

# A Parallelized Phase-Frequency Detector based Modified LSPF-DPLL for Wireless Communication – Comparative Study with Modified LSPF-DPLL

Sabyasachi Bhattacharyya<sup>1,a</sup>, Aradhana Misra<sup>1</sup>, Kandarpa Kumar Sarma<sup>2</sup> and Nikos Mastorakis<sup>3</sup>

<sup>1</sup>Gauhati University Institute of Science and Technology, Department of Electronics and Communication Engineering, Guwahati, India

<sup>2</sup>Gauhati University, Department of Electronics and Communication Technology, Guwahati, India

<sup>3</sup>Technical University of Sofia, Bulgaria

**Abstract.** In recent times, communication technologies have evolved at a brisk rate. 3G and 4G networks have already been widely deployed. However, with increase in user number and applications, alternative strategies in the form of 5G systems have been devised. Thus, it is desirable to have communication systems which can provide low error levels with minimum delay. DPLL based systems play a versatile role in modern day communication receivers. Popular DPLL receiver packages based on Zero Crossing (ZC) and Least Square Polynomial Fitting (LSPF) techniques have been recently proposed to serve standalone reception in communication setups. Such systems involve intensive computations and thus show excellent error performance ; but are bottlenecked by poor time performance. The LSPF-DPLL system was aided by a Modified Phase Resolving Numerically Controlled Oscillator (MPR-NCO) to achieve both improved error and time performance in another subsequent DPLL design that was proposed of late. This design is modified in the DPLL system proposed here by incorporating a Parallelized LSPF based Phase-Frequency Detector to achieve further improvements in time performance while maintaining the system's error performance.

## 1 Introduction

Digital Phase Locked Loops (DPLLs) are the sampled domain extraction of the Phase Locked Loops (PLLs) realized by means of a digital Phase-Frequency Detector (PFD) [1,2]. Popularly referred to as an FM demodulator, DPLL plays diverse roles in communication systems which include recovering carrier information in non-coherent systems where the channel state is not known to the receiver [2], symbol retrieval in coherent links for known channel [3], multi-clock synchronization and clock distribution into different sub-systems using divider unit in PLLs besides time synchronizing such systems [2]. DPLL based packages have therefore been emphasized as standalone means of reception in recent wireless communication setups [3].

A ZC-DPLL has been proposed in [4] for high Doppler environments which incorporates a hyperbolic non-linearity block and a sigma-delta modulator unit which helps the loop filter to adapt better. The proposed system exhibits improved time jitter performance and widened lock-in range. The ZC-DPLL proposed for Rayleigh faded channels in [3] utilizes the conclusions of ZC Algorithm in its PFD and retrieves the original phase of transmitted QPSK symbols. The system exhibits excellent error rates but the intensive Left/Right (L/R) Shift algorithm leads to poor time performance. Similarly, a highly intense L/R Shift algorithm based

Numerical Controlled Oscillator (NCO) and LSPF based PFD aided DPLL based signal recovery system is presented in [2]. This system also shows excellent error performance but lacks time efficiency. An MPR-NCO based LSPF-DPLL system is proposed in [1] for wireless fading environments and emerges as an improved design over the existing LSPF-DPLL [2] with better error and time performance.

The recent trend in wireless communications as can be seen in [1,4] is time efficient symbol recovery. Certain DPLL designs however emphasize on significant reduction in BER levels and therefore involve intensive computations which lead to poor time performance [2, 3]. The LSPF-DPLL in [2] additionally also involves another hugely time-intensive LSPF based PFD unit. An MPR-NCO based LSPF-DPLL has been recently proposed in [1] and exhibits improved error and time performance over the existing design in [2]. The time efficiency improvement in [1] is solely due to the incorporation of the MPR-NCO. The design presented here tries to further improve system time performance by parallelizing the operations of intensive LSPF based PFD in [1] using the different cores of a Quad Core processor while maintaining the system's error performance.

The following sections constitute the remaining portion of this paper. Section 2 discusses the proposed system model for the Parallelized Phase Frequency Detector based Modified LSPF-DPLL system in terms of

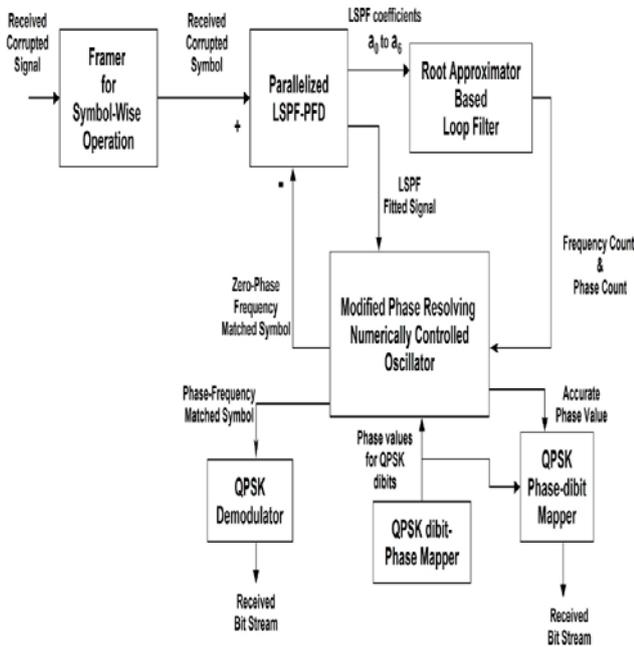
<sup>a</sup> Sabyasachi Bhattacharyya: sabya005@gmail

its crucial components and their functioning, different mathematical relationships involved, parameters settings for simulation and the modifications made to parallelize the DPLL in [1] are also discussed in details. Section 3 analyzes the performance of the proposed system and a comparative study of the system with the MPR-NCO based LSPF-DPLL system in [1] is also presented here. Section 4 deals with the advantages and limitations of the system proposed and finally Section 5 concludes the discussion.

## 2 System Model

The proposed DPLL system with parallelized PFD is subjected to degraded QPSK modulated symbols under the combined effects of Rayleigh and Rician faded wireless environments and Additive White Gaussian Noise (AWGN).

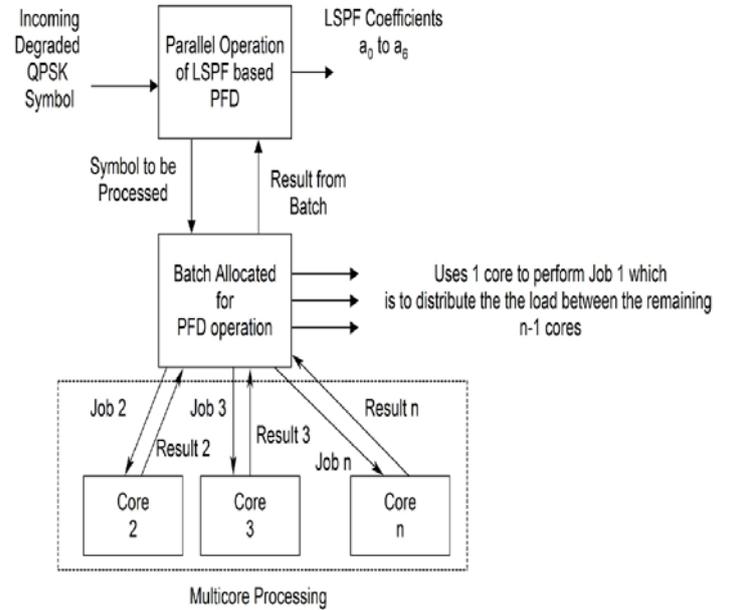
### 2.1 System Architecture



**Figure 1.** Parallelized PFD based Modified LSPF-DPLL

The architecture overview for the proposed DPLL system is depicted in Figure 1. It consists of a framer for symbol-wise operation, a Parallelized version of the LSPF based Phase-Frequency Detector (LSPF-PFD) in [1], a Root Approximator (RA) based Loop Filter (RA-LF) and an MPR-NCO which are integrated for proper system functioning. QPSK dibit-phase and phase-dibit mapper units aid the proposed DPLL topology. The Framer unit splits the received signal into frames of length same as QPSK symbol period. The time-intensive LSPF-PFD proposed in [1,2] is modified here by parallelizing the PFD action in [1] using the different cores of a Quad core processor as shown in Figure 2. The time efficient parallelized PFD provides the best fit coefficients  $a_0$  to  $a_6$  containing ZC information of the

incoming signal. The RA-LF utilizes these coefficients to measure the frequency and phase counts associated with the incoming degraded symbols. It does so by extracting the roots and Eigen values of the best-fit estimate. The MPR-NCO uses the frequency and phase counts from RA-LF and fitted signal from the LSPF-PFD and provides three crucial outputs.



**Figure 2.** Proposed model for Parallelized PFD in LSPF-DPLL

They are the zero-phase frequency matched signal which is fed back to the LSPF-PFD, the phase-frequency matched signal with original modulation information and also the accurate phase value associated with the received symbol computed from the phase count of RA-LF. The NCO determines the accurate phase value using the look-up information of dibit-phase mapper. The accurate phase is further mapped to the correct dibits using the phase-dibit mapper to reduce the need for complicated branched QPSK demodulation as suggested in [2, 3, 5].

### 2.2 Parallel PFD based LSPF-DPLL Analysis

The modified LSPF-DPLL with Parallelized PFD proposed for QPSK modulated symbols recovers digital intelligence under degraded effects of Rayleigh and Rician fading channels and AWGN noise.

#### 2.2.1 QPSK signal model

Quadri-phase signals significantly conserve bandwidth by transmitting dibits and also maintain low error rates using distinct decision boundaries as elaborated in [2,6,7]. The proposed system is thus inspected for QPSK modulated symbols. A QPSK modulated signal is mathematically represented as in eq. (1)

$$q(t) = A \cos(2\pi f_c t + \theta_i) \quad (1)$$

$$0 \leq t \leq T, i = 1,2,3,4$$

$$\theta_i = \frac{(2i-1)\pi}{4}$$

where,  $A$  is the signal amplitude,  $f_c$  is the transmission carrier frequency and  $\theta_i$  is the QPSK phase. The four possible phases are :  $\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}$ .

### 2.2.2 Rayleigh and Rician Fading Model

Rayleigh and Rician fading models provide accurate estimates of the wireless fading environment prevalent in urban, sub-urban and rural areas [5,7]. The proposed Parallel PFD based modified DPLL system is tested under degraded effects of Rayleigh and Rician fading.

The received signal  $s(t)$  under Rayleigh faded NLOS transmission scenario between transmitter and receiver is depicted in eq. (2):

$$s(t) = \sum_{k=1}^N a_k \cos(\omega_c t + \omega_{dk} t + \phi_k) \quad (2)$$

where  $a_k$  is the gain in the  $k^{\text{th}}$  path,  $\omega_c$  is the angular carrier frequency for transmission,  $\omega_{dk}$  is the Doppler Frequency due to relative motion in between the transmitter and receiver for  $k^{\text{th}}$  path and  $\phi_k$  is the random phase for the  $k^{\text{th}}$  path [2,5].

Similarly, the received signal under Rician faded wireless environment composite of NLOS and LOS paths is depicted as shown in eq. (3) :

$$r(t) = s(t) + k_d \cos(\omega_c t + \omega_d t) \quad (3)$$

where  $s(t)$  represents the contribution due to the NLOS components,  $k_d$  indicates the strength of the LOS component, and  $\omega_d$  is the Doppler frequency component in the direct path[1, 2, 5].

### 2.2.3 Component wise analysis of Proposed Model

The proposed Parallel PFD based Modified LSPF-DPLL depicted in Figure 1 incorporates *five* functional blocks as already mentioned in Section 2.1. The Framer Unit provides symbol-wise DPLL operation. The detailed functioning of the four major units are elaborated below.

#### Parallelized LSPF based PFD

The LSPF based PFD performs phase-frequency detection at moderate sampling rates over ZC based designs as suggested in [1, 2]. The presented  $6^{\text{th}}$  order polynomial fitting PFD [2] however involves intensive computations and thus leads to poor time performance of the PFD. The design presented in [1] incorporates an MPR-NCO which leads to improvements in processing time. The phase-frequency detection process in [1] was realized using the intensive LSPF unit and so in addition to the time efficiency achieved due to the MPR-NCO, there remained scope for further improvement in time performance with necessary modifications made in the

PFD unit in [1]. To achieve this improvement in system performance, the intensive computations involved in the serial LSPF-PFD unit of [1] are parallelized using the different cores of a Quad Core Processor. The parallelized version of the LSPF-PFD provides the best-fit coefficients  $a_0$  to  $a_6$  which are indicative measures of phase and frequency content associated with the incoming symbol.

If  $(t_1, y_1), (t_2, y_2), \dots, (t_n, y_n)$  represent the corrupted signal samples of received QPSK symbols along with time index, the received symbol can therefore be fitted to a  $6^{\text{th}}$  order polynomial, such that the sum of squared residuals  $S$  is minimized as discussed in [2]. The fitted signal is mathematically expressed as in eq. (4):

$$\hat{y} = a_0 + a_1 t + a_2 t^2 + \dots + a_6 t^6 \quad (4)$$

$$S = \sum_{i=1}^N [y_i - \hat{y}]^2 \quad (5)$$

On partially differentiating  $S$  in eq. (5) with respect to best-fit coefficients  $a_0$  to  $a_6$ , and equating each of them to zero, an important matrix equality is obtained in eq. (6) from which the coefficients  $a_0$  to  $a_6$  can be obtained.

$$\begin{bmatrix} nS & \sum t_i & \sum t_i^2 & \sum t_i^3 & \sum t_i^4 & \sum t_i^5 & \sum t_i^6 \\ \sum t_i & \sum t_i^2 & \sum t_i^3 & \sum t_i^4 & \sum t_i^5 & \sum t_i^6 & \sum t_i^7 \\ \sum t_i^2 & \sum t_i^3 & \sum t_i^4 & \sum t_i^5 & \sum t_i^6 & \sum t_i^7 & \sum t_i^8 \\ \sum t_i^3 & \sum t_i^4 & \sum t_i^5 & \sum t_i^6 & \sum t_i^7 & \sum t_i^8 & \sum t_i^9 \\ \sum t_i^4 & \sum t_i^5 & \sum t_i^6 & \sum t_i^7 & \sum t_i^8 & \sum t_i^9 & \sum t_i^{10} \\ \sum t_i^5 & \sum t_i^6 & \sum t_i^7 & \sum t_i^8 & \sum t_i^9 & \sum t_i^{10} & \sum t_i^{11} \\ \sum t_i^6 & \sum t_i^7 & \sum t_i^8 & \sum t_i^9 & \sum t_i^{10} & \sum t_i^{11} & \sum t_i^{12} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum t_i y_i \\ \sum t_i y_i^2 \\ \sum t_i y_i^3 \\ \sum t_i y_i^4 \\ \sum t_i y_i^5 \\ \sum t_i y_i^6 \end{bmatrix} \quad (6)$$

Eq. (6) provides the optimum LSPF coefficients  $a_0$  to  $a_6$ , which contains phase and frequency contents of the received QPSK symbol. Moreover, it computes the LSPF best fit curve which was additionally incorporated in [1] to obtain information about whether the signal phase being tracked is below or beyond  $\pi$  radians. As can be seen, a number of intensive computations are involved in eq. (6) to compute the best fit LSPF coefficients. Sequential execution of such intense computations using a single core of the processor leads to hugely degraded time performance. So, these intense computations of the PFD are distributed for parallel execution on multiple processor cores as depicted in Figure 2. A Quad core processor is used to achieve speed up in the PFD and thus overall time performance improvement over [1]. The proposed system incorporates *batch mode of processing* to initiate a batch consisting of  $n$  cores. Each PFD computation is termed as a *job*. The batch uses *one* core to execute its own workload distribution and result accumulation process whereas the remaining  $n-1$  cores are used to parallelly execute  $n-1$  jobs at a time as shown in Figure 2. Upon completion of all the necessary jobs

(computations), the results of all the jobs are combined together to provide the PFD output, i.e., the LSPF coefficients  $a_0$  to  $a_6$  (fed to RA-LF) and the fitted signal  $\hat{y}$  (fed to MPR-NCO). It is important to note that the computations in eq. (6) are to be performed per symbol of the received QPSK signal. The parallelization and subsequent speed up are therefore achieved for each received symbol. This speed up achieved per symbol of received data, will definitely lead to an overall time performance improvement for a standard transmission with multiple QPSK symbols [2]. The computation time is directly related to the number of signal samples  $ns$  used to represent each incoming symbol and thus the variation of processing time with  $ns$  has been analyzed.

### RA based LF

The Loop Filter Unit for the proposed DPLL is realized using a Root Approximation unit as proposed in [1, 2]. It utilizes a matrix  $M_{RA}$  formed by the coefficients  $a_0$  to  $a_6$  from the PFD unit to compute the roots signifying signal zero crossing index as shown in eq. (7) below.

$$M_{RA} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 & -a_5 & -a_6 \\ a_0 & a_0 & a_0 & a_0 & a_0 & a_0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

From the matrix in eq. (7), six eigen values  $e_1$  to  $e_6$  are computed and further using eq. (8) six roots  $r_1$  to  $r_6$  are evaluated as below.

$$r_i = \frac{1}{e_i} \quad (8)$$

As elaborated in [1,2], the frequency and phase counts,  $f_{cnt}$  and  $p_{cnt}$ , are computed from the significant roots  $r_2$  to  $r_5$  using eq. (9) and (10).

$$f_{cnt} = \frac{1}{r_3 - r_1} \quad (9)$$

$$p_{cnt} = \frac{\pi}{r_3 - r_2} \times r_2 \quad (10)$$

The MPR-NCO uses the computed values of  $f_{cnt}$  and  $p_{cnt}$  for further processing as shown in Figure 1.

### MPR-NCO

The Parallelized DPLL system incorporates the Modified NCO proposed in [1] to compute accurate phase values associated with incoming QPSK symbols by

avoiding sample-wise reorientation [8, 9, 10] resulting in improved time as well as error performance [1]. The MPR-NCO uses  $f_{cnt}$  and  $p_{cnt}$  from RA-LF,  $\hat{y}$  from Parallelized PFD unit and possible phase values from look up table of QPSK dibit-phase mapper unit. It performs the following steps for proper phase and frequency matching as suggested in [1].

- It uses  $f_{cnt}$  to regenerate the carrier zero-phase carrier which acts as free running NCO signal.
- It studies the first zero crossing of  $\hat{y}$  to set a flag  $f$  as either '0' or '1' as elaborated in [1].
- The estimated phase  $est_p$  is thus computed using eq. (11) in a similar manner to [1].

$$est_p = p_{cnt} + (f \times \pi) \quad (11)$$

- The accurate phase  $acc_p$  is finally computed as the phase value from the look-up table having minimum difference with  $est_p$ . Using  $acc_p$  and  $f_{cnt}$ , the modified NCO generates the phase frequency matched signal.  $acc_p$  is fed to dibit-phase mapper unit for simpler bit recovery [1].

### QPSK dibit-phase and phase-dibit mapper

A similar dibit-phase and phase-dibit mapper unit as proposed in [1] is incorporated here for simpler bit recovery by avoiding branched demodulation structure suggested in [2,11,12]. The dibit-phase mapper aids NCO action by providing different possible phases whereas the phase-dibit mapper uses the look-up table of dibits and phases shown in Table 1 to map  $acc_p$  to the correct dibits facilitating simpler recovery of digital data.

Table 1. QPSK dibit-phase look-up table

| Dibit | Phase Associated |
|-------|------------------|
| 00    | $\frac{\pi}{4}$  |
| 01    | $\frac{3\pi}{4}$ |
| 10    | $\frac{5\pi}{4}$ |
| 11    | $\frac{7\pi}{4}$ |

### 2.2.3 Simulation Parameter Settings

The Modified LSPF-DPLL system proposed with Parallelized PFD is tested under *non-coherent transmission scenario*. Table 2 shows the different parameters and corresponding specifications under which the proposed system has been simulated.

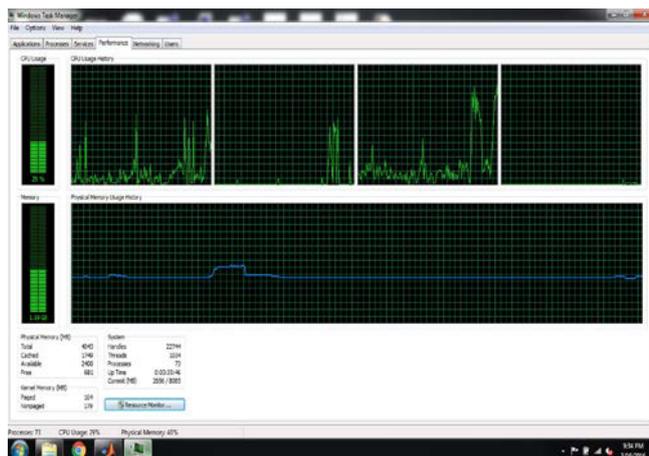
**Table 2.** Simulation Parameters

| Parameters                              | Values or specification                               |
|---|---|
| Modulation Scheme                       | QPSK  |
| Carrier Frequency                       | 900 MHz   |
| Fading                                  | Rayleigh and Rician (5 and 10 path)                   |
| AWGN Noise                              | -10 to 10 dB (SNR)                                    |
| Number of bits in simulation            | 20000   |
| LOS factor                              | 0.5   |
| LSPF order                              | 6   |
| Processor Used                          | Intel Core i5, Quad Core, 4 GB (RAM), 64-bit, 2.4 GHz |
| Number of cores used to parallelize     | three and four  |
| Sample size variation (per QPSK symbol) | 20,00000 to 2,60,00000 (in steps of 20,00000)         |
| Parallelization Method Used             | Batch Processing                                      |

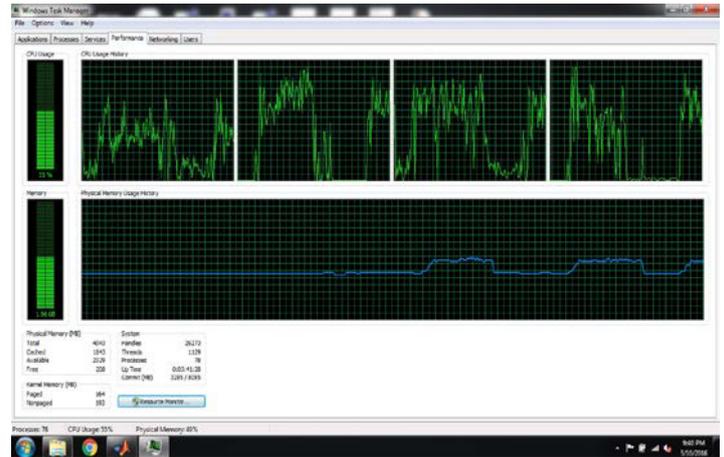
As depicted in Table 2, the number of cores used for parallel execution of the load is varied between *three and four*. The mode of processing is batch type, so *one* core is always used to execute the batch process. Therefore, when *three* cores are used, effectively the PFD load is distributed between *two* cores. Similarly, the load is divided between *three* cores when the total number of cores used is *four*.

### 3 Results and Discussion

The analysis presented in this section can be broadly classified into *two* parts. Firstly, the system performance improvement due to parallelization of PFD unit is studied in terms of CPU utilization, processing time requirement and percentage speed up achieved with respect to the serial DPLL [1]. Secondly, the error performance of the system is analyzed in terms of a comparative BER analysis under different fading conditions to ensure that BER levels of Serial DPLL [1] are preserved.

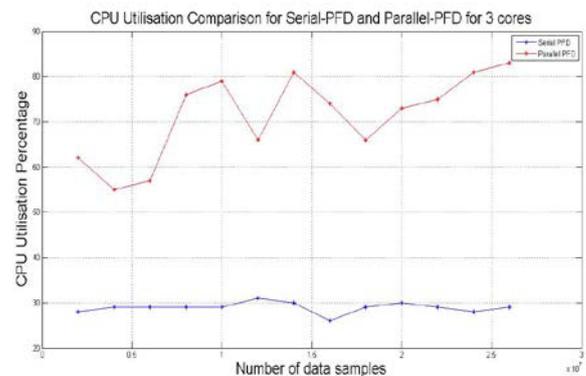


**Figure 3.** CPU processing for Serial DPLL (40,00000 samples)

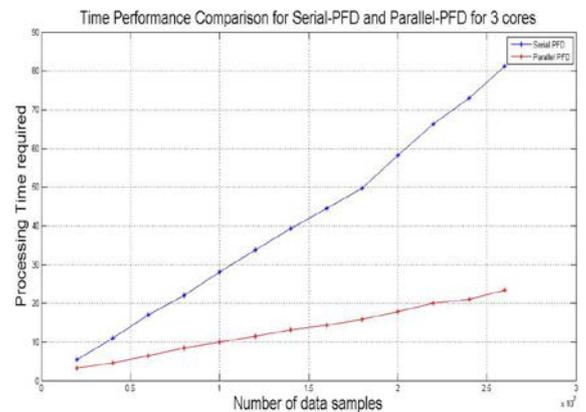


**Figure 4.** CPU processing in Parallel DPLL (40,00000 samples)

Figures 3 and 4 depict the CPU processing carried out for Serial and Parallel DPLL respectively. The sample size in each case is 40,00000. It can be very clearly seen that for Serial DPLL in Figure 3, only one processor is being used and the CPU cores are not utilized properly. On the other hand, for Parallel DPLL in Figure 4, all the processor cores are utilized properly resulting in better CPU utilization.

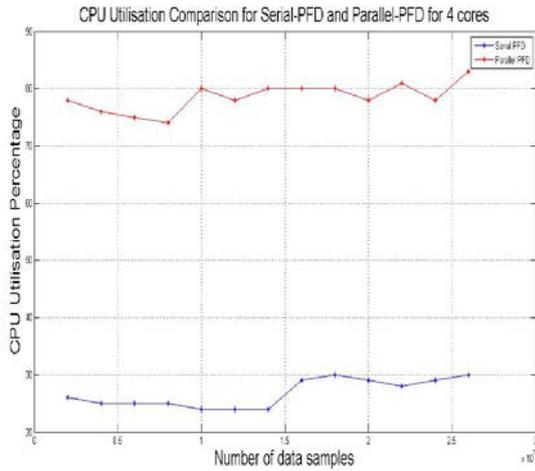


**Figure 5.** CPU utilization for Serial and Parallel DPLL (3 core)

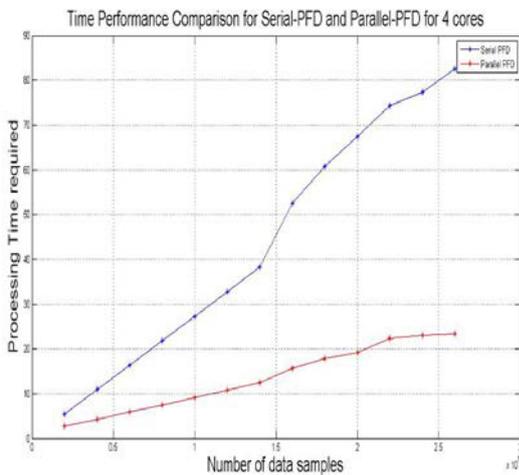


**Figure 6.** Processing time for Serial and Parallel DPLL (3 core)

Figures 5 and 6 exhibit average improvement in CPU utilization by 43 % and speed-up as high as 71 % for Parallel DPLL over Serial DPLL under 3-core parallelization.



**Figure 7.** CPU utilization for Serial and Parallel DPLL (4 core)

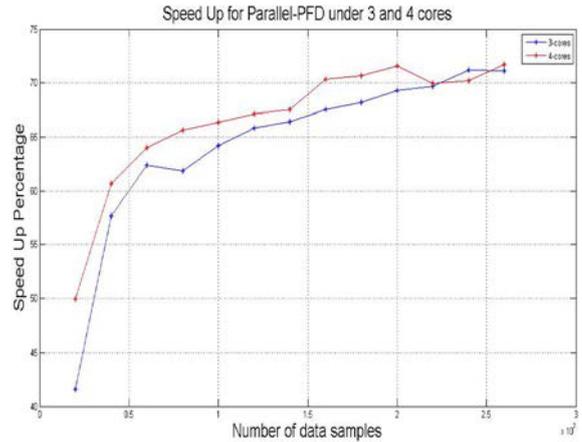


**Figure 8.** Processing time for Serial and Parallel DPLL (4 core)

Figures 7, 8 and Table 3 show average improvement in CPU utilization for Parallel DPLL over Serial DPLL by 53 % and speed up as high as 72 % with 4 cores.

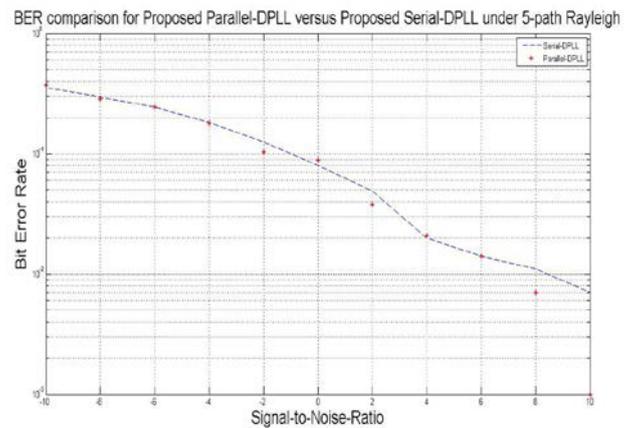
**Table 3.** CPU utilization and Time Performance Comparison

| Sample Size | CPU Utilization (%) |                         | 4-core Processing Time (seconds) |                         |
|-------------|---------------------|-------------------------|----------------------------------|-------------------------|
|             | Serial DPLL         | Parallel DPLL (4 cores) | Serial DPLL                      | Parallel DPLL (4 cores) |
| 2000000     | 26                  | 78                      | 5.449                            | 2.729                   |
| 6000000     | 25                  | 75                      | 16.353                           | 5.882                   |
| 8000000     | 25                  | 74                      | 21.846                           | 7.501                   |
| 12000000    | 24                  | 78                      | 32.834                           | 10.787                  |
| 26000000    | 30                  | 83                      | 82.499                           | 23.328                  |

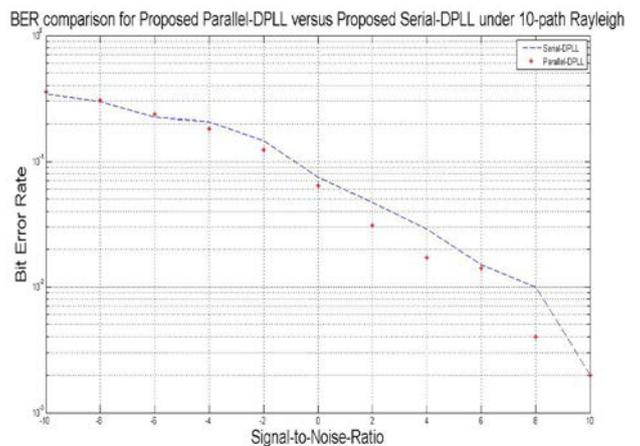


**Figure 9.** Speed up for Parallel DPLL under 3 and 4 cores

Figure 9 depicts a comparative analysis of the speed up achieved for the proposed DPLL with Parallel PFD under 3 and 4 cores. It is quite clear that speed up achieved is more with 4 cores than that with 3 cores because with increase in number of cores, the speed up increases.

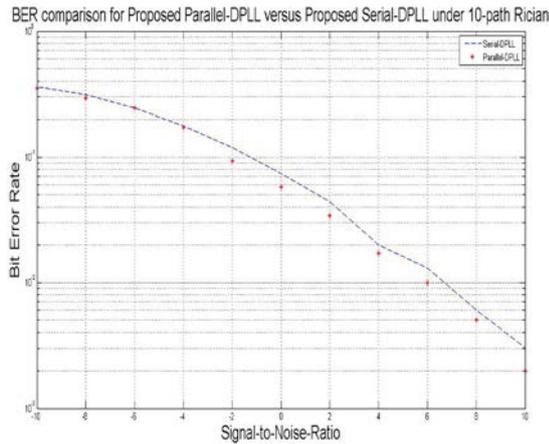


**Figure 10.** BER comparison under 5-path Rayleigh (3 core)



**Figure 11.** BER comparison under 10-path Rayleigh (3 core)

Figures 10, 11 and 12 clearly indicate that the proposed system with 3 cores maintains the BER levels exhibited by the Serial DPLL [1] under 5-path Rayleigh, 10-path Rayleigh and 10-path Rician respectively in addition to providing improvements in time performance and CPU utilization.



**Figure 12.** BER comparison under 10-path Rician (3 core)

Table 4 depicts the comparative BER analysis of the proposed Parallel DPLL with Serial DPLL [1] under 4-core parallelization. As in case of 3 core, the proposed system maintains BER levels similar to those of Serial DPLL proposed in [1] for 4 core case also.

**Table 4.** BER Comparison under 10-path Rayleigh and Rician

| SNR (in dB) | BER under 10-path Rayleigh |                         | BER under 10-path Rician |                         |
|-------------|----------------------------|-------------------------|--------------------------|-------------------------|
|             | Serial DPLL                | Parallel DPLL (4 cores) | Serial DPLL              | Parallel DPLL (4 cores) |
| -10         | 0.3430                     | 0.3350                  | 0.3600                   | 0.3480                  |
| -4          | 0.2070                     | 0.1790                  | 0.1780                   | 0.1790                  |
| 0           | 0.0750                     | 0.0650                  | 0.0740                   | 0.0570                  |
| 2           | 0.0470                     | 0.0430                  | 0.0440                   | 0.0330                  |
| 4           | 0.0290                     | 0.0220                  | 0.0200                   | 0.0120                  |
| 6           | 0.0150                     | 0.0100                  | 0.0130                   | 0.0070                  |
| 10          | 0.0020                     | 0.0020                  | 0.0030                   | 0.0010                  |

Thus, the different results and illustrations presented in this section are substantial evidences for us to claim the Modified LSPF-DPLL proposed with Parallelized PFD as an improved design over the Serial LSPF-DPLL with MPR-NCO[1]. The proposed system exhibits performance improvement in terms of processing time, percentage speed up and CPU utilization while maintaining BER levels similar to the Serial DPLL [1].

## 4 Advantages and Limitation

The proposed system has certain advantages and also some limitations which are outlined below.

### 4.1 Advantages

In addition to possessing the advantages suggested in [1], the system also exhibits the following features:

- Reduced computational load per core due to distribution of PFD operations into different cores of the Quad core processor.
- Further improvement in processing speed as compared to the Serial DPLLs presented in [1] [2].
- Improved CPU resource utilization over systems proposed in [1,2]; serves as a dedicated receiver package for DPLL based communications.

### 4.2 Limitation

In some specific cases where either the sample size per QPSK symbol or the total number of such symbols transmitted are comparatively smaller than the data sizes for a standard transmission, the parallelization overheads may consume more time leading to ineffective parallelization and thus degraded time performance.

## 5 Conclusion and Future Direction

This paper proposes a Parallelized PFD based LSPF-DPLL system. The previously proposed Serial DPLL with MPR-NCO [1] is modified here by incorporating a new parallel processing aided LSPF-PFD unit to time efficiently track accurate phase values for retrieving digital intelligence. The system is tested under varying conditions of Rayleigh and Rician fading. Specifically, the performance of PFD unit is rigorously tested under different number of cores and with varying sample sizes. The proposed DPLL system is found to exhibit improvements in time performance, percentage speed up and CPU utilization over the Serial DPLL in [1] making it a suitable for today's technology focussed on minimizing delays. In addition to this, the system also maintains similar BER levels to those suggested in [1]. The proposed work could be extended further by analyzing the system under realistic estimates of Nakagami-*m* fading channels composite of Rayleigh and Rician behaviour.

## Acknowledgement

The authors are thankful to Mr. Basab Bijoy Purkayastha, Dept. of Physics, Indian Institute of Technology Guwahati, Mr. Kaustubh Bhattacharyya, Dept. of ECE, School of Technology, Assam Don Bosco University for their valuable suggestions and also to the Ministry of Communication and Information Technology, Govt. of India, for facilitating the research.

## References

1. S.Bhattacharyya, A. Misra, K. K. Sarma, "A Modified NCO based LSPF-DPLL Phase Resolver for Wireless Communication", *3<sup>rd</sup> IEEE International Conference on Recent Advances in Information Technology*, 1-7, Dhanbad (2016)
2. B.B. Purkayastha, K.K. Sarma, *A Digital Phase Locked Loop based Signal and Symbol Recovery System for Wireless Communication* (Springer India, 2015)
3. S.Bhattacharyya, R.N. Ahmed, B.B. Purkayastha, K. Bhattacharyya, "Zero Crossing Algorithm based Phase Recovery for DPLL based Wireless Communication", *IEEE International Conference on Computation, Communication and Automation*, 1220-1226, Greater Noida (2015)
4. E. Salahat, M.A. Qutayri, S.R. Al-Araji, H. Saleh, "A Zero Crossing Digital Phase Locked Loop Architecture with Hyperbolic Non-Linearity for High Doppler Environments", *21<sup>st</sup> IEEE International Conference on Electronics, Circuits and Systems*, 56-59, Marseille (2014).
5. B. B. Purkayastha, K. K. Sarma, "A Digital Phase Locked Loop based System for Nakagami-m Fading Channel Model", *Int. J Comp. App.*, **42(9)**: 1-8 (2012).
6. E. Salahat, S. R. Al-Araji, M. Al-Qutayri, "A Frequency Synthesizer based on Zero Crossing Digital Phase Locked Loop", *20<sup>th</sup> IEEE International Conference on Electronics, Circuits and Systems*, 835-838, Abu Dhabi (2013).
7. S. Kandeepan, "Steady State Distribution of a Hyperbolic Digital Tanlock Loop for Wireless Communications", *IEEE Trans. Wir. Comm.*, **8(2)** : 890-897 (2009).
8. Q. Nasir, S. Al-Araji, "Linearized Phase Detector Zero Crossing DPLL Performance Evaluation in Faded Mobile Channels", *Circ. and Sys.*, **2(3)** : 139-144 (2011).
9. S. R. Al-Araji, E. Salahat, D. Kilani, S. A. Yasin, H. Alkhoja, J. Aweya, "Adaptive Zero Crossing Digital Phase Locked Loop for Packet Synchronization", *11<sup>th</sup> IEEE International New Circuits and Systems Conference*, 215-221, Paris (2013).
10. S. Levantino, C. Samori, "Non-Linearity Cancellation in Digital PLLs", *IEEE Custom Integrated Circuits Conference*, 1-8, San Jose (2013).
11. E. Salahat, M. A. Qutyari, S. R. Al-Araji, "Robust Hyperbolic  $\Sigma$ - $\Delta$  Based No Delay Tan Lock Loop for Wireless Communication", *80<sup>th</sup> IEEE Vehicular Technology Conference*, 512-517, Vancouver (2014).
12. C. C. Hung, S.I. Liu, "A 40 GHz Fast Locked All Digital Phase Locked Loop using Modified Bang-Bang Algorithm", *IEEE Trans. Circ. And Sys.*, **58(6)** : 321-325 (2011).