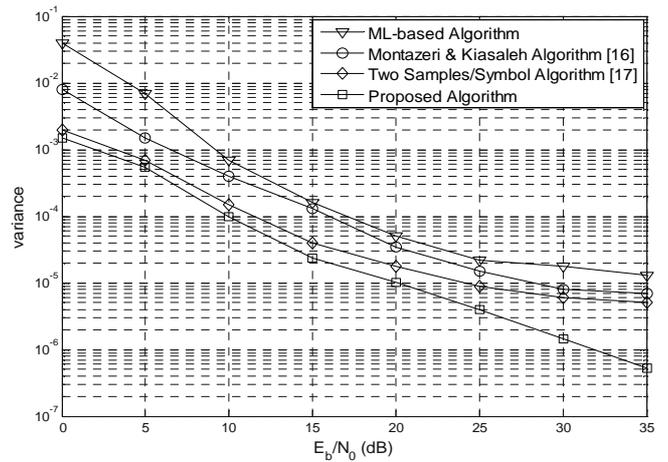
The comparison of variances of the proposed and other well-known SCO estimators is shown in Figure 6. The performance metric (i.e. estimator's variance) is given as

$$\text{var}(\tilde{\varepsilon}) = \frac{1}{M} \sum_{i=1}^M (\tilde{\varepsilon}_i - E(\tilde{\varepsilon}))^2 \quad (11)$$

where

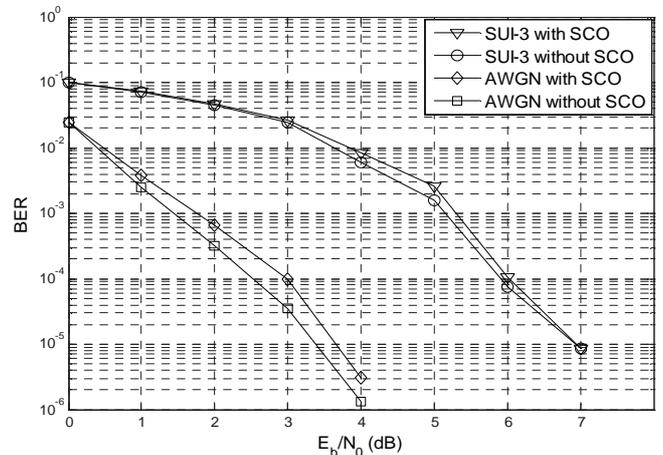
$$E(\tilde{\varepsilon}) = \frac{1}{M} \sum_{i=1}^M \tilde{\varepsilon}_i \quad (12)$$

is the sample mean of the post-filtered estimate  $\tilde{\varepsilon}_m$  and  $M$  is the number of realizations. In this simulation, we have taken  $M = 2000$ . For comparison, ML-based algorithm [17], Montazeri & Kiasaleh's estimator [7] and a two samples/symbol based feedforward algorithm given in [8] have been considered. It can be seen that the proposed estimator shows considerable performance improvement when compared to other estimators at all SNRs. It is also worth mentioning here that the algorithms given in [7] and [8] are applicable only when the time delay is fixed i.e.  $\varepsilon(t) = \varepsilon$ , whereas the proposed estimator is capable of estimating both the fixed and time varying delays efficiently.



**Figure 6.** Performance comparison of the proposed estimator with other well-known estimators

Now, we present the Bit Error Rate (BER) performance of the overall all system using the proposed timing synchronization approach. In Figure 7, the Bit Error Rate (BER) performance in AWGN and Stanford University Interim-3 (SUI-3) channel model [18] with and without sampling clock errors is shown. It can be seen that the BER performance of the system with proposed timing synchronization is robust against the sampling clock errors even at low SNRs and in the presence of multipath fading effects.



**Figure 7.** BER performance of the system with proposed sampling clock recovery

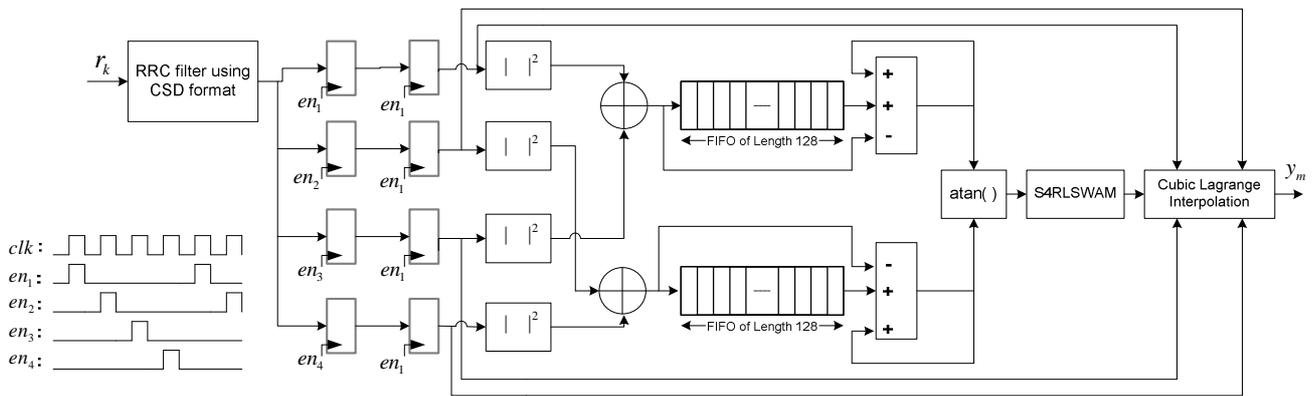


Figure 8. Proposed FPGA implementation of the proposed sampling clock recovery algorithm

## 7. Efficient FPGA Implementation

In this section, the hardware architectures of the proposed sampling clock recovery and burst detection algorithms for FPGA implementation on SDR platform have been presented. The FPGA device used for the implementation is XC3SD3400A which belongs to the Spartan-3A DSP family of FPGAs. This family of FPGA offers density of 3.4million system gates. The Spartan-3A DSP family builds on the success of the Spartan-3A FPGA family by adding XtremeDSP™ DSP48A slices. New features improve system performance and reduce the cost of configuration. These Spartan-3A DSP FPGA enhancements, combined with proven 90 nm process technology, deliver more functionality and bandwidth per dollar than ever before, setting the new standard in the programmable logic and DSP processing industry.

Figure 8 shows the hardware architecture for sampling clock recovery algorithm. First of all, multiplier-less RRC filter of order 16 has been implemented using Canonic Signed Digit (CSD) representation of filter coefficients [19]. After filtering, a high data rate parallel processing based realization of (7) and (8) for the estimation of each incoming sample has been presented. This high data rate parallel processing is specific to the case when  $N = 4$ . A similar approach can be followed for higher values of  $N$ , but it will certainly increase the complexity of the system. The four enable signals used in the realization are also shown in Figure 8. After squaring and summation, the sliding window computation of (8) has been implemented by the use of two First-Input First-Output (FIFOs). The inphase and quadrature components are then fed to  $\text{atan}()$  block which is implemented using the CORDIC core of Xilinx core generator. The resulting SCO estimate then goes to S4RLSWAM block for post-filtering. The block-level architecture of implemented S4RLSWAM has been shown in Figure 9. Finally, the proposed cubic Lagrange interpolation is implemented which takes the post-filtered SCO estimate and the corresponding four samples to perform the cubic interpolation for SCO compensation. The device utilization summary of the FPGA implementation is shown in Table 1.

Tracking performance of the proposed SCO estimator (for negative offset) captured from ChipScope after implementation on FPGA is shown in Figure 10. The input data is taken at an SNR = 2. It is obvious that the estimate

has very low variance. The transition occurs from 32766 to 32762 (i.e. from 1.9999 to 1.996 for Q2.14 format), which is almost equal to that of simulated result. This is further elaborated in Figure 11 which shows the comparison of SCO estimate obtained from MATLAB simulation and FPGA hardware. It can be seen that both the estimates are identical since they are overlapping each other. The bottom sub-figure shows the magnified SCO estimate of the last few samples from the first figure to have a clear visualization.

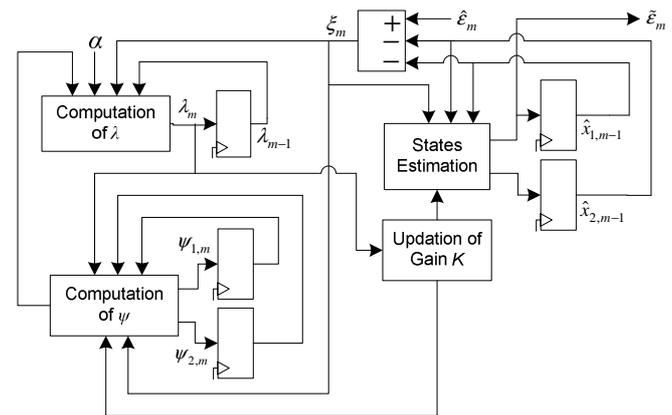
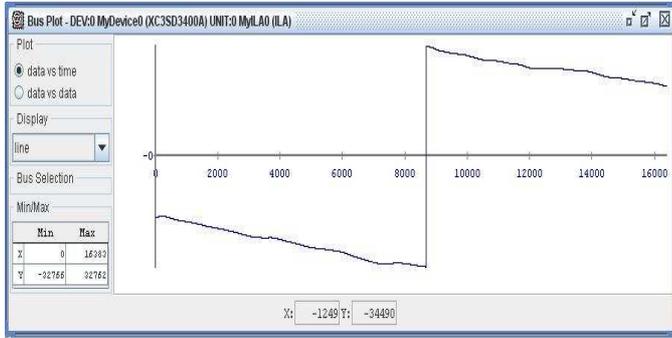
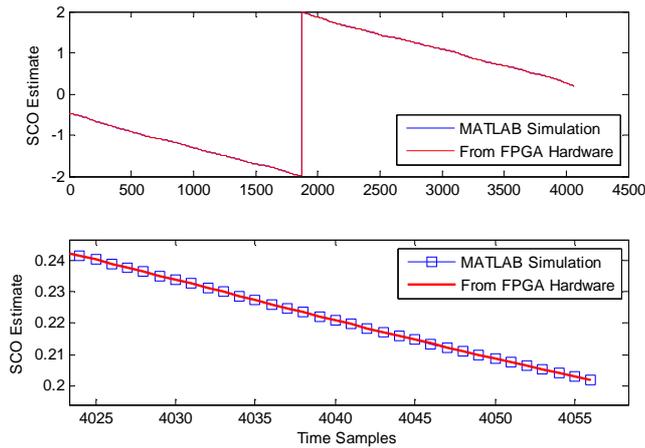


Figure 9. Block level architecture for S4RLSWAM implementation on FPGA

The proposed algorithm is actually part of a complete wideband waveform physical layer design that has been implemented, tested and verified. The proposed sampling clock recovery algorithm has been implemented on Field Programmable Gate Array (FPGA) and the remaining portions of the receiver like Burst detection, Carrier Frequency Offset (CFO) estimation, channel estimation, RAKE reception and QPSK demodulation has been implemented on FPGA and in software on a Digital Signal Processor (DSP) by effective design partitioning. The resultant device area and timing constraints and code execution time constraints are very easily met by the FPGA and DSP, respectively. It is possible to achieve higher data rates by selecting a greater chip rate, though certain modifications to the receiver design may be required.



**Figure 10.** Performance of implemented SCO estimator on FPGA (captured from ChipScope)



**Figure 11.** Comparison of MATLAB simulation and FPGA hardware results of the proposed SCO estimator

**Table 1.** Device Utilization Summary

Resources	Used	Total	Percentage
Number of Slices	4076	23872	17%
Number of Slice Flip Flops	5002	47744	9%
Number of 4 input LUTs	5585	47744	11%
Number of BRAMs	14	126	11%
Number of DSP48s	47	126	37%

## 8. Conclusion and Future Work

In this paper, practically efficient algorithm for sampling clock recovery for burst mode wideband networking waveform of software defined radio has been proposed. Sampling clock recovery plays a key role in the adaptive time slot measurement for switching rate of medium access control. The proposed sampling clock recovery algorithm includes proposed modified square timing estimation, S4RLSWAM based post-filtering and proposed cubic interpolation based compensation. The proposed algorithm shows considerable performance improvement when compared to other well-known algorithms. Our simulation has also shown that the proposed estimator is capable of estimating both the fixed and time varying delays. Practical FPGA architectures and implementation results for the proposed algorithm on FPGA platform have also been

presented. It has been shown that the hardware results are identical to the simulation results.

Some other post-filtering approaches can be applied to the proposed SCO estimation as future research work. Moreover, the FPGA implementation can be further optimized to achieve lesser resource utilization.

## References

- [1] J. Rohde, and T. S. Toftgaard, "Adaptive cognitive radio technology for low power wireless personal area network devices", *Wireless Personal Communications*, Vol. 58, No. 1, pp. 111-123, 2011.
- [2] S. Han, J-H. Park, H-H. Shin, and B-S. Kim, "Performance enhancements in TDMA-based tactical wireless networks", in *Proceedings of Vehicular Technology Conference*, Vol. 1, pp. 1-5, 2012.
- [3] M. Nakagawa, "Consumer communications based on spread spectrum technologies", in *Proceedings of IEEE 3rd International Symposium on Spread Spectrum Techniques and Applications*, Vol. 1, pp. 138-145, 1994.
- [4] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley, 1995.
- [5] E. Grass, K. Tittelbach-Helmrich, U. Jagdhold, A. Troya, G. Lippert, et al., "On the Single-Chip Implementation of a Hiperlan/2 and IEEE 802.11a Capable Modem", *IEEE Personal Communications*, Vol. 8, No. 6, pp. 48-57, 2001.
- [6] B. Ai, Y. Shen, Z. D. Zhong, and B. H. Zhang, "Enhanced sampling clock offset correction based on time domain estimation scheme", *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 2, pp. 696-704, 2011.
- [7] A. Montazeri, K. Kiasaleh, "Design and performance analysis of a low complexity digital clock recovery algorithm for software defined radio applications", *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 3, pp. 1258-1263, 2010.
- [8] W-P. Zhu, Y. Yan, M. O. Ahmed, and M. N. S. Swamy, "Feedforward symbol timing recovery technique using two samples per symbol", *IEEE Transactions on Circuits and Systems-I*, Vol. 52, No. 11, pp. 2490-2500, 2005.
- [9] C-F. Li, Y-S. Chu, J-S. Ho, and W-H. Sheen, "Cell search in WCDMA under large-frequency and clock errors: Algorithm to hardware implementation", *IEEE Transactions on Circuits and Systems-I*, Vol. 55, No. 2, pp. 659-671, 2008.
- [10] U. Mengali, M. Morelli, "Data-aided frequency estimation for burst digital transmission", *IEEE Transactions on Communications*, Vol. 45, No. 1, pp. 23-25, 1997.
- [11] A. Goldsmith, *Wireless Communications*, Cambridge University Press, UK, 2005.
- [12] B. Sklar, *Digital Communications; Fundamentals and Applications*, 2nd Ed., Prentice Hall, 2002.
- [13] K. Fazel, and S. Kaiser, *Multicarrier and Spread Spectrum Systems*, John Wiley and Sons, 2003.
- [14] M. Oerder, and H. Meyr, "Digital filter and square timing recovery", *IEEE Transactions on Communications*, Vol. 56, No. 5, pp. 605-612, 1988.

- [15] M. B. Malik, "State-space recursive least squares with adaptive memory", *Signal Processing Journal*, Vol. 86, pp. 1365-1374, 2006.
- [16] Y. Bar-Shalom, and X-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York, Wiley, 2001.
- [17] U. Mengali, and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*, New York, Plenum Press, 1997.
- [18] IEEE 802.16 Broadband Wireless Access Working Group, Channel models for fixed wireless applications, <http://www.ieee802.org/16/tg3/contrib>. [Accessed on 10 Dec. 2009].
- [19] S. A. Khan, *Digital Design for Signal Processing Systems: A Practical Approach*, 1st Ed., John Wiley and Sons, 2011.