Synchronization in IEEE 802.15.4 Zigbee Transceiver using Matlab Simulink

Kavya Gorantla
National Institute of Technology
Warangal-506004
Email: ggkavya19@gmail.com

V.V.Mani
National Institute of Technology
Warangal-506004
Email: vvmani@nitw.ac.in

***Abstract*—Zigbee standard has all set of protocols which supports low data rate, low power consumption, low cost and short range communications. The main application of Zigbee is Wireless Sensor Networks(WSNs). In digital communications, data is being transmitted by converting input bit stream in to sample functions of analog waveforms. When the analog RF signal passes through bandlimited channel, it results signal degradation in terms of symbol delay, carrier frequency and phase offsets. Various synchronization techniques can be used to estimate these characteristics in coherent receivers. In this paper, phase offset can be estimated using costas loop carrier recovery circuit and early late gate timing recovery algorithm is used to estimate symbol timing before data being decoded. The receiver performs well for different SNR (Signal to Noise Ratio) values varying from -10dB to 14dB. Transceiver system design is developed using Matlab Simulink and its performance can be analysed when it undergoes through Additive White Gaussian Noise (AWGN) channel and Rayleigh fading channel.**

***Index Terms*—Costas loop carrier recovery, Early late gate timing recovery, IEEE 802.15.4, LR-WPAN, Zigbee transceiver.**

I. Introduction

The word synchronization refers to the process of making two or more events occur at the same time and at the same frequency. In order to convey the information correctly a communication receiver must be synchronized with the corresponding transmitter [1]. For demodulation and detection at receiver, estimation of the carrier frequency and synchronization of symbol timing are necessary. Symbol timing is used for estimating the received symbol which is used at sampling. Carrier frequency and phase are used at all coherent demodulators.

Carrier synchronization refers to the process of generating sinusoidal signal that closely tracks the phase and frequency of a received noisy carrier. Symbol synchronization is the process of deriving the receiver timing signals that indicate where in time the transmitted symbols are located. Among all timing recovery algorithms, early late gate synchronizer is flexible to design [2]. Zigbee/IEEE 802.15.4 is the standard which supports low data rate, low power consumption, low cost and short range communication. It mainly operates at 2.4 GHz frequency range, designed using OQPSK (Orthogonal Quadrature Phase Shift Keying) modulation scheme along with half sine pulse shaping and DSSS (Direct Sequence Spread Spectrum) technique [3]. Hence, Zigbee transceiver is complex circuit and it also introduces some channel delay, phase offset. Because of the above problems the final decoded bit stream may not be equal to the original transmitted signal which degrades the system performance. In order to enhance the performance of the Zigbee transceiver those parameters need to be estimated using synchronization techniques.

In this paper Zigbee transceiver is designed as per the specifications given by IEEE 802.15.4 in Matlab Simulink [4]. Carrier synchronization is done with costas loop carrier recovery technique [5] and for symbol synchronization early late gate timing recovery algorithm is used [6]. Matlab Simulink has been chosen for simulation because of its flexibility, cost effective and efficient in design and implementation.

This paper is organised as follows. In section II the basic structure of Zigbee Receiver Design with carrier and symbol synchronization in Simulink are presented along with their Mathematical equations. Simulation results and conclusions are given in III and IV respectively.

II. Synchronization in Receiver

There are several factors contributing towards the time, frequency and phase offsets between transmitter and receiver such as physical distance between transmitter and receiver, the difference between the oscillators at transmitter and receiver etc. Receiver design itself is a very difficult task which involves carrier recovery and symbol timing recovery circuits [7]. Achieving synchronization is most challenging aspect in digital communications. The model in Fig.1. include carrier and symbol synchronization.

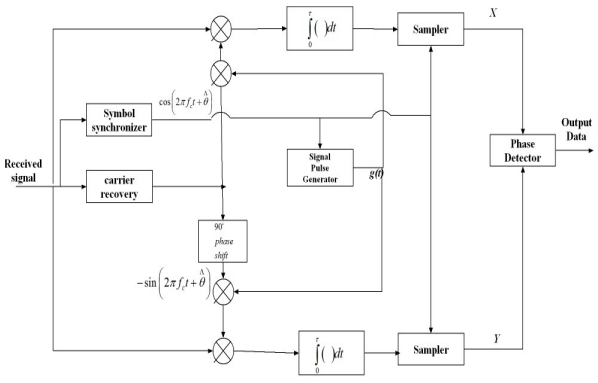

Fig. 1: Receiver Model

The two reference signals for two correlators are $g(t)\cos(2\pi f_c t + \hat{\theta})$ and $-g(t)\sin(2\pi f_c t + \hat{\theta})$ [8].

- The carrier phase estimate $\hat{\theta}$ from carrier recovery circuit is used in generating the reference signal for these correlators.
- The symbol synchronizer control the sampler and the output of the signal pulse generator.

The incoming received signal is applied to two synchronous correlators, consisting of a multiplier followed by a lowpass filter. Each multiplier is supplied with OQPSK envelope and one reference signal. This multiplier output contains the baseband data and high frequency harmonics. To avoid these harmonics, the multiplied signal is passed through a low pass filter. Then the resultant data is passed through a sampler which holds the data for one bit period.

The phase detector compares the received signal phase with the transmitted signal phase. The function of the phase detector is to generate an error signal, which is used to return the oscillator phase whenever its output deviates from original input signal. Costas loop method for carrier recovery and early late gate algorithm for timing recovery is explained in detail in further sections.

A. Carrier Phase Synchronization

In a digital data transmission system, the transmission chain includes several oscillators for modulation, demodulation, up- and down-conversions and sampling. A carrier phase error causes a rotation in the signal space projections and a constant frequency error causes time varying rotation in the signal projections. If the rotation is large enough, the signal space projections in each possible symbol lie in the wrong decision region. As a consequence, decision errors occurs even with perfect symbol timing synchronization and in the absence of additive noise.

In non-coherent systems, accurate frequency adjustment is enough but in coherent systems accurate phase recovery is needed. The phase locked loop (PLL) and costas loop are closed loop control systems and generates a feedback signal whose phase and frequency are aligned to the phase and frequency of the reference signal in locked condition [9]. In case of QPSK/OQPSK modulation the carrier cannot be simply tracked with phase locked loop at the receiver, but a more complex method of carrier recovery is needed. Costas loop is a good choice for the complex circuit like Zigbee.

1) Mathematical Model for Costas Loop Carrier Recovery: Consider the OQPSK costas loop shown in Fig.2.

The input signal is,

$$m(t)\cos(\theta(t)) - m(t)\sin(\theta(t)), \tag{1}$$

where m(t)=± 1 is the transmitted data, $\sin(\theta(T))$ and $\cos(\theta(T))$ are carriers. The output of Voltage Controlled Oscillator (VCO) is supposed to be sinusoidal $\cos(\hat{\theta}(t))$ [10].

After multiplying input signal (I) by the VCO signal, we get

$$i(t) = (m(t)\cos(\theta(t)) - m(t)\sin(\theta(t))\cos(\hat{\theta}(t)) \tag{2}$$

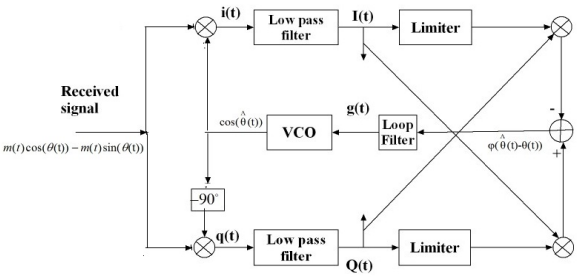

Fig. 2: The OQPSK Costas Loop

Similarly, multiplying input signal (Q) by the VCO signal shifted by -90^0, we get

$$q(t) = (m(t)\cos(\theta(t)) - m(t)\sin(\theta(t))\sin(\hat{\theta}(t)) \tag{3}$$

After filtering these signals by low pass filter,

$$I(t) = \int_0^t h(t-\tau)i(\tau)d\tau, \tag{4}$$

$$Q(t) = \int_0^t h(t-\tau)q(\tau)d\tau, \tag{5}$$

where, $h(t-\tau)$ is an impulse function of the Filter. Filter removes higher order frequency components and used to obtain one of the input signal carrier.

After filtration, both signals I(t) and Q(t) are passed through limiter circuit. Then the output of limiter are multiplied as shown in Fig.2. [11]. The phase detector (PD) is a non linear element with the output of $\varphi(\hat{\theta}(t)-\theta(t))$. The difference $\hat{\theta}(t)-\theta(t)$ is called as phase error $(\theta_e(t))$.

The phase error is then filtered by the loop filter to produce a control voltage $v(t)$ which is used to set the phase of the VCO. The signal after loop filter gives control signal g(t) for the VCO. where,

$$g(t) = \int_0^t \gamma(t-\tau)\varphi(\hat{\theta}(\tau) - \theta(\tau))d\tau. \tag{6}$$

The VCO output $y(t) = \cos(\hat{\theta}(t)$ is related to the input g(t) via the phase relationship,

$$\hat{\theta}(t) = k_0 \int_{-\infty}^t g(x)dx \tag{7}$$

Where k_0 is a proportionality constant, called the VCO gain.

After certain transient processes, the carrier and VCO phases becomes equal and control input to VCO becomes zero $i.e.$ g(t)=0. It means estimated phase equals actual phase and the error value then becomes zero.

2) Implementation of Costas Carrier Recovery in Simulink: The costas loop carrier recovery is designed using basic Simulink blocks as explained in previous section. The received signal at 2.4 GHz is first lead to the costas loop which consists of low pass filters, loop filter, VCO as shown in Fig.4.

The two low pass filters are realized with transfer function $F(s) = \frac{1}{1+0.02s}$, Loop filter is with transfer function $F(s) = \frac{1}{s+4}$. Low pass filter removes the high frequency harmonics. Loop filter filters the phase error to produce the control voltage g(t).

VCO subsystem consists of constant signal block VCOfreq=2.4 GHz which determines self frequency of VCO, Integrator block, summing block, gain=30, and two blocks which defines sinusoidal waveforms as shown in Fig.3.

The control signal g(t) is then fed to the VCO and helps to generate the signal with 2.4 GHz with variable phase. After some transient processes, control signal becomes zero and phase (estimated phase $\hat{\theta}(t)$) of VCO output will becomes equal to original carrier phase $\theta(t)$. The loop adjusts the control voltage g(t) to produce the phase estimate which drives the phase error to zero.

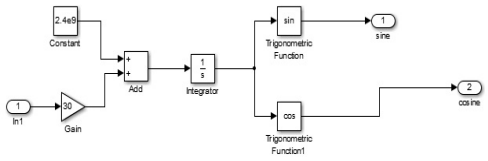

Fig. 3: Voltage controlled Oscillator Block

B. Symbol Synchronization

Symbol timing error introduces inter symbol interference and reduces noise margin. Among all timing recovery algorithms, early late gate synchronizer is flexible to design.

In this technique, recovering symbol timing is performed using a Delay locked loop (DLL). The goal of the DLL is to sample the output of the low pass filter at the peaks. In Zigbee transceiver model the signal gets sampled at 1 μ sec after low pass filter by using zero order hold circuit. But it doesn't sample exactly at peak of the signal. Hence, it results some delay in received signal. This Delay gets encountered by adding delay to PN sequence at the despreading [12].

Now DLL circuit will resolve this problem of delay by extracting sample time at peak of the signal $i.e.$ Early-late gate symbol time synchronization

1) Algorithm: When DLL begins running, it arbitrarily selects a sample, called the on-time sample (current sample) from the low pass filter. It captures the two adjacent samples (on either side of the sampling instant 'T') that are separated by δ seconds. The sample at the index T-δ is called Early sample and the sample at the index T+δ is called Late sample. Fig.5.a. shows the waveform when the on-time sample is at the peak and early sample and late sample difference is zero [13].

Fig.5.b. and Fig.5.c. show examples in which the on-time sample comes before a peak and after the peak. To achieve the best performance in the presence of noise, the timing of on-time samples must be adjusted using DLL to coincide with peaks in the waveform.

There are three cases:

1) In Fig.5.a, the on-time sample is already at the peak. The next on-time sample will be at another peak when it takes the sample after T_{symb} samples.
2) In Fig.5.b, the on-time sample is too late. To move closer to the next peak, the next on-time sample to be taken $T_{symb}+1$ samples after the current on-time sample.
3) In Fig.5.c, the on-time sample is too early. To move closer to the next peak, the next on-time sample to be taken $T_{symb}-1$ samples after the current on-time sample [14].

The offset decision block uses the on-time ($x(kT_s)$), early ($x(kT_s+\delta)$), and late samples ($x(kT_s-\delta)$) to determine whether sampling is at a peak, too early, or too late. It then sets the time at which the next on-time sample is taken.

The input to the offset decision block is $[x(kT_s)(x(kT_s+\delta)-x(kT_s-\delta))]$, called the decision statistics. The offset decision block could adjust the time at which the next on-time sample is taken based only on the decision statistics.

2) Implementation of Early-late gate algorithm in Simulink: In this model early sample signal is taken as incoming signal, on-time sample signal is the delayed version of incoming signal with one delay block, late sample signal is also the delayed version but with two delay blocks. Then these signals are fed to a sampler to get the above three samples which can control with the enable input starts initially at 0.435×10^{-6} sec. Enable input is given by programmable pulse wave generator with 1×10^{-6} sec of period , pulse width of 10 percent. Total no.of samples between two peaks are 16, then sample time is 0.0625×10^{-6}.

The Early-late gate timing recovery block in Simulink is shown in Fig.6.

Then the resultant three samples are lead to the process of Decision statistics $i.e.$ $[x(kT_s)(x(kT_s+\delta)-x(kT_s-\delta))]$ which is input of offset decision block.

- When the decision statistic is negative, the on-time sample is too early. Next sampling instant = current sample time$-\delta$
- When it is zero, the on-time sample is at a peak.
- when decision statistics is positive, the on-time sample is too late. Next sampling instant= current sample time$+\delta$

Where, $\delta = 0.0625\times10^{-6}$ (sample time)

Then offset decision block output is fed back to phase delay of pulse generator which again controls the sampler enable. The initial sampling instant value should be updated by sending modified sampling instant to Simulink workspace and retrieving it from there, because the Simulink block does not accept value as feedback. After some iterations the decision statistics will become zero and sampling instant gets aligned with the peak of the signal.

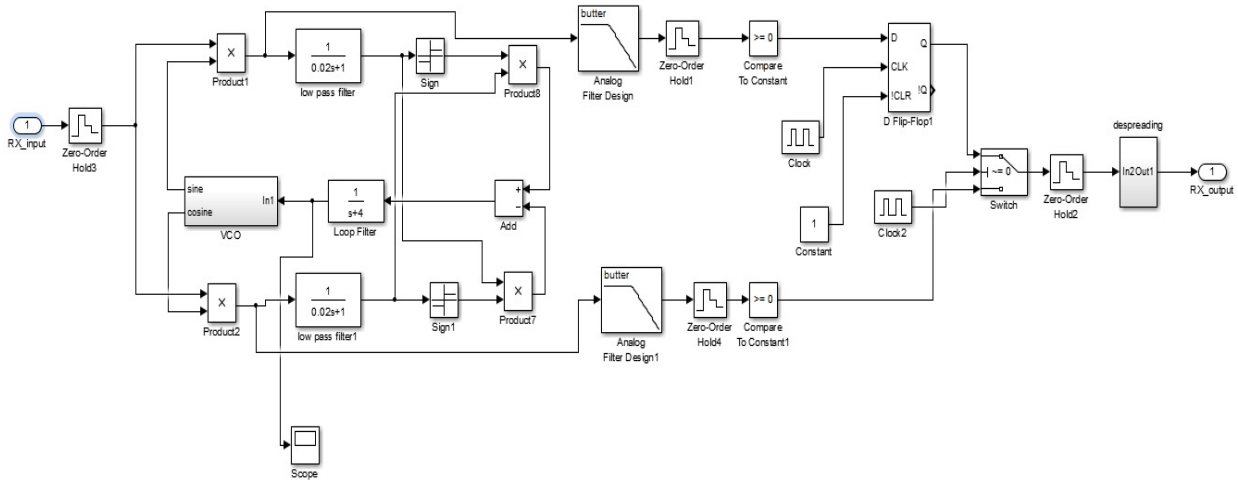

Fig. 4: Costas loop carrier recovery circuit in Simulink

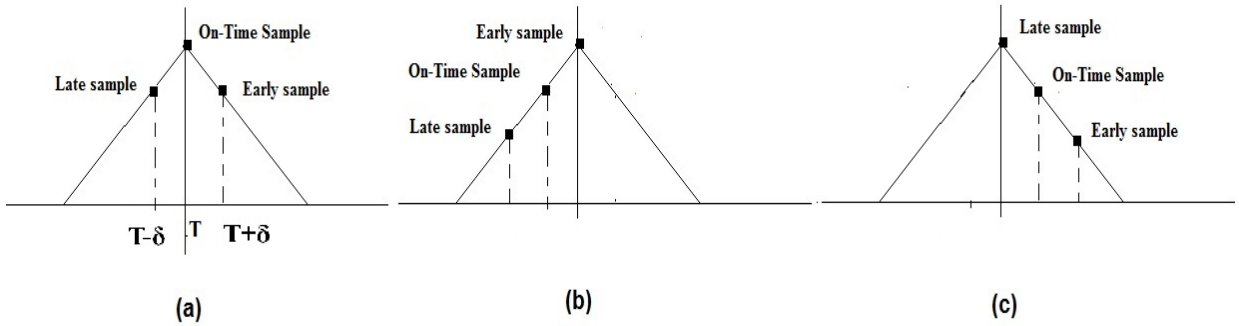

Fig. 5: Three Different Combinations of Sampling

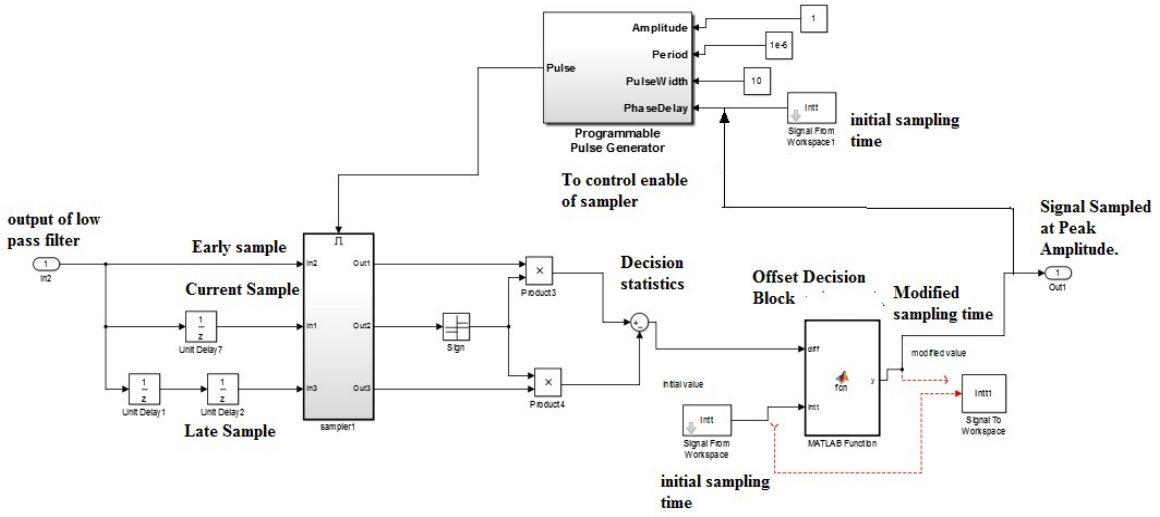

Fig. 6: Early-Late Gate Timing Recovery Block

III. Simulation Results

As discussed in carrier recovery algorithm, after some transient processes estimated phase will become equal to original phase of the carrier. then error value remains zero. In Fig.7. Matlab Plot between estimated phase and time. Initially the estimated

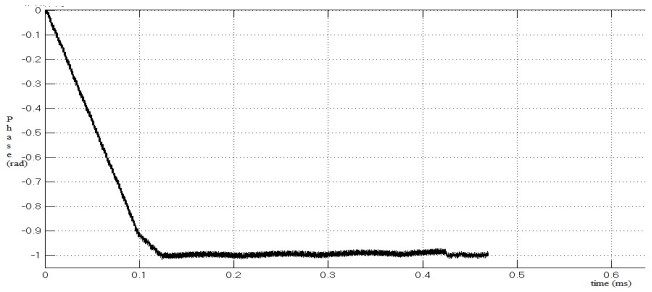

Fig. 7: Graph for estimated phase Vs time

phase is assumed to be zero. So the loop takes some time to reach the carrier phase value. However, once it reaches original phase value it stays there with small fluctuations.

The performance of this proposed system is analysed when it is passed through AWGN channel and Rayleigh fading channel at different SNR values ranging from -10dB to 14dB. In Rayleigh fading channel maximum doppler shift frequency is 100 Hz. Recovered bit stream gets generated after phase recovery, demodulation, base band demodulation using matched filter and early late gate timing recovery at sampler. It is then compared with input bit stream to get Bit error rate values. Fig.8. shows the graph between BER(Bit Error Rate) vs SNR when the signal undergone AWGN channel and Rayleigh fading channel. When SNR increases, BER value gradually decreases with negative 10 powers.

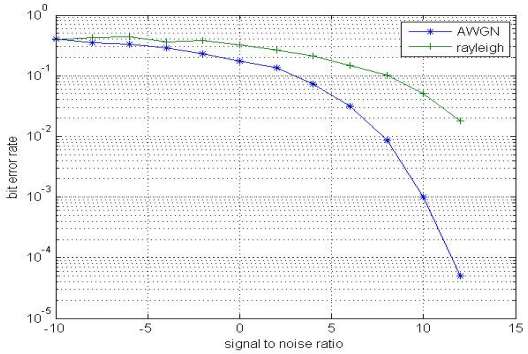

Fig. 8: BER curve for the system with Synchronization

The Zigbee transceiver with synchronization shows better performance when it undergone with some symbol timing delay and phase offset. Our proposed Zigbee transceiver applied to Rayleigh fading channel shows poor performance than when it fed to AWGN channel.

IV. Conclusion and Future Work

Carrier Synchronization by costas loop method and symbol synchronization by early late gate timing recovery method have been achieved in Zigbee transceiver.

Improvement in performance of the Zigbee transceiver has achieved by using symbol synchronization with Early-late gate timing recovery algorithm and carrier synchronization with costas loop Phase Locked Loop in Matlab Simulink.

Phase offset results rotation of the signal constellation. Channel delay introduces inter symbol interference and reduces noise margin. These synchronization techniques eliminates the effects of channel delay, phase offset. This paper thus proposes an effective receiver for Zigbee system which can be used at low power consumption, low data, low cost and short range communication networks like WSN applications.

Performance of the Zigbee transceiver can be further improved by using frame synchronization techniques. This model can be transformed into Hardware Description Language (HDL) and can be burned onto Field Programmable Gate Array (FPGA) to generate the Integrated Chips (ICs).

References

[1] G. Kolumbán, M. P. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos. II. chaotic modulation and chaotic synchronization," *IEEE Transactions on, Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, no. 11, pp. 1129–1140, 1998.

[2] C. Dick, F. Harris, and M. Rice, "Synchronization in software radios. carrier and timing recovery using FPGAs," *Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on*, pp. 195–204, 2000.

[3] N.-J. Oh and S.-G. Lee, "Building a 2.4-GHz radio transceiver using IEEE 802.15. 4," *Circuits and Devices Magazine, IEEE*, vol. 21, no. 6, pp. 43–51, 2006.

[4] W. Kluge, F. Poegel, H. Roller, M. Lange, T. Ferchland, L. Dathe, and D. Eggert, "A fully integrated 2.4-GHz IEEE 802.15. 4-compliant transceiver for ZigBee^{TM} applications," *IEEE Journal of, Solid-State Circuits*, vol. 41, no. 12, pp. 2767–2775, 2006.

[5] M. J. Kumar and M. R. Mehra, "Costas loop implementation for synchronous detection of carrier in radio receiver." IEEE, 2011, pp. 975–980.

[6] P. Shachi, R. Mishra, and R. K. Jatoth, "Coherent BPSK demodulator using costas loop and early-late gate synchronizer." IEEE, 2013, pp. 1–6.

[7] B. A. Bhatti, M. Umer, W. Ahmed, M. H. Tariq, and U. Ali, "Carrier and symbol synchronization in digital receivers using feedback compensation loop and early late gate on FPGA," *2014 International Conference on, Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, pp. 146–150, 2014.

[8] U. Devi, D. Madhavi, and A. Kumar, "Carrier recovery and clock recovery for QPSK demodulation," *International Journal of Research in Engineering and Technology*, 2014.

[9] W. C. Lindsey and C. M. Chie, "A survey of digital phase-locked loops," *Proceedings of the IEEE*, vol. 69, no. 4, pp. 410–431, 1981.

[10] R. E. Best, N. V. Kuznetsov, G. A. Leonov, M. V. Yuldashev, and R. V. Yuldashev, "Simulation of analog costas loop circuits," *International Journal of Automation and Computing*, vol. 11, no. 6, pp. 571–579, 2014.

[11] M. Yuldashev, *Mathematical models and simulation of Costas loops*. University of Jyväskylä, 2013.

[12] G. Kavya and V.-V. Mani, "Simulink model for zigbee transceiver using OQPSK modulation under fading channels," *4th International Conference on, Communication and Signal Processing*, 2015.

[13] M. Benssalah and M. Djeddou, "Implementation of synchronization algorithms, carrier phase recovery and symbol timing recovery, on a digital signal processor," *International Symposium on, Signals, Circuits and Systems, 2009. ISSCS 2009*, pp. 1–4, 2009.

[14] P. Zicari, P. Corsonello, and S. Perri, "A high flexible early-late gate bit synchronizer in FPGA-based software defined radios," *4th European Conference on, Circuits and Systems for Communications, 2008. ECCSC 2008*, 2008.