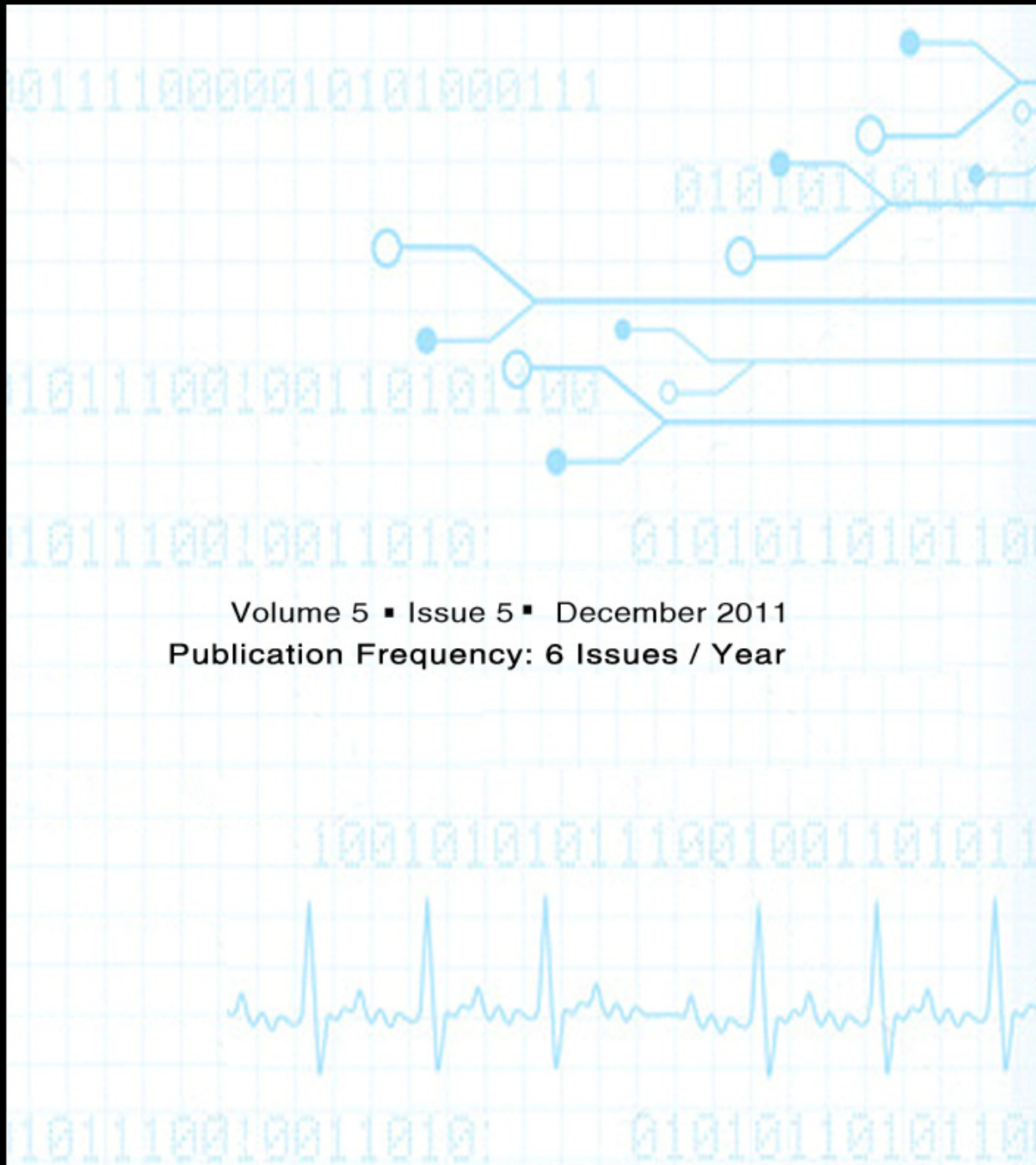


Editor-in-Chief  
Dr Saif alZahir

# SIGNAL PROCESSING (SPIJ)

AN INTERNATIONAL JOURNAL

ISSN : 1985-2339



Volume 5 ■ Issue 5 ■ December 2011  
Publication Frequency: 6 Issues / Year

Copyrights © 2011 Computer Science Journals. All rights reserved.

CSC PUBLISHERS  
<http://www.cscjournals.org>

# **SIGNAL PROCESSING: AN INTERNATIONAL JOURNAL (SPIJ)**

**VOLUME 5, ISSUE 5, 2011**

**EDITED BY  
DR. NABEEL TAHIR**

ISSN (Online): 1985-2339

International Journal of Computer Science and Security is published both in traditional paper form and in Internet. This journal is published at the website <http://www.cscjournals.org>, maintained by Computer Science Journals (CSC Journals), Malaysia.

SPIJ Journal is a part of CSC Publishers

Computer Science Journals

<http://www.cscjournals.org>

## **SIGNAL PROCESSING: AN INTERNATIONAL JOURNAL (SPIJ)**

Book: Volume 5, Issue 5, December 2011

Publishing Date: 15-12-2011

ISSN (Online): 1985-2339

This work is subjected to copyright. All rights are reserved whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provision of the copyright law 1965, in its current version, and permission of use must always be obtained from CSC Publishers.

SPIJ Journal is a part of CSC Publishers

<http://www.cscjournals.org>

© SPIJ Journal

Published in Malaysia

Typesetting: Camera-ready by author, data conversion by CSC Publishing Services – CSC Journals, Malaysia

**CSC Publishers, 2011**

## EDITORIAL PREFACE

This is fifth issue of volume five of the Signal Processing: An International Journal (SPIJ). SPIJ is an International refereed journal for publication of current research in signal processing technologies. SPIJ publishes research papers dealing primarily with the technological aspects of signal processing (analogue and digital) in new and emerging technologies. Publications of SPIJ are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics covers by SPIJ are Signal Filtering, Signal Processing Systems, Signal Processing Technology and Signal Theory etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, SPIJ appears in more focused issues. Besides normal publications, SPIJ intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of SPIJ is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position SPIJ as one of the top International journal in signal processing, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to signal processing fields.

SPIJ editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for SPIJ. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. SPIJ provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

### **Editorial Board Members**

Signal Processing: An International Journal (SPIJ)

## **EDITORIAL BOARD**

### **EDITOR-in-CHIEF (EiC)**

**Dr Saif alZahir**

University of N. British Columbia (Canada)

### **ASSOCIATE EDITORS (AEiCs)**

---

**Professor. Wilmar Hernandez**

Universidad Politecnica de Madrid  
Spain

**Dr. Tao WANG**

Universite Catholique de Louvain  
Belgium

**Dr. Francis F. Li**

The University of Salford  
United Kingdom

### **EDITORIAL BOARD MEMBERS (EBMs)**

---

**Dr. Thomas Yang**

Embry-Riddle Aeronautical University  
United States of America

**Dr. Jan Jurjens**

University Dortmund  
Germany

**Dr. Jyoti Singhai**

Maulana Azad National institute of Technology  
India

**Assistant Professor Weimin Huang**

Memorial University  
Canada

**Dr Lihong Zhang**

Memorial University  
Canada

## TABLE OF CONTENTS

Volume 5, Issue 5, December 2011

### Pages

- 185 - 202      Frequency and Power Estimator for Digital Receivers in Doppler Shift Environments  
*M. Saber, M. T. A. Khan, Y. Jitsumatsu*
- 203 – 214      Refining Underwater Target Localization and Tracking Estimates  
*C. Prabha, Supriya M. H., P. R. Saseendran Pillai*
- 215 - 226      Word Recognition in Continuous Speech and Speaker Independent by Means of Recurrent  
Self-Organizing Spiking Neurons  
*Tarek Behi, Najet Arous, Nouredine Ellouze*

# Frequency and Power Estimator for Digital Receivers in Doppler Shift Environments

**M. Saber**

*Department of Informatics  
Kyushu University  
744 Motooka, Nishi-ku, Fukuoka-shi, 89-0395, Japan*

*mohsaber@tsubaki.csce.kyushu-u.ac.jp*

**M. T. A. Khan**

*Ritsumeikan Asia Pacific University, College of Asia Pacific Studies  
1-1 Jumonjibaru, Beppu, Oita, 874-8577, Japan*

*tahir@apu.ac.jp*

**Y. Jitsumatsu**

*Department of Informatics  
Kyushu University  
744 Motooka, Nishi-ku, Fukuoka-shi, 89-0395, Japan*

*jitsumatsu@inf.kyushu-u.ac.jp*

---

## Abstract

A frequency estimator well suited for digital receivers is proposed. Accurate estimates of unknown frequency and power of input sinusoidal signal, in the presence of additive white Gaussian noise (AWGN), are provided. The proposed structure solve the problems of traditional phase locked loop (PLL) such as, narrow tracking range, overshoot, long settle time, double frequency ripples in the loop and stability. Proposed method can estimate frequencies up to half the sampling frequency irrespective of the input signal power. Furthermore, it provides stability and allows fast tracking for any changes in input frequency. The estimator is also implemented using field programmable gate array (FPGA), consumes 127 mW and works at a frequency of 225 MHz. Proposed method can estimate the fluctuation in frequency of transmitter's oscillator, can be used as a frequency shift keying receiver and can also be applied as a digital receiver in Doppler shift environment.

**Keywords:** Digital Phase Locked Loop (DPLL), Frequency Estimator, FPGA.

---

## 1. INTRODUCTION

The main objective of a receiver is to generate an accurate replica of the transmitted symbol sequence. Synchronization is very important for proper detection. It involves recovery of reference parameters from the received signal and using these parameters to demodulate and detect data [1]. Synchronization includes carrier recovery and symbol timing recovery. Carrier recovery is the estimation and compensation of carrier frequency and phase. In case of coherent detection, knowledge of both frequency and phase of the carrier is necessary. The frequency offset is generated at the receiver as a result of factors such as variations between the oscillators at transmitter and receiver, Doppler shift caused by relative motion between transmitter and receiver and the phase noise generated by other channel impairments [2]. The estimation of frequency of periodic signals in the presence of noise is also important in many other practical applications of signal processing.

PLLs are a very important building block in the modern communications. They are used in synthesizing carrier signals, demodulation and synchronization of frequency and phase of the received signal. Conventional PLL consists of three components, a phase detector (usually multiplier), a loop filter which is low pass filter (LPF) and voltage controlled oscillator (VCO) arranged in feedback manner. Conventional PLL suffers from problems such as high frequency ripples in the loop, overshoots and narrow tracking range. Higher order loop filters avoid these

problems; However use of higher order loop filters raise the issues such as stability, longer settling time, and more narrow tracking range [3:8].

We propose a simple design for frequency estimator which can estimate both frequency and power of received sinusoidal signal in the presence of background noise. Proposed algorithm can estimate wide range of frequencies up to half the sampling frequency, irrespective of the received signal power. The structure works in a feedback manner like a PLL and allows fast tracking for any changes in the received frequency; however, no loop filter is needed. The basic design is simple and comprises of only a first order system which provides stability. Computer simulations performed confirm the validity of analytical results. Furthermore, the estimator is modeled with VHDL and implemented using FPGA. Proposed structure's hardware implementation consumes 127 mW and works at a frequency of 225 MHz.

This paper is organized as follows: next section explains the working of conventional PLL and their characteristics. Section 3 presents the proposed method; explain the structure, operation and mathematical equations of proposed estimator. Computer simulations are presented in section 4. FPGA implementation is mentioned in section 5 and finally conclusions are given in section 6.

## 2. CONVENTIONAL PLL

A conventional PLL as shown in Fig. 1 consists of three main components i.e. a phase detector (multiplier), loop filter and VCO. PLL receives the input sinusoidal signal with input frequency and phase. The input signal is multiplied with the generated signal from VCO. Multiplication results in two signals the first with higher frequency term and the other with lower frequency term. Loop filter is assumed to remove the high frequency signal. The result output signal from loop filter is the low frequency signal which represents the frequency and phase difference between input and generated signals. This difference is used to control the generated phase and frequency from VCO. When there is no phase difference, VCO generates center frequency only.

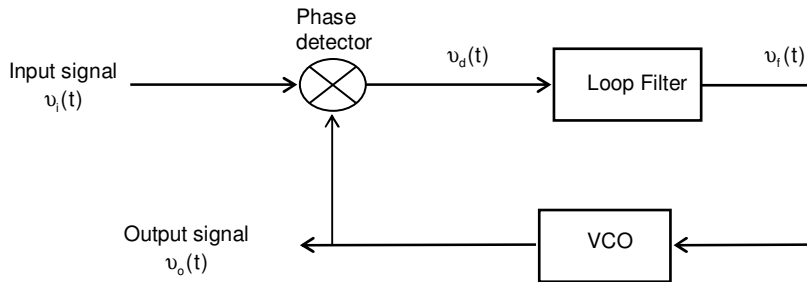


FIGURE 1: PLL Block diagram.

Assume an input sinusoidal signal

$$v_i(t) = A_i \sin(\omega_i t + \theta_i(t)) = A_i \sin \psi_i(t), \quad (1)$$

where  $\omega_i$  is the angular frequency and  $\theta_i(t)$  is the unknown phase of input signal. The signal generated by VCO is

$$v_o(t) = A_o \cos(\omega_o t + \theta_o(t)) = A_o \cos \psi_o(t), \quad (2)$$



where  $\omega_o$  is the estimation of angular frequency of VCO and  $\theta_o(t)$  is the estimated phase of VCO. Input signal is multiplied by the VCO output, then

$$\begin{aligned} v_d(t) &= k_m v_i(t) \times v_{vco}(t) \\ &= \frac{A_i A_o k_m}{2} \sin((\omega_i - \omega_o)t + \theta_i(t) - \theta_o(t)) + \frac{A_i A_o k_m}{2} \sin((\omega_i + \omega_o)t + \theta_i(t) + \theta_o(t)) \\ &= k_d [\sin(\psi_i(t) - \psi_o(t)) + \sin(\psi_i(t) + \psi_o(t))], \end{aligned} \quad (3)$$

where  $k_m$  is the gain of phase detector with dimension  $[1/V]$ ,  $k_d = \frac{A_i A_o k_m}{2}$ . In the simplest case we assume that low-pass filter removes the upper sideband with frequency  $\omega_i + \omega_o$  but passes the lower sideband  $\omega_i - \omega_o$ . VCO's tuning voltage will be

$$v_f(t) = k_d \sin(\psi_i(t) - \psi_o(t)) = k_d \sin \psi_e(t), \quad (4)$$

where  $\psi_e(t)$  is the phase difference between input and output VCO signals

$$\psi_e(t) = \psi_i(t) - \psi_o(t). \quad (5)$$

This difference will be used to control the frequency and phase generated by VCO. If the error signal is zero, VCO produces just its free running frequency ( $\omega_c$ , center frequency). If the error signal is other than zero, then VCO responds by changing its operating frequency.

$$\omega_o(t) = \omega_c + k_o v_f(t), \quad (6)$$

where the constant  $k_o$  is the gain of VCO. After integration of the above equation and substituting into (5), the phase difference is

$$\psi_e(t) = \psi_i(t) - \omega_c t - \int_{-\infty}^t k_o v_f(\tau) d\tau. \quad (7)$$

This can be rearranged as follows:

$$\psi_e(t) = \omega_i t - \omega_c t - \int_{-\infty}^t k_o k_d \sin \psi_e(\tau) d\tau. \quad (8)$$

Differentiating (8) w.r.t. 't' gives

$$\frac{d}{dt} \psi_e(t) = \Delta\omega - K \sin \psi_e(t) \quad (9)$$

where  $\Delta\omega = \omega_i - \omega_c$  and  $K = k_o k_d$  is the gain of PLL. The PLL continues to vary the phase of VCO  $\omega_o$  until locked, i.e. frequency and phase of the input signal are the same as those generated by VCO. After getting locked, PLL follows the changes in frequency and phase of input signal.

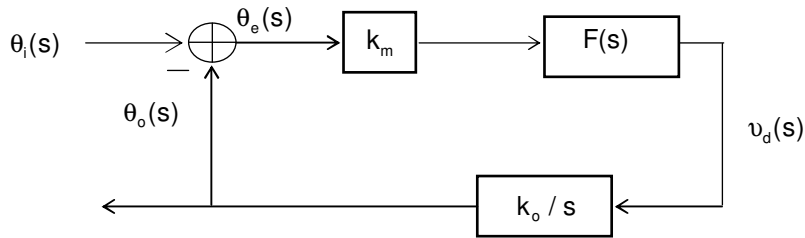
It can be concluded from the above analysis that the phase lock arrangement is described by the non-linear equation (9). Solution of this equation is not known for arbitrary values of  $\Delta\omega$ , and  $K$ . Without an aperiodic solution, the feedback system (PLL) cannot achieve phase stability, i.e.,

output frequency of VCO  $\omega_i$  will never be equal to input frequency  $\omega_o$ . Simplifications are needed to solve the equation [9]. One solution is the linear solution, in which we assume that, for small values of  $\psi_e(t)$

$$\sin \psi_e(t) \approx \psi_e(t) \tag{10}$$

The linear model of PLL in the continuous time-domain (S-domain) is shown in Fig. 2 [10]. Since phase is integral of the angular frequency, the VCO transfer function

$$\frac{\theta_o(s)}{v_i(s)} = \frac{k_o}{s} \tag{11}$$



**FIGURE 2:** Linear model of the PLL.

The transfer function of loop filter is  $F(s)$ . The closed loop transfer function of a PLL is

$$H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{k_m k_o F(s)}{s + k_m k_o F(s)} \tag{12}$$

$F(s)$  will have at least one pole (first order LPF) so the order of PLL always exceeds the order of loop filter by one. If we assume that the loop filter is first order LPF then the PLL will be a second order PLL [11].

A PLL has two main operation modes; Acquisition mode in which PLL is either out of lock or just starts to lock to a signal, the frequency range in which the PLL can acquire the input signal is called capture range. Tracking mode in which PLL is already locked to the input signal and begins to follow the changes in frequency and phase of input signal, the frequency range of PLL in which it can track the input signal is called tracking range. The tracking range is larger than capture range. The loop filter bandwidth is chosen according to the capture range and tracking range of PLL. There is a trade-off between the loop filter bandwidth and the noise in PLL. In case the bandwidth is narrow, the noise level is low but locking time will be long. While larger bandwidth implies large noise level but provides faster locking time [12:14].

**2.1 Drawbacks of PLL**

The performance of PLL in estimating and tracking frequency has some limitations such as

- 1- Double frequency ripples cannot be suppressed easily using first-order LPF, so higher order LPF is needed. The problem is that stability is not guaranteed for higher order PLLs [15].
- 2- All PLLs will exhibit some ringing or settling time. PLLs settling and lock times are primarily a function of the loop filter bandwidth. There is, in general, a trade off relationship between stability/noise immunity and settling/lock time [16].
- 3- The maximum allowable bandwidth of PLL is usually about 10% of the input frequency i.e. the tracking range for PLL is very limited and must be near the center frequency of

VCO, in case of first-order loop filter this range is  $f_{vco} + 0.1 \times f_{vco}$ . As the order of LPF increases tracking range of PLL decreases [17].

### 3. PROPOSED FREQUENCY ESTIMATOR

The proposed estimator consists of three components, i.e. frequency detector, accumulator and quadrature numerically controlled oscillator (NCO). The frequency detector component receives the two input sinusoidal quadrature signals and the two quadrature signals from NCO. Multiplication between these signals is done to obtain two quadrature signals which represent frequency difference between the input frequency and NCO signal frequency. These two sinusoidal signals are passed through a differentiator and squaring circuit to extract the frequency difference from the argument of sinusoidal signal. The final result from the frequency detector is the frequency difference term. This difference is directed to the accumulator which accumulates the differences until no difference is founded between input and generated signal frequencies. At this point the output of the accumulator represents input radian frequency and the generated signal from NCO runs at the same input frequency.

We first discuss the design of a digital receiver which can help in understanding our proposed method. A typical digital receiver structure implemented in software defined radio (SDR), except the radio frequency (RF) front-end, is shown in Fig. 3 [18]. The analog RF signal is received and fed through a low noise amplifier (LNA) to a mixer to convert RF to intermediate frequency. Signal is then passed to an analog to digital converter (ADC). A digital down-converter (DDC) converts ADC output to digitized real signal centered at zero frequency. DDC consists of direct digital synthesizer (DDS), low pass filter (LPF) and decimator. In addition to lowering frequency, DDC decimates the signal to a lower sampling rate allowing application of lower speed processors.

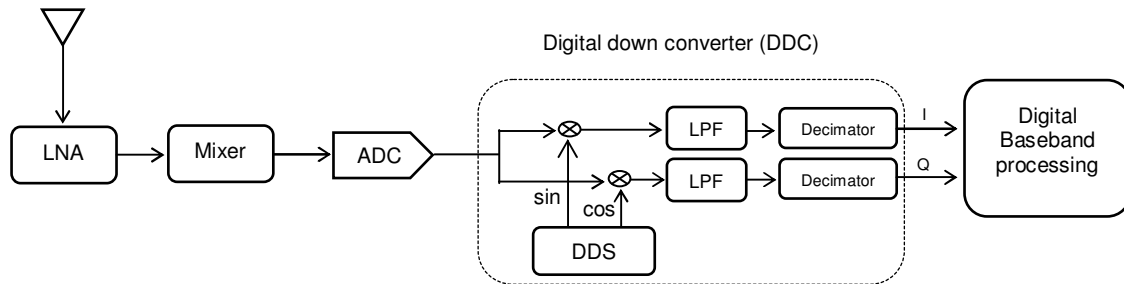
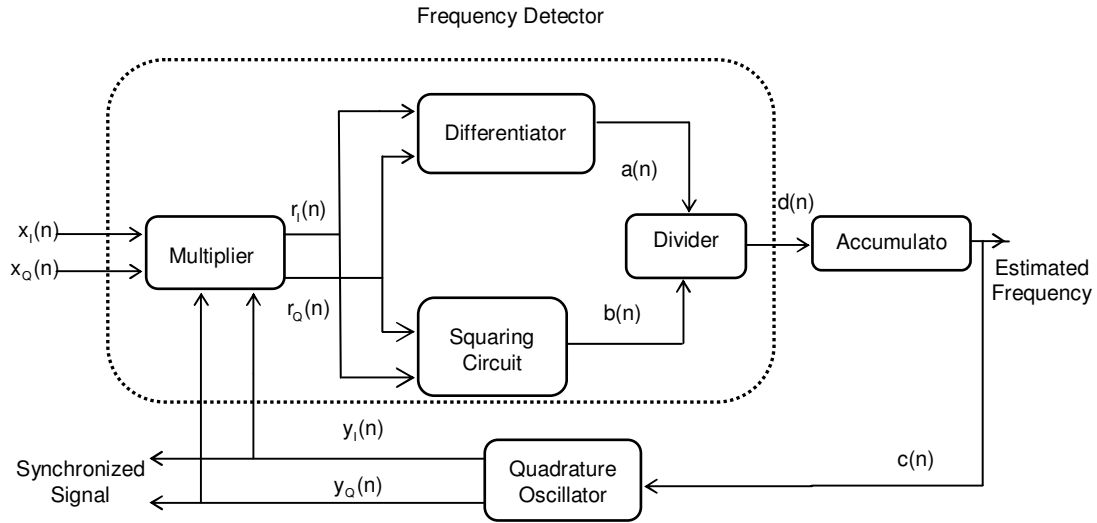


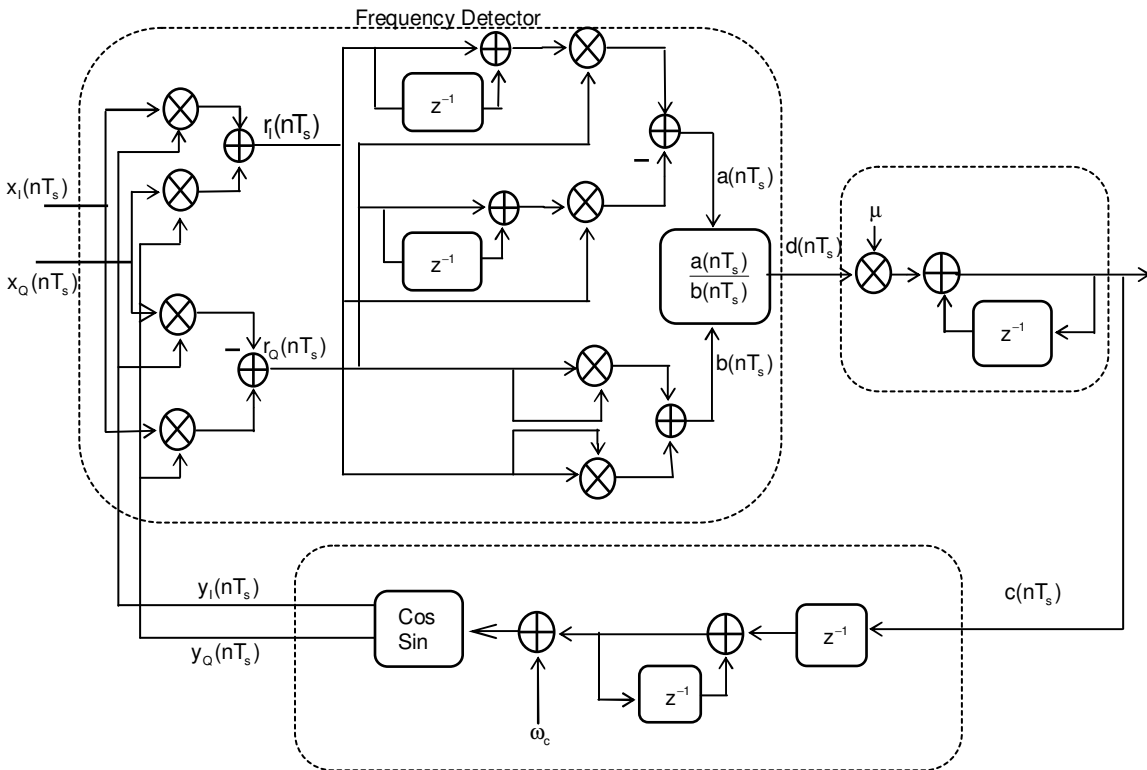
FIGURE 3: Digital receiver structure.

A block diagram of the proposed frequency estimator is given in Fig. 4. It is very similar to a PLL structure except that the phase detector and loop filter are replaced with a frequency detector and an accumulator respectively. The estimator receives the quadrature input signal from DDC, which is then sent to the frequency detector. Frequency detector consists of three components i.e., multiplier, differentiator and squaring circuit.



**FIGURE 4:** Block diagram of proposed estimator.

The block diagram of Fig. 4 can be implemented in analog or digital domain. Since the proposed estimator is designed mainly for digital receivers, description in the discrete time domain will be more accurate. Fig. 5 explains the detailed discrete time implementation of the proposed estimator structure. Mathematical and working details are explained below:



**FIGURE. 5** Discrete time model of proposed estimator.

### 3.1 Frequency and Power Estimation

We assume the input quadrature signal and the quadrature signal generated by the oscillator are

$$x_I(nT_s) = \sqrt{2p(nT_s)} \cos(\omega_i nT_s + \theta_i). \quad (13)$$

$$x_Q(nT_s) = \sqrt{2p(nT_s)} \sin(\omega_i nT_s + \theta_i). \quad (14)$$

$$y_I(nT_s) = \cos(\omega_o nT_s + \theta_o). \quad (15)$$

$$y_Q(nT_s) = \sin(\omega_o nT_s + \theta_o). \quad (16)$$

Where  $p(nT_s)$  is the power of input signal.  $\omega_i, \omega_o, \theta_i$  and  $\theta_o$  are the input and output radian frequencies and phases respectively,  $T_s$  is the sampling time.

#### A. Frequency detector

##### A.1. Multiplier

It multiplies input quadrature signals with quadrature signals of oscillator. This results in two quadrature signals ( $r_I(n), r_Q(n)$ ) which represent the difference between input frequency and generated frequency of oscillator.

$$r_I(nT_s) = x_I(nT_s)y_I(nT_s) + x_Q(nT_s)y_Q(nT_s) = \sqrt{2p(nT_s)} \cos(\Delta\omega nT_s + \Delta\theta), \quad (17)$$

$$r_Q(nT_s) = x_I(nT_s)y_Q(nT_s) - x_Q(nT_s)y_I(nT_s) = -\sqrt{2p(nT_s)} \sin(\Delta\omega nT_s + \Delta\theta). \quad (18)$$

Where  $\Delta\omega = \omega_i - \omega_o, \Delta\theta = \theta_i - \theta_o$ .

##### A.2. Differentiator

The quadrature signals ( $r_I(n), r_Q(n)$ ) pass through a differentiator circuit. This circuit is used to extract frequency difference from the argument of ( $r_I(n), r_Q(n)$ ). The output will be the quadrature signals multiplied by the frequency difference and amplitude of both signals.

$$\begin{aligned} \frac{d}{dt}r_I(t) \Big|_{t=nT_s} &\approx \{r_I(nT_s) - r_I(n-1)T_s\} \times \frac{1}{T_s} \\ &= -\sqrt{2p(nT_s)}\Delta\omega \sin(\Delta\omega nT_s + \Delta\theta) + \left(\frac{d}{dt}\sqrt{2p(t)}\right) \Big|_{t=nT_s} \times \cos(\Delta\omega nT_s + \Delta\theta), \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{d}{dt}r_Q(t) \Big|_{t=nT_s} &\approx \{r_Q(nT_s) - r_Q(n-1)T_s\} \times \frac{1}{T_s} \\ &= -\sqrt{2p(nT_s)}\Delta\omega \cos(\Delta\omega nT_s + \Delta\theta) - \left(\frac{d}{dt}\sqrt{2p(t)}\right) \Big|_{t=nT_s} \times \sin(\Delta\omega nT_s + \Delta\theta). \end{aligned} \quad (20)$$

$$\begin{aligned} a(nT_s) &= r_Q(nT_s) \times \frac{d}{dt}r_I(t) \Big|_{t=nT_s} - r_I(nT_s) \times \frac{d}{dt}r_Q(t) \Big|_{t=nT_s} \\ &= 2p(nT_s)\Delta\omega [\sin^2(\Delta\omega nT_s + \Delta\theta) + \cos^2(\Delta\omega nT_s)] \\ &= 2p(nT_s)\Delta\omega \end{aligned} \quad (21)$$

### A.3. Squaring circuit

The quadrature signal  $(r_i(nT_s), r_Q(nT_s))$  also passes through squaring circuit. Squaring circuit and divider eliminate the amplitude term from the result of differentiator. The output of squaring circuit can be considered as power of input signal.

$$\begin{aligned} b(nT_s) &= r_i^2(nT_s) + r_Q^2(nT_s) \\ &= 2p(nT_s)[\sin^2(\Delta\omega nT_s + \Delta\theta) + \cos^2(\Delta\omega nT_s)] \\ &= 2p(nT_s). \end{aligned} \quad (22)$$

### A.4. Divider

The divider is used to eliminate the amplitude part from (5). Its output is

$$d(nT_s) = \frac{a(nT_s)}{b(nT_s)} = \Delta\omega. \quad (23)$$

### B. Accumulator

Frequency difference is passed to the accumulator which accumulates this difference until saturation which means the input frequency is same as the generated frequency of oscillator. The accumulator action is given by the following equation

$$\mu d(nT_s) = c(nT_s) - c((n-1)T_s). \quad (24)$$

Where  $\mu$  is a variable whose value determines the speed of locking and stability of the system.

### C. Quadrature oscillator

The oscillator considered in this paper is numerically controlled oscillator (NCO). Output of the accumulator controls the value of generated frequency of NCO. The feedback loop of the estimator continues until there is no difference between the input and generated frequency. At this point, output of the accumulator represents the estimated input radian frequency. NCO equation is

$$\omega_o(nT_s) = \omega_c + \sum_{k=0}^{n-1} c(kT_s). \quad (25)$$

Where  $\omega_c$  is the center frequency. When the generated frequency from NCO is equal to the input frequency, the accumulator saturates. Thus at  $\omega_i(nT_s) = \omega_o(nT_s)$

$$c(nT_s) = \omega_i T_s = 2\pi f_i T_s \quad (26)$$

## 3.2 Stability

Stability of the estimator is analyzed in this sub-section. Linear model (z-model) of the system is built to get system's transfer function, which is the ratio between output and input signals. Fig. 6 shows z-model of the system. It may be noted that NCO equivalent model is only a delay circuit. The input is frequency and the output is also frequency so no integration is needed unlike PLL. Following equations are obtained from Fig. 6.

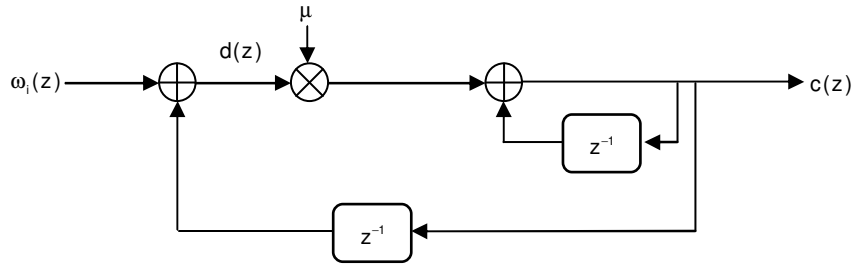


FIGURE. 6 Z-model of proposed estimator.

$$c(z) = \omega_i(z) - z^{-1}c(z), \quad (27)$$

$$\mu d(z) = c(z)(1 - z^{-1}), \quad (28)$$

$$\omega_i(z) = c(z)\left(z^{-1} + \frac{1 - z^{-1}}{\mu}\right). \quad (29)$$

Therefore, transfer function of the proposed estimator

$$T(z) = \frac{c(z)}{\omega_i(z)} = \frac{\mu z}{z - (1 - \mu)}. \quad (30)$$

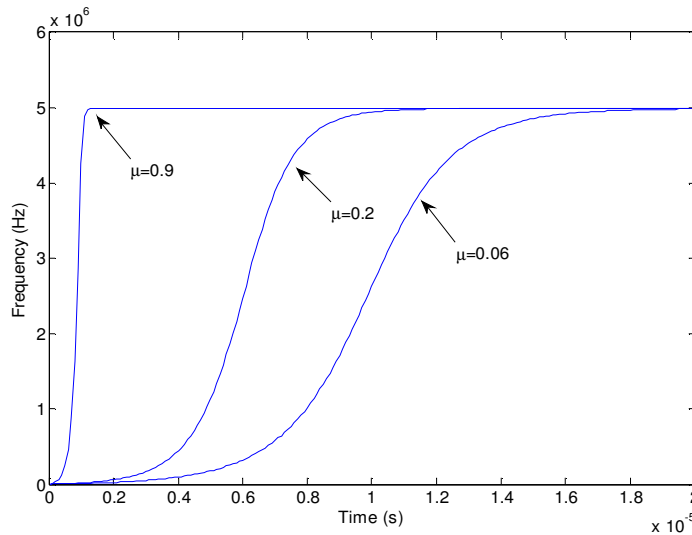
The variable  $\mu$  controls stability of the system and its range must be  $0 \geq \mu < 1$ . For small values of  $\mu$  the system is more stable but takes long settle time, for greater values of  $\mu$  settling time decreases.

#### 4. COMPUTER SIMULATIONS

Validity of the above mentioned mathematical analysis is checked by performing computer simulations. In this section we also present a comparison between conventional PLL and proposed estimator. The results of these simulations are discussed in the sub-sections below.

##### 4.1 Frequency Estimation

We first perform simulations for frequency estimation. The proposed system input frequency is set at  $f_i = 500$  kHz, sampling frequency  $f_s = 1/T_s = 10$  MHz with different values of  $\mu$ . Fig. 7 shows the output of the estimator for each value of  $\mu$ . It is clear that for small values of  $\mu$  frequency locking takes more time compared to that when  $\mu$  is large. Stability is guaranteed as  $\mu$  is in the specified range.



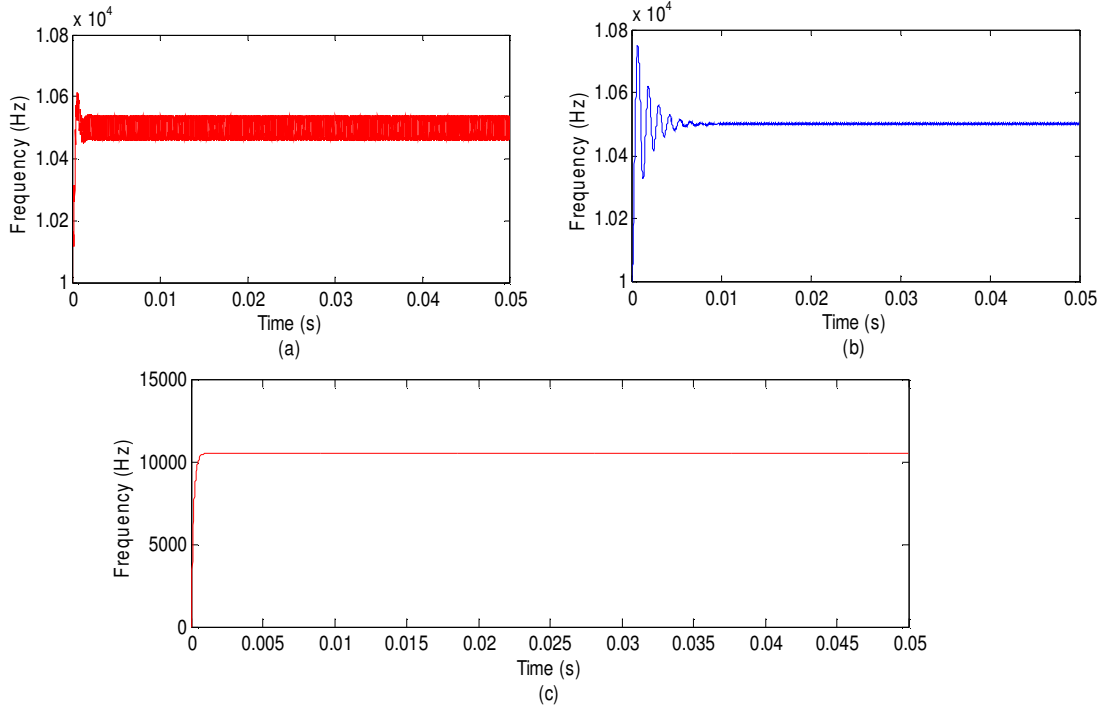
**FIGURE. 7** Frequency estimation with different  $\mu$  .

Following simulations provide a frequency estimation comparison of proposed and conventional PLL. An input signal with frequency of  $f_i = 10$  kHz is applied. In these simulations we used two different loop filters with the PLL; the first loop filter is first order LPF with cut-off frequency 1000 Hz, the other is second order LPF with the same cut-off frequency. The parameters of PLL are:  $k_m = 1/2$ ,  $k_o = 1024$ ,  $f_{in} = 10.5$  kHz,  $f_{nco} = 10$  kHz,  $f_{cut} = 1000$  Hz,  $f_s = 100$  kHz . While the parameters of proposed estimator are  $f_{nco} = 100$  Hz,  $f_s = 100$  kHz .

In Fig. 8 (a) conventional PLL with first-order LPF (second order PLL) did not suppress the double frequency ripple completely (variance is 665.3610), and overshoot (maximum overshoot reaches 1.06 KHz) occurs, settling time (the time required to reach correct estimated frequency is about 1 ms) [19]. In Fig. 8 (b) conventional PLL with second-order PLL (third order PLL) the ripple is suppressed better than the first-order LPF (variance is 1.2759), overshoots (maximum overshoot reaches 1.075 KHz) and ringing occur, settling time (increased to 7.5 ms seconds), estimation range is reduced and stability becomes critical.

In Fig. 8 (c) the proposed estimator estimates the input frequency at 0.5 ms with no ripples (Although no loop filter is used system variance is  $1.0839e-18$ ), the estimation range is wide from 0 to half the sampling frequency and more stable than PLL. The previous comparison clearly shows that the proposed estimator is faster, more stable, has a wide range than conventional PLL. The variance of estimations is also given in Table 1. It can be seen that estimation's variance is extremely low in case of the proposed method showing its superiority.





**FIGURE. 8** Frequency estimation curves of (a) Conventional PLL with first-order loop filter, (b) Conventional PLL with second-order loop filter, (c) Proposed Estimator.

Architecture	Variance	Settling time	Max. overshoots	Tracking range
Second order PLL	665.3610	1 ms	1.06 kHz	8950:11050 HZ
Third order PLL	1.2759	7.5 ms	1.075 KHz	9095:10904 Hz
Proposed estimator	1.0839E-18	0.5 ms	-	0:50 kHz

**TABLE 1:** Comparison between PLL and proposed estimator.

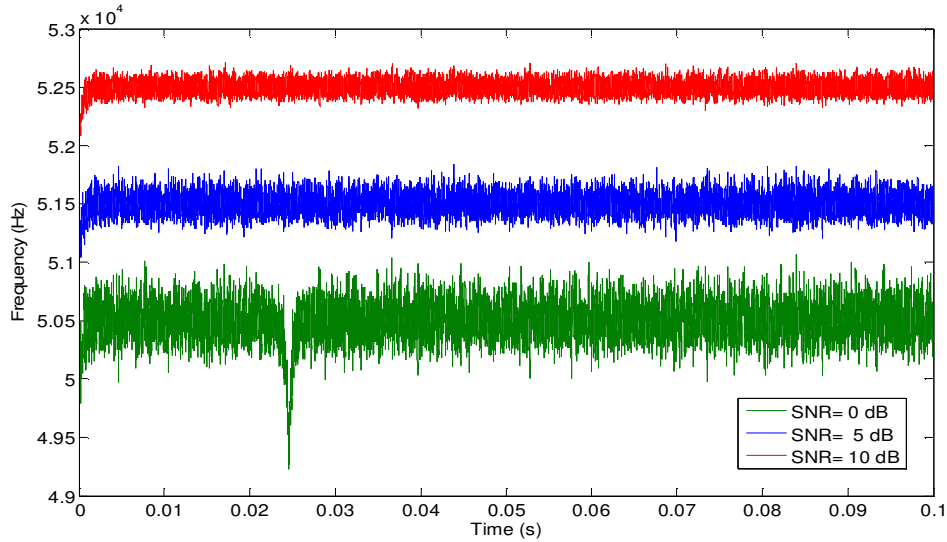
#### 4.2 Frequency Estimation With AWGN

In practical situations the received input signal is corrupted with AWGN and this will affect the results. Some design modifications are performed to take care of this situation. Block diagram of the modified design is shown in Fig. 15. Two moving average filters (MAF) with length of 10 stages are used for I and Q phase input signals. Another filter is used for smoothing the output. The MAF is a low pass finite impulse response (FIR) filter commonly used for smoothing an array of sampled data/signal. It takes M samples of input at a time and computes the average of those samples and produces a single output point. As the filter length (number of delays) increases the smoothness of output increases [20].

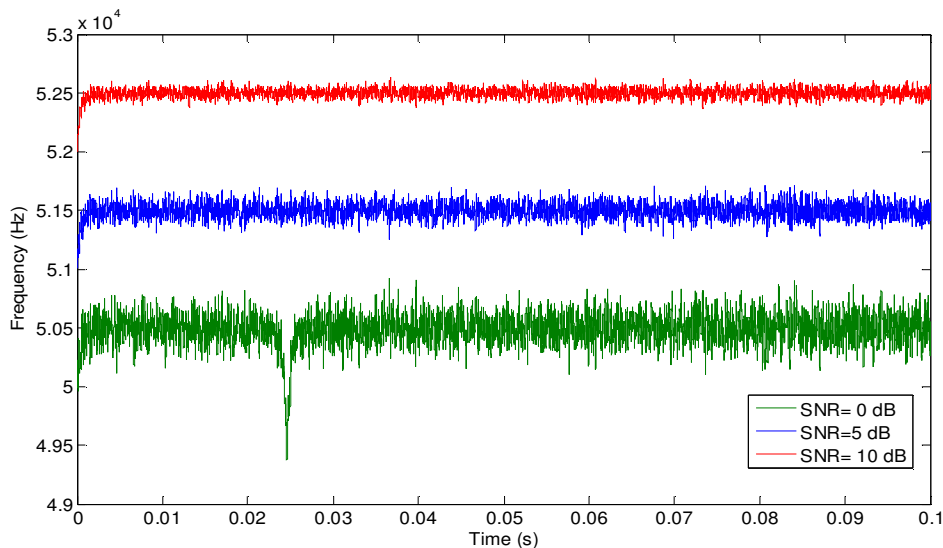
The following simulations are performed to investigate the effect of AWGN on both conventional DPLL (second and third order DPLL) and proposed estimator. MAF with the same lengths are added to both architectures. An input signal is applied to three architectures with different frequencies 5.05E4, 5.15E4, 5.25E4 Hz and SNR 0, 5, 10 dB at sampling frequency of 10 MHz.

Fig. 9 shows estimations of the second order DPLL with different frequencies and SNRs. At frequency of  $5.05E4$  and SNR =  $0$  dB (green line) the noise level affected the estimation and variance is  $2.2728E4$ . At frequency of  $5.15E4$  and SNR=  $5$  dB (blue line) the noise level is reduced and variance is  $6.6063E3$ . At frequency of  $5.25E4$  and SNR= $10$  dB (red line) a higher reduction in noise level causes the variance to further reduce to  $3.1311E3$ .

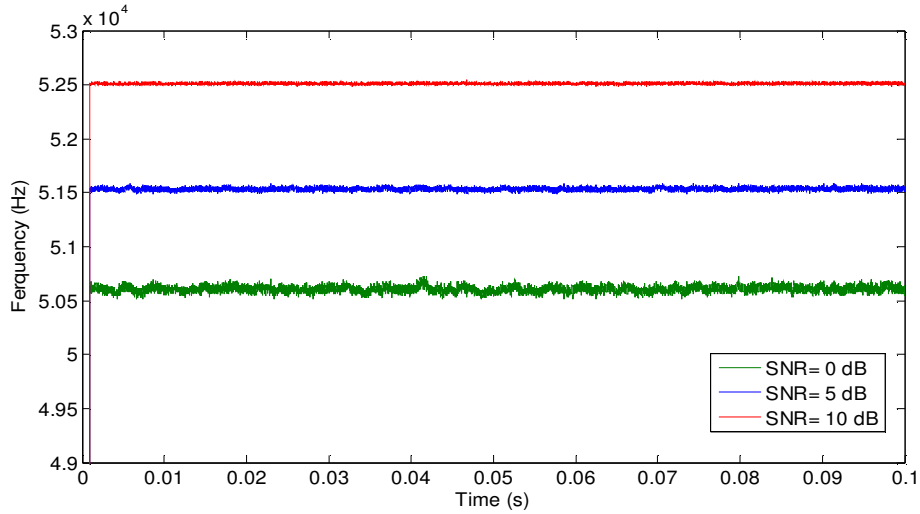
The estimations are improved using third order DPLL as shown in Fig. 10. The proposed estimator results are shown in Fig. 11. Variance of estimates of each architecture is summarized in Table. 2 it can be seen that, the proposed architecture produces the smallest values for each estimated frequency although loop filter is not used.



**FIGURE. 9** Frequency estimation with different SNRs using second order DPLL.



**FIGURE. 10** Frequency estimation with different SNRs using third order DPLL.



**FIGURE. 11** Frequency estimation with different SNRs using proposed estimator.

Architecture	SNR	Variance
Second order PLL	0 dB	2.2728E4
	5 dB	6.6063E3
	10 dB	3.1311E3
Third order PLL	0 dB	1.7288E4
	5 dB	4.1194E3
	10 dB	1.6340E3
Proposed estimator	0 dB	3.4036E-4
	5 dB	3.5282E-4
	10 dB	3.6620E-4

**TABLE 2:** Comparison between PLL and proposed estimator with different SNR.

### 4.3 Tracking Performance

We also checked performance of the proposed method in tracking frequency changes and estimating amplitude/power variations. Frequency of the input sinusoidal signal is varied randomly within the range 10 kHz-60 kHz. The input signal power is also randomly varied within the range 0.06125-0.21125 watt. The simulations are performed with SNR 10 dB and sampling frequency of 1 MHz. Fig. 12 shows the tracking of the input frequency while Fig. 13 gives estimation of power. It can be seen that the estimator tracks changes in input frequency and amplitude very fast.

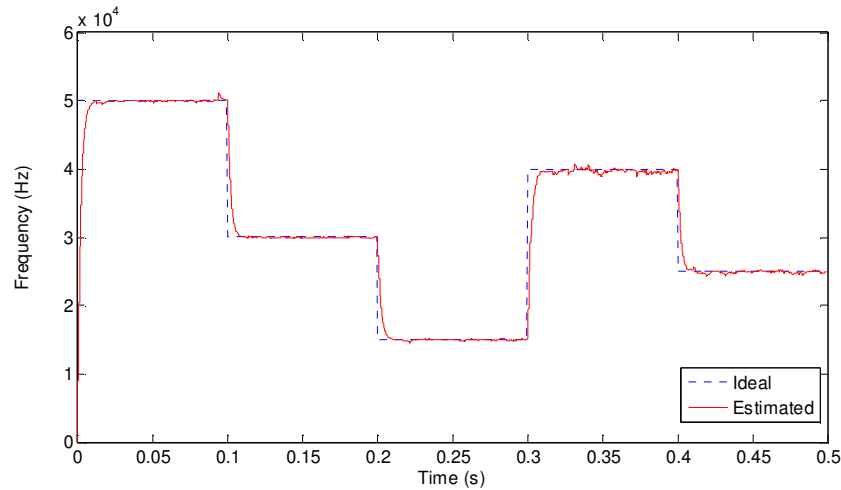


FIGURE. 12 Frequency tracking.

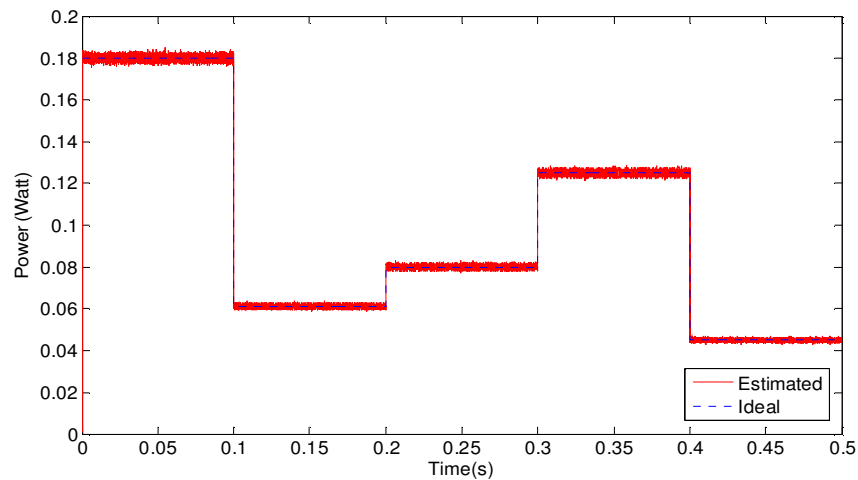
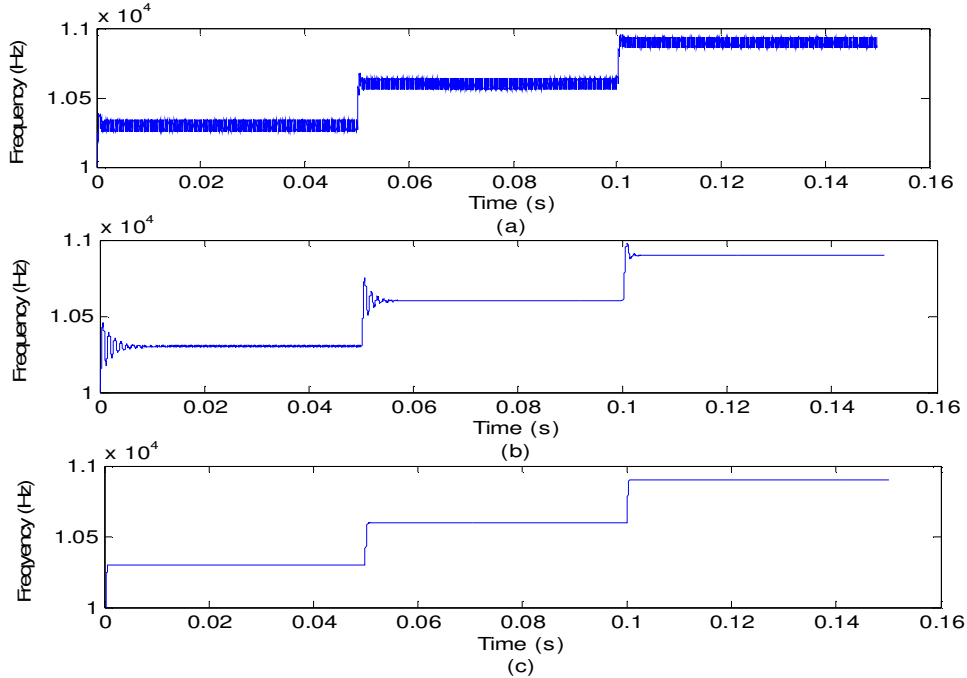


FIGURE. 13 Power estimation.

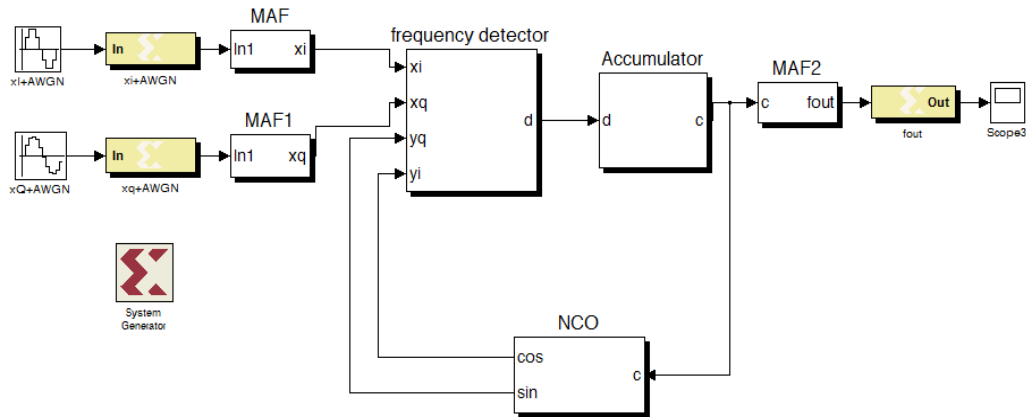
In order to investigate the performance of conventional PLL to track the frequency change, we have to change the previous estimation range because it exceeds PLL capabilities. A signal with input frequency changed from 10.3-10.9 kHz with sampling frequency of 100 kHz without AWGN is applied to conventional PLL with second-order LPF with  $f_{nco} = 10$  kHz. As shown in Fig. 14 for each tracked frequency the overshooting and ringing occurred. The situation becomes worst when the input signal have AWGN. As a result of using second-order LPF the estimation range is reduced to 10:10.9 kHz.



**FIGURE. 14** Frequency tracking using [1] Conventional second order DPLL. b) Conventional third order DPLL. c) Proposed estimator.

## 5. FPGA IMPLEMENTATION

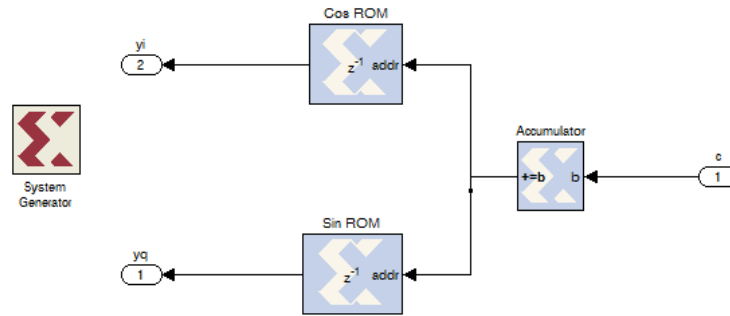
The estimator is modeled using Xilinx system generator tool [21,22]. VHDL code is used to describe the proposed estimator. Fig. 15 shows the system model used to generate the VHDL code [23, 24]. All signals of the estimator model are fixed point signals with 16 bits.



**FIGURE. 15.** System generator model of proposed estimator.

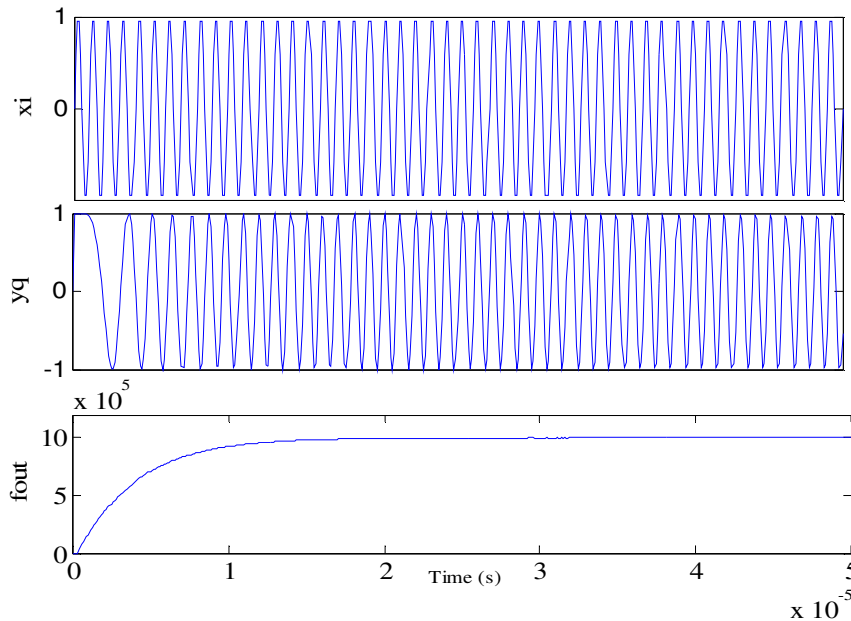
The model receives quadrature input signal with AWGN then the signals are passed to the moving average filter (MAF) to generate signals named 'xi' and 'xq'. The generated quadrature signals from NCO are 'yi' and 'yq'. These two quadrature signals are directed to the frequency detector block whose output is the signal 'd'. This signal enters the accumulator block to generate

the signal 'c'. Another MAF is used to enhance the output signal 'fout'. The model of quadrature NCO as shown in Fig. 16 consists of accumulator and two read only memory (ROM), one ROM stores sine waveform samples and the other stores cosine waveform samples. The accumulator receives the signal 'c' and generates both signals 'yi' and 'yq'. One rotation of the accumulator represents one complete cycle of sine or cosine waveform.



**FIGURE. 16.** System generator model of NCO.

The estimator was implemented using XCSD1800A-4FG676C Spartan-3A DSP board. Fig. 17 shows simulation results of the VHDL model when input and sampling frequencies are 1 MHz and 10 MHz respectively. It can be seen that the hardware model results agree with the simulations in locking time and stability. FPGA resources utilization for proposed architecture, which indicates how many hardware components are used by the model, is provided in Table 3. It can be seen that the proposed method requires fewer hardware resources. Implementation results indicate that the proposed design provides fast operation and low power consumption. Furthermore, it can be observed that the method has a simple design and avoids complexities.



**FIGURE. 17.** VHDL simulation results.

Component	used	Avilable	Utilization
Number of slice flip flops	76	33280	1%
Value Number of 4 input LUTs	583	33280	1%
Number of occupied slices	320	16640	1%
Number of DSP48As	6	84	7%
Number of Bonded IOBs	81	519	15%
Maximum frequency	225 MHz		
Power consumption	127 mW		

**TABLE 3:** FPGA resources utilization.

## 6. CONCLUSIONS

PLL is wide used in communication application such as frequency tracking. Conventional PLL has many drawbacks such as overshoot, ringing, limited tracking range and instability. The aim of this paper is to present a proposed estimator solve limitation in conventional PLL. The proposed estimator accurately estimates unknown frequency and power of input sinusoidal signal simultaneously. Proposed method worked well even in the presence of AWGN. The estimator rapidly tracked any changes in input signal frequency and amplitude. Operation of the proposed structure is similar to that of a PLL. The phase detector component and the low pass filter in PLL have been replaced with frequency detector and accumulator. Computer simulations performed showed promising results. The paper also discussed hardware implementation of the system. The proposed architecture is designed and modeled using VHDL and implemented using FPGA circuit. Implementation results indicate that the estimator has a simple design, faster operation and low power consumption.

## 7. ACKNOWLEDGEMENT

This research is partially supported by Grant-in-Aid for scientific Research (B) no.20360174, and the Aihara Project, the FIRST program from JSPS, initiated by CSTP.

## 8. REFERENCES

- [1] B. Sklar. Digital Communications: Fundamentals and Applications. Prentice Hall Inc.,2001.
- [2] J. G. Proakis, M. Salehi. Digital Communications. Mcgraw-Hill., 2008.
- [3] D. Green. "Lock-In, Tracking, and acquisition of AGC-aided phase-locked loops." IEEE Transaction on Ciruits and Systems, vol. 32, pp. 559-568,1985.
- [4] S. Changhong, C. Zhongze, Z. Lijun, L. Yong. "Design and Implementation of Bandwidth Adaptable Third-order All digital Phase-Locked Loops," in Proc. WICOM, 2010, pp.1-4.
- [5] P. Hanumolu. "W Gu-Yeon, M. un-ku, A Wide- Tracking Range clock and Data recovery Circuit." IEEE Transaction on Solid state circuits, vol. 43, pp. 425:439, 2008.
- [6] X. hongbing, G. Peiyuan, L. Shiyi. "Modeling and simulation of Higher-order PLL rotation tracking system in GNSS receiver," in Proc. ICMA, 2009, pp. 1128-1133.
- [7] A. Carlosena, A. Manuel- Lazaro. "Design of Higher-Order Phase-Lock Loops." IEEE Transaction on Circuits and systems II, vol. 54, pp. 9-13, 2007.
- [8] B. Razavi, Monolithic Phase-Locked Loops and clock recovery circuits: Theory and Design, Wiley-IEEE Press, 1996.

- [9] V.F. Kroupa. Phase Lock Loops and Frequency Synthesis. John Wiley & Sons, 2003.
- [10] F.M Gardner. Phase lock techniques. Wiley-Interscience, 2005.
- [11] R. E. Best. Phase locked loops Design, simulation, and Application. Mcgraw-Hill, 2007.
- [12] L. Kyoohyun, C. Seunghee, K. Bemsup. "Optimal loop bandwidth design for low noise PLL applications," in Proc. ASP-DAC,1997, pp. 425-428.
- [13] W. Ping-Ying, F. Chia-Huang. "All digital modulation bandwidth extension technique for narrow bandwidth analog fractional-N PLL," in Proc. ESSCIRC,2010, pp. 270-273.
- [14] J.Roche, W. Rahadjandrabey, L. Zady, G. Bracmard, D. Fronte. "A PLL with loop bandwidth enhancement for low-noise and fast-settling clock recovery," in Proc. ICECS, 2008, pp. 802-805.
- [15] M. Saber, Y. Jitsumatsu, M.T.A. Khan. "Design and Implementation of Low Ripple Low Power Digital Phase-Locked Loop." Signal Processing: An international journal (SPIJ), vol. 4, pp. 304-317, Feb. 2011.
- [16] M. Min, V. Mannama, T. Paavle."Settling time minimization in PLL frequency synthesizer," in Proc. ICCSC, 2005, pp. 366-369.
- [17] L. Verrazzani . "Pull-In Time and Range of Any Order Generalized PLL. " IEEE Transaction on Aerospace and Electronic systems, vol. AES14, pp.329-333, 2007.
- [18] M. Dillinger, K. Madani, N. Alonistioti. Software Defined radio: Architectures, Systems, and Functions. Wiley, 2003.
- [19] M. J.Roberts, R.A. Riccardo. Student's Guide to Analysis of Variance. Routledge, 1999.
- [20] W.S. Steven. The Scientist & Engineer's Guide to Digital Signal Processing. California Technical Pub,1997.
- [21] P. Pong Chu. RTL Hardware Design Using VHDL: Coding for Efficiency, Portability and Scalability. Wiley-IEEE Press, 2006.
- [22] Xilinx Inc. system generator for DSP user guide. Xilinx, 2009.
- [23] W.Y. Yang. Matlab/Simulink for digital communication. A-Jin, 2009.
- [24] Pong p. chu. FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version. Wiley-Interscience, 2008.



# Refining Underwater Target Localization and Tracking Estimates

**C. Prabha**

*Department of Electronics  
Cochin University of Science and Technology  
Kochi – 682 022, Kerala, India.*

*prabhasuma@gmail.com*

**Supriya M. H.**

*Department of Electronics  
Cochin University of Science and Technology  
Kochi – 682 022, Kerala, India.*

*supriya@cusat.ac.in*

**P. R. Saseendran Pillai**

*Department of Electronics  
Cochin University of Science and Technology  
Kochi – 682 022, Kerala, India.*

*prspillai@cusat.ac.in*

---

## Abstract

Improving the accuracy and reliability of the localization estimates and tracking of underwater targets is a constant quest in ocean surveillance operations. The localization estimates may vary owing to various noises and interferences such as sensor errors and environmental noises. Even though adaptive filters like the Kalman filter subdue these problems and yield dependable results, targets that undergo maneuvering can cause incomprehensible errors, unless suitable corrective measures are implemented. Simulation studies on improving the localization and tracking estimates for a stationary target as well as a moving target including the maneuvering situations are presented in this paper.

**Keywords:** Localization, Tracking, Maneuvering, Kalman filter.

---

## 1. INTRODUCTION

Underwater noise sources can be categorized mainly as natural and man-made, based on their source of origin. Processing of the received acoustic signals helps in locating, tracking or even identification of the noise source. Localization and tracking of underwater targets bear lots of significance and has attracted great attention in the past few decades due to its importance in oceanographic, fisheries and military applications. One of the main requisites of surveillance operations is the precise position estimates of targets, which is implemented using certain geometrical constructions and well known algorithms on real time data from such systems. Various techniques for underwater localization and tracking have been devised, one of which utilizes the acoustic emanations from the targets in question by making use of passive listening concepts [1-3].

Improving the localization and tracking estimates of targets using various techniques in an underwater scenario is a problem of unfathomable extent, owing to the characteristics of the ambient environment. As the localization estimates may vary due to sensor and environmental errors, Kalman filtering techniques are applied to obtain reliably accurate estimates of localization. The results of tracked targets can be mostly misleading, if enough measures for minimizing errors in every stage of the system are not employed. One of the major problems faced by underwater target tracking systems is the effects of noises of various forms, right from the ambient noises to system induced noises, which have to be dealt with for reliable results. Adaptive filters like the Kalman filter are very powerful tools to ward off the noises that affect the reliability of such a system [4, 5].

Target tracking systems basically produce a stream of data related to the position of the target. This problem can be further divided into one dimensional motion of the target with inherent noises of different forms such as process noise and measurement noise. A study of one dimensional system is carried out and then extended to two dimensions, which can further be generalized to a multi-dimensional system depending on the nature of the problem.

The observed errors in the case of a maneuvering target are far more complex in nature than the one in the case of a target which is moving with constant velocity and hence need to be mitigated by using suitable estimation techniques. The main cause of such errors in tracking targets is their stochastic maneuvering, which becomes difficult to be identified by the tracking device. This major issue is also associated with tracking of maneuvering targets with highly adaptive generic filters like the Kalman filter, since these filters end up producing an output, erroneously considering the measured values in response to the maneuvering target, as noise. Hence optimizing the performance of the Kalman filter in a maneuvering target scenario warrants certain modifications which are discussed in the final section of this paper.

## 2. KALMAN FILTER

Kalman filter is a recursive approximation algorithm that provides an efficient computational means to estimate the state of a dynamic system from a series of incomplete and erroneous measurements [6-9]. This filter supports estimation of past, present and even future states and it can do so even when the precise nature of the system is unknown. Unlike most of the data processing concepts, the Kalman filter does not require all the previous data to be stored and reprocessed for each new measurement which simplifies the practical implementation of the filter. This optimal linear estimator helps to refine the localization measurements and leads to more reliable position information by judiciously taking care of the variances in the measurements.

### 2.1 System Model

Given a physical system, a mathematical model is developed that adequately represents some aspects of the behavior of the system, termed as system model. The structure and modes of responses of a system can be investigated using such mathematical models supplemented with appropriate mathematical tools. In order to observe actual system behavior, measurement devices are constructed to output data signals, proportional to certain variables of interest. These output signals and the known inputs to the system are the only information that is discernable about the system behavior [10, 11]

The general problem of estimating the state variable  $x$ , of a discrete-time controlled random process, that is governed by the linear stochastic difference equation can be expressed as,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k, \quad (1)$$

with a measurement  $z_k$ , that is

$$z_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (2)$$

The matrix  $\mathbf{A}$  in the difference equation relates the state  $x$  at the current time step  $k$  to the state  $x$  at the next time step  $k+1$ , in the absence of either an optional control function  $\mathbf{u}$  or process noise  $\mathbf{w}$ . The matrix  $\mathbf{B}$  relates the optional control input to the state, while the matrix  $\mathbf{H}$  in the measurement equation relates the state to the measurement. The process is presumed to be stationary and hence the matrices are considered as constants. The normal probability distribution random variables  $\mathbf{w}_k$  and  $\mathbf{v}_k$  represent the process and measurement noise respectively and are assumed to be independent and white with constant covariance.

### 2.2 Algorithm

The Kalman filter is essentially a set of mathematical equations that implements a predictor-corrector type estimator which minimizes the mean square error. The time update equations or the predictor equations are responsible for projecting forward the current state and error

covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations or corrector equations are responsible for mapping the predicted values into the a priori estimate to obtain an improved a posteriori estimate.

The following are the Predictor/corrected Kalman filter Equations.

Predictor Equations:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k \quad (3)$$

$$\mathbf{P}_{k+1}^- = \mathbf{A}\mathbf{P}_k \mathbf{A}^T + \mathbf{Q} \quad (4)$$

Corrector Equations:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (5)$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \mathbf{H} \hat{\mathbf{x}}_{k+1}^-) \quad (6)$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H})\mathbf{P}_{k+1}^- \quad (7)$$

$\hat{\mathbf{x}}_{k+1}^-$ , above is the a priori state estimate at step  $k+1$ , which is the estimate of the state based on measurements at previous time-steps and  $\hat{\mathbf{x}}_{k+1}$  is the a posteriori state estimate at step  $k+1$ , given measurement  $\mathbf{z}_{k+1}$ . The a priori estimate error covariance is given by  $\mathbf{P}_{k+1}^- = E [e_{k+1}^- e_{k+1}^{-T}]$ , and the a posteriori estimate error covariance by,  $\mathbf{P}_{k+1} = E [e_{k+1} e_{k+1}^T]$ , where the a priori estimate error is  $e_{k+1}^- = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-$  and the a posteriori estimate error is  $e_{k+1} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}$ . The matrix  $\mathbf{K}$  is the Kalman gain or blending factor that minimizes the a posteriori error covariance. The difference  $(\mathbf{z}_{k+1} - \mathbf{H} \hat{\mathbf{x}}_{k+1}^-)$  is called the measurement innovation or the residual. The residual reflects the discrepancy between the predicted measurement and the actual measurement. A residual of zero means that the two are in perfect agreement. The  $\mathbf{Q}$  and  $\mathbf{R}$  values represent process and measurement noise covariance respectively.

### 3. SCENARIO OVERVIEW

Simulation of underwater target localization is carried out using an ocean surveillance system consists of sensor networks that has to be deployed in the ocean which compute location of the target by measuring angles to it, from known positions of the sensor nodes using passive listening concepts [12, 13]. The results of localization are applied to the Kalman filter so as to minimize the error leading to more accurate estimates of the localization information. This paper considers stationary target as well as moving target, represented in Cartesian co-ordinate system for analysis. Suitable transformations can be used if the measurement data are in a format other than the Cartesian system. However, the tracking system and design challenges are relatively insensitive to the choice of the co-ordinate system [14].

A target moving with nearly constant velocity is characterized by a state vector with position and velocities as elements. The observations made can be assumed as a linear combination of the state vector corrupted by additive measurement noise. The Kalman gain is used to derive the filtered estimates of the state vector which in turn is used to compute the estimates predicted for the next measurement state.

The residual value is the difference between the observed and predicted values. In addition to being used for updating the filtered estimates, the residual values can be checked for consistency. This consistency check can be used to adjust the filter parameters when large

residual values are interpreted as due to increased target dynamics or the detection of maneuvering of the target. The estimation accuracy provided by the Kalman filter through the covariance matrix is useful for detection of maneuver. Upon detecting such maneuver, the Kalman filter also provides an efficient way to adapt to a scenario of varying target dynamics [15-17].

### 3.1 Improving localization Estimates of a Stationary Target

The latitude longitude pair obtained from the localizer [12, 13] may not be accurate due to the variations in the estimation of direction of arrival of the signals emanating from the target and the mathematical approximations involved in the range computation. These inaccuracies are resolved to a certain extent by applying the concepts of Kalman filter, making use of which the refinement of localization of the underwater target is carried out by reducing the mean square error. Here the target is assumed to be stationary and the filter is applied in both dimensions independently in order to get more accurate position estimates.

### 3.2 Tracking of a Moving Target

In this model, the state vector consists of the target position and velocity. The elementary laws of motion can be applied for computing the velocity  $v$  for an arbitrary time step  $k+1$  and can be written as  $v_{k+1} = v_k + Tu_k$  where  $u$  is acceleration and  $T$  is the time interval. This velocity will be perturbed by noise due to the wave action and other physical parameters of the ocean. Hence a more realistic equation for velocity  $v$  is

$$v_{k+1} = v_k + Tu_k + \tilde{v}_k \tag{8}$$

where  $\tilde{v}$  is the velocity noise. A similar equation for position  $s$  can be expressed as,

$$s_{k+1} = s_k + Tv_k + \frac{1}{2} T^2 u_k + \tilde{s}_k \tag{9}$$

where  $\tilde{s}$  is the position noise.

For an  $n$  dimensional system, the state vector at time step  $k$ , can be described as,

$$\mathbf{x}_k = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \\ v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{10}$$

For a target moving in one dimension,

$$\mathbf{x}_k = \begin{bmatrix} s_1 \\ v_1 \end{bmatrix} \tag{11}$$

Since the measurement vector contains only the position element, the linear system equations can be represented as,

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \mathbf{u}_k + \mathbf{w}_k \tag{12}$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \tag{13}$$

Process noise  $w_k$ , represents the trajectory perturbations due to uncertainty in the target state whereas the measurement noise  $v_k$ , represents the inability of the tracking device to precisely

measure the position of the target due to unavoidable errors in the measurement system. Both these noises are assumed to be random Gaussian processes. The acceleration  $u$  can be assumed to be zero without disturbing the generality of the system for a target moving with a constant velocity.

When the target is moving in two dimensions with a constant velocity, the state, prediction and correction equations of the model are the same as that of the one dimensional scenario, except that all the vectors are of dimension 2.

The true position of the target at the time  $k+1$ , given the position at time  $k$  is:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \quad (14)$$

The state vector at time step  $k$ , when  $n= 2$  is,

$$\mathbf{x}_k = \begin{bmatrix} s_1 \\ s_2 \\ v_1 \\ v_2 \end{bmatrix} \quad (15)$$

and the state transition model  $\mathbf{A}$  is:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \quad (16)$$

where  $\mathbf{I}_2$  represents the identity matrix of order 2.

The state vector keeps track of the positions of the target and velocities in different dimensions which usually are the X and Y dimensions. The purpose of the Kalman filter is to estimate the true state vector given a series of discrete measurements. The state transition model updates the state vector in each time step by updating each position by adding the time interval between each measurement multiplied by the velocity in the same dimension.

Again, the measurement vector is a function of the state vector and a random noise process, expressed as,

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (17)$$

where the measurement vector is: 
$$\mathbf{z}_k = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad (18)$$

and the observation model  $\mathbf{H}$  is: 
$$\mathbf{H} = [\mathbf{I}_2 \quad \mathbf{0}] \quad (19)$$

As the velocity is not measured directly, the observation model  $\mathbf{H}$  is operated on the state vector to obtain the measurement vector.

### 3.3 Tracking of a Maneuvering Target

The standard Kalman filter cannot be applied while considering a maneuvering target that executes a turn or an evasive action to elude the detection, since the target movement appears as an extensive process noise on the target model which cannot be circumvented by the process noise variance.

In order to detect a maneuver, the difference between each measurement and its corresponding predicted value is computed, which is called residual or innovation. When the number of components in each measurement is more than one, a normalized distance function or total distance,  $d^2$  is computed. This is done by squaring the differences in each of the component measurements, dividing by the respective error variances and then summed to form a total normalized distance. A generalized form of normalized distance function can be formed with the application of Kalman filter by using the residual vector  $\tilde{z}_k$  and the residual covariance matrix  $S$ ,

$$d_k^2 = \tilde{z}_k^T S^{-1} \tilde{z}_k \quad (20)$$

$$\text{where } \tilde{z}_k = z_k - H\hat{x}_k^- \text{ and } S = S_k^- = HP_k^- H^T + R \quad (21)$$

A maximum allowable value for the residual is set using the accuracy statistics of the prediction and measurement values and is normally set to at least thrice the residual standard deviation assuming zero mean Gaussian statistics, for one dimensional movement of the target. The computed differences are compared with the above derived maximum allowable error value and if the difference exceeds the same, a target maneuver is considered as detected.

Since in the subject case, the target has two dimensions of physical freedom, the normalized distance function is the sum of squares of two zero mean, unit standard deviation Gaussians, and thus featuring a chi-square probability distribution with degrees of freedom equal to the number of the measurement dimensions which in this case is 2. Based on this chi square table for  $d^2$ , a threshold can be determined to detect the target maneuver [14, 16]. Once the maneuver is detected, the Kalman filter parameters are reset and the filter is reinitialized using the last two measurements

#### 4. SIMULATION

The simulation of improving localization and tracking estimates using Kalman filter has been implemented using Matlab. The erroneous localizer output is considered in Cartesian coordinates and its latitude and longitude values are taken as X and Y dimensional values separately and filtered using Kalman filter for reducing the error in both the dimensions, for an assumed stationary target scenario. Distinct values for the latitude and longitude pair with an error distribution around zero and a deviation of one were simulated from the localizer output ( 10°04'00" E and 76°21'00" N). The simulated erroneous values were fed to the Kalman filter for refinement of the estimates. By iterating these values and minimizing the covariance, the Kalman filter eventually converges. Measured values for the tracking scenario are also simulated from the localizer output by adding appropriate random functions. It is assumed that the target is moving in a straight line with constant velocity and when the corresponding measured values are loaded, the Kalman filter generates the corrected values.

The algorithm implemented for the tracking of a maneuvering target is illustrated below, which detects the maneuvering of the target and upon detection it reinitializes the Kalman filter. For all scenarios the output figures vivify the effectiveness of the algorithm.

##### 4.1 Algorithm for Tracking of a Maneuvering Target

```

Start
LABEL : Obtain position measurements
        Form state vector containing position and velocity
        Implement KF tracking algorithm on the state vector
        Compute the distance function
        Select a threshold from the chi-distribution table
        If threshold > distance, go to LABEL
    
```

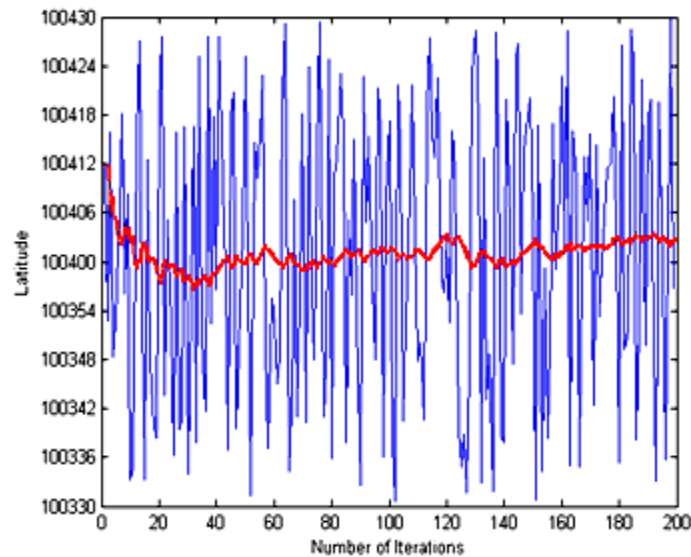
Reset Kalman filter parameters  
Reinitialize the filter from previous iterations  
If surveillance active, go to LABEL  
Stop

Using the chi-square distribution table, the value of the threshold, TH is taken as 10 which is equal to the value of the chi-square corresponding to a probability of 0.99, beyond which the target is detected to be under maneuvering.

## 5. RESULTS AND DISCUSSIONS

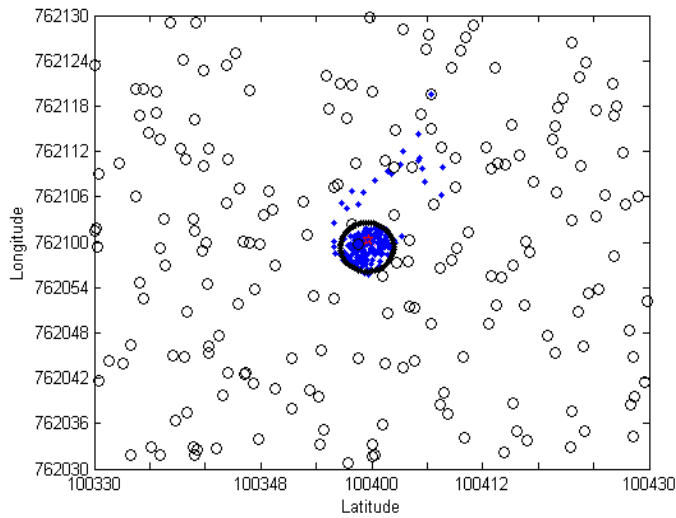
### 5.1 Improving localization Estimates of a Stationary Target

In the case of a stationary target and when simulated with erroneous latitude data, the output of the Kalman filter is depicted in Figure. 1 where the erroneous data is generated by adding randomness to the latitude value  $10^{\circ}04'00''$  from the localizer output. When the Kalman gain and the error covariance converge and remain stable, the output of the Kalman filter is considered to be reliable in the subsequent iterations. It can be seen that after 31st iteration the Kalman filter converges and generates the latitude value very close to the true value. The same algorithm is extended to the erroneous longitude values too, generating the corrected longitude value.



**FIGURE 1:** Kalman Filter output for the latitude data

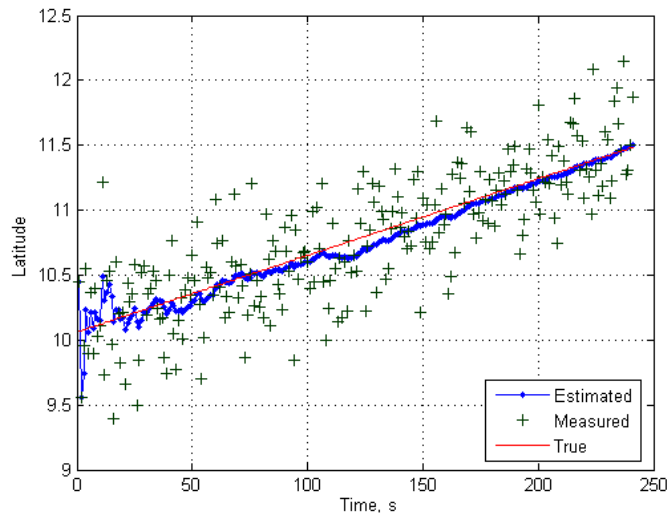
A set of randomly fluctuating positional values indicated by 'o' markings, in Figure. 2, have been used for generating the corrected positional values by the Kalman filter. These values are simulated by adding random error to the assumed values of latitude and longitude data, viz.,  $10^{\circ}04'00''$  and  $76^{\circ}21'00''$  respectively. The estimated Kalman output comprise of the points marked with '.' markings within the circled region in this figure. The filter converges and the estimated output of the Kalman filter is the position with latitude  $10^{\circ}03'59.6''$  and longitude  $76^{\circ}21'0.2''$  which is very close to the true value.



**FIGURE 2:** KF applied to the noisy measurements of a stationary target

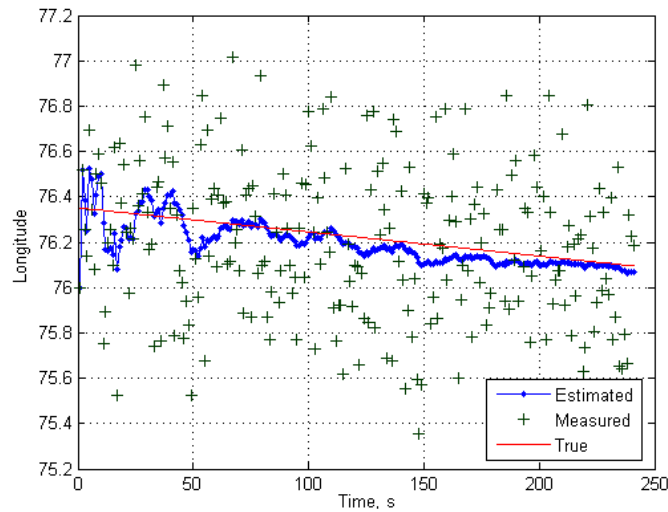
### 5.2 Tracking of a Moving Target

In the second case analyzed herein, it is assumed that the target is moving in a straight line with a constant velocity. The true position of the target, the measured position and the estimated position in latitude and longitude dimensions are charted out in Figure. 3 and Figure. 4. The true positions and the estimated positions are almost close to be distinguished from one another after convergence of the Kalman filter while the '+' marks are the measured positions. The fact that Kalman filter reduces the minimum mean square errors with elapsed time is clearly proven in the given plots. The initial predictions are not accurate as shown, but the filter adapts and converges after few iterations.



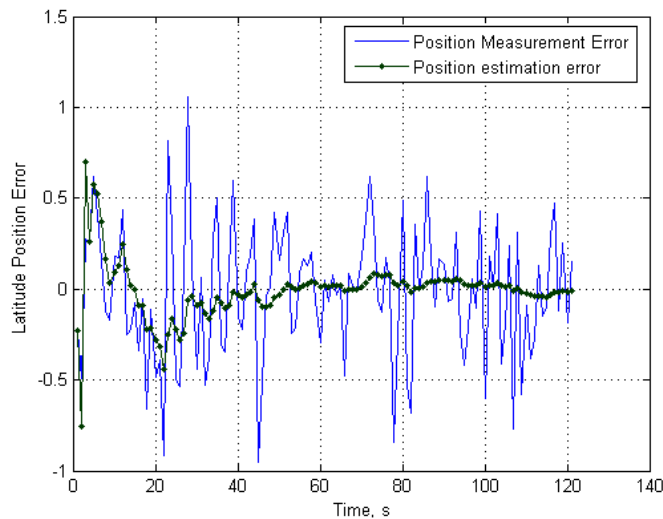
**FIGURE 3:** True, measured and estimated positions in X dimension





**FIGURE 4:** True, measured and estimated positions in Y dimension

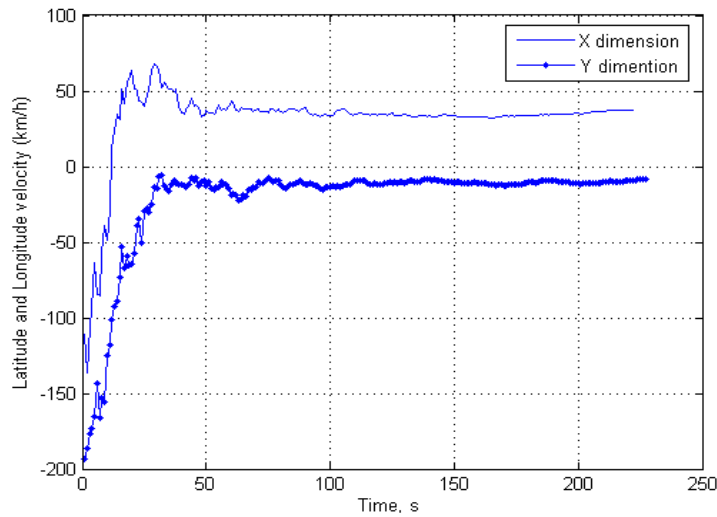
In order to compare the accuracy of the estimates generated by the Kalman filter, Figure. 5 depicts the error values between the true position and the measured position as well as between the true position and the estimated position as provided by the filter. It is clear from this plot that the positions that are estimated or predicted by the Kalman filter are much closer to the true positions almost at every data points. This plot also shows how the Kalman estimates improve over time.



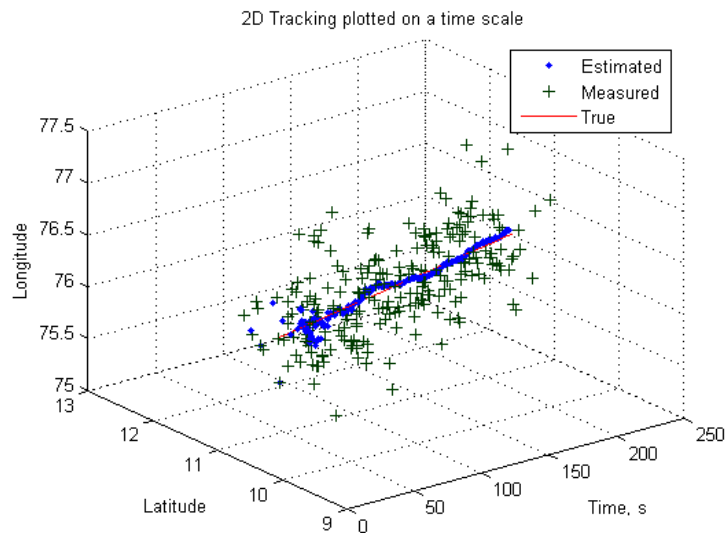
**FIGURE 5:** The residual plot of measurement and estimation with respect to the true values

The relative accuracy of the Kalman output is demonstrated in Figure. 6, which shows the target velocity estimate which is a part of the state variable  $x$ , along with the position estimate. Here the estimated velocity is plotted, as the system does not provide any velocity data measurements. As seen from the plot, the velocity estimates are remarkably more stable in both the dimensions. Though the initial predictions are not accurate, the filter adapts and gets tuned to the variations and limits the error range after a few iterations. The readings reinforce the fact that the Kalman

filter is a powerful approach and reduces the error considerably in both dimensions and also that the accuracy of the filter improves over time, as vivified in the 2D plot shown in Figure. 7.



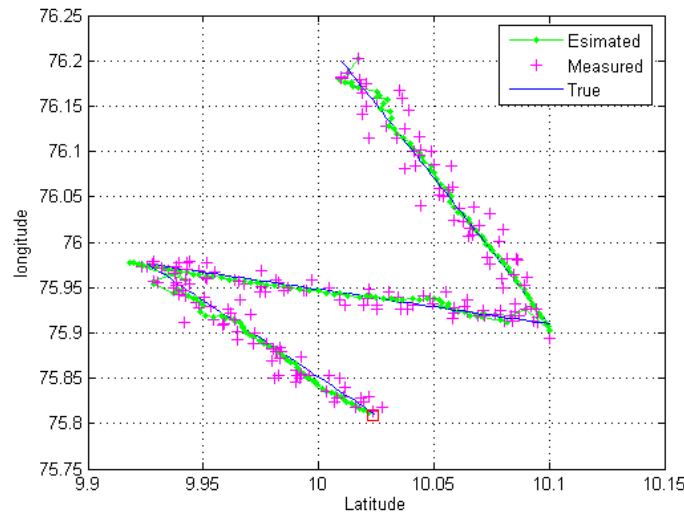
**FIGURE 6:** Velocity variation in X and Y dimensions



**FIGURE 7:** Two dimensional tracking over time

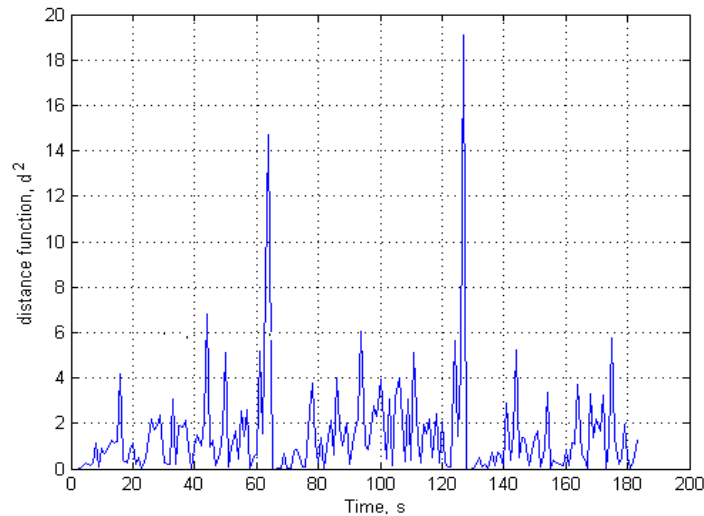
### 5.3 Tracking of a Maneuvering Target

This case assumes the system model to be the same as that of the two dimensional scenario, but with the target presumed to be maneuvering twice. Maneuver of the target is detected by comparing the distance function with the threshold determined by the chi square probability distribution. The threshold is set as 10 in this simulation, which corresponds to a probability of 0.99 in the chi-square distribution function. Once maneuver is recognized, the Kalman filter parameters are reset and the filter is reinitialized using previous measurements.



**FIGURE 8:** True, measured and estimated positions of a maneuvering target

As depicted in Figure. 8, the filter resets upon detecting a maneuver and thus it provides more accurate predictions over time. The position residual plot of the target maneuvering scenario depicted in Figure. 9 shows detection of the target maneuvers as graphical peaks at positions 63 and 128, which closely correlates with the simulated maneuvers at the positions 60 and 125 respectively. The validity of chi-square test relies on the assumption that the process is Gaussian and independent, which is not necessarily valid in practice. Nevertheless chi-square tests are used in these situations because of its simplicity even though it is not necessarily optimal. Also the measurements expressed in Cartesian coordinates are not independent, but the effect of ignoring this fact is negligible in practice [14, 17].



**FIGURE 9:** Position residual graph showing the maneuvering points

## 6. CONCLUSIONS

Implementing Kalman filters to target tracking systems yield reliable results, given that the nature of the system can be modeled suitably. Applying such an adaptive filtering to a simulated stationary and moving system has yielded encouraging results, even when stochastic maneuvers were introduced on the target. Various techniques that were implemented in the filter to circumvent the errors induced due to generic noises and the maneuvering of the target have been

studied. The system can be extended to multiple dimensions of correlated parameters by appending desirable modifications in the system model. More reliable results can be obtained with the incorporation of other efficient techniques like neural networks and fuzzy logics.

## 7. REFERENCES

- [1] W. C. Knight, R. G. Pridham and S. M. Ray, "Digital Signal Processing for Sonar", *Proceedings of IEEE*, 69:1451-1505, 1981
- [2] R. O. Nielsen, *Sonar Signal Processing*, Artech House, (1991)
- [3] R. J. Urick, *Principles of Underwater Sound*, McGraw Hill, (1975)
- [4] G. Welch and G. Bishop, "An Introduction to the Kalman Filter", Department of Computer Science, University of North Carolina, Jul 24, 2006.  
Available: <http://www.cs.unc.edu/~welch/kalman>
- [5] Monson H. Hayes, *Statistical Digital Signal Processing and Modeling*, New York, John Wiley & Sons Inc., (2003)
- [6] C. M. Kwan, and F. L. Lewis, "A Note on Kalman Filtering" , *IEEE Transactions on Education*, 42:225-228,1999
- [7] D. Simon, "Kalman Filtering" , *Embedded Systems Programming*, 14 (6 ):72-79, 2001
- [8] D. Simon and T. Chia, "Kalman Filter with state equality constraints", *IEEE Transactions on Aerospace and Electronic Systems*, 38:128-136, 2002,
- [9] Kalman R.E., "A new approach to linear filtering and prediction", *Journal of Basic Engineering* (ASME), 82:35-45, 1960
- [10] P. S. Maybeck, "Stochastic Models, Estimation and Control", Academic Press, Inc. New York, (1979)
- [11] M. Isabel Ribeiro, "Gaussian Probability Density Functions : Properties and Error Characterization", (2004)
- [12] C. Prabha, Supriya M. H., P. R. Saseendran Pillai, "Improving the Localization Estimates using Kalman filters", Proceedings of International Symposium on Ocean Electronics, India, 2009 and published in IEEE Xplore
- [13] C. Prabha, Supriya M. H., P. R. Saseendran Pillai, "Model Studies on the Localisation of Underwater Targets using Sensor Networks", Proceedings of International Symposium on Ocean Electronics, India, 2007
- [14] Samuel S. Blackman, *Multiple-Target Tracking with Radar Applications*, Norwood: Artech House, (1986).
- [15] Radhakisan S. Baheti, "Efficient Approximation of Kalman Filter for Target Tracking", *IEEE Transactions on Aerospace and Electronic Systems*, 22:8-14, 1986
- [16] Y. T. Chan, J. R. T. Bottomley, "A Kalman Tracker with a Simple input Estimator", *IEEE Transactions on Aerospace and Electronic Systems*,18:235-241, 1982
- [17] X. Rong li and Vesselin p. Jilkov, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models", *IEEE Transactions on Aerospace and Electronic systems*,39:1333-1364, 2003

# Word Recognition in Continuous Speech and Speaker Independent by Means of Recurrent Self-organizing Spiking Neurons

## Tarek Behi

tarekbehi@gmail.com

*Ecole Nationale d'Ingénieurs de Tunis  
Electrical Engineering Department  
Signal, Image and Pattern Recognition Research Unit  
University Tunis El Manar  
Tunis, 1002, Tunisia*

## Najet Arous

Najet.Arous@enit.rnu.tn

*Ecole Nationale d'Ingénieurs de Tunis  
Electrical Engineering Department  
Signal, Image and Pattern Recognition Research Unit  
University Tunis El Manar  
Tunis, 1002, Tunisia*

## Noureddine Ellouze

N.Ellouze@enit.rnu.tn

*Ecole Nationale d'Ingénieurs de Tunis  
Electrical Engineering Department  
Signal, Image and Pattern Recognition Research Unit  
University Tunis El Manar  
Tunis, 1002, Tunisia*

---

## Abstract

Artificial neural networks have been applied successfully in many static systems but present some weaknesses if patterns involve a temporal component. Let's note for example in speech recognition or contextual information, where different of the time interval, is crucial for comprehension. Speech, being a temporal form of sensory input, is a natural candidate for investigating temporal coding in neural networks. It is only through comprehension of the temporal relationship between different sounds which make up a spoken word or sentence that speech becomes intelligible. In fact we present in this paper presents three variants of self-organizing maps (SOM), the Leaky Integrators Neurons (LIN), the Spiking\_SOM (SSOM) and the recurrent Spiking\_SOM (RSSOM) models. The proposed variants is like the basic SOM, however it represents the characteristic to modify the learning function and the choice of the best matching unit (BMU). The case study of the proposed SOM variants is word recognition in continuous speech and speaker independent. The proposed SOM variants show good robustness and high word recognition rates.

**Keywords:** Word Recognition, Kohonen Map, Spiking Neural Networks, Leaky Integrators Neurons, Spiking SOM, Recurrent Spiking SOM.

---

## 1. INTRODUCTION

The majority of artificial neural network models were based on a computational paradigm involving the propagation of continuous variables from one unit to another. A new generation of pulsed neural networks has emerged, which focuses upon the mathematical formalization of the computational properties of biological neurons [1], [2]. The models which communicate through spikes use the timing of these spikes to encode and compute information. In Spiking Neuron Networks (SNNs), the presence and timing of individual spikes is considered as the means of communication and neural computation. This compares with traditional neuron models where analog values are considered, representing the rate at which spikes are fired. A simple spiking neural model can carry out computations over the input spike trains under several different modes [3]. Thus, spiking neurons compute when the input is encoded in temporal patterns, firing rates, firing rates and temporal correlations, and space-rate codes. An essential feature of the spiking neurons is that they can act as

coincidence detectors for the incoming pulses, by detecting if they arrive in almost the same time [4, 5].

Spontaneous speech production is a continuous and dynamic process. This continuity is reflected in the acoustics of speech sounds and, in particular, in the transitions from one speech sound to another [6]. To take account of time in a system of data processing poses two great constraints. First, this system must be able to manage the succession of the various events which must be to treat in a sequential way, it is then a question of sequential treatment. Thus, if the duration of the events is relevant for the task to carry out, the system must be able to treat the temporal structure. However, in the context of the speech recognition, the use of the static networks of neurons is difficult sight the absence of the parameter time in their structure.

In order to classify temporal sequences many technique have been used to model temporal relation in connectionist model [7], [8] like the networks of recurring neurons [9], the temporal self-organizing map [10], [11], [12] and networks of impulse neurons [13], [8] which prove the existence of robust techniques of recognition and classification.

In our model the temporal information is taken into account by using spiking neurons. Spiking neural networks (SNN) have become quite popular recently, due to their biological plausibility. Using spiking neuron models, SNN are able to encode temporal information into both spike timing and spiking rates. The model which realizes the spiking neurons as coincidence detectors encodes the training input information in the connection delays.

The structure of the rest of this paper is as follows: In section 2, we present the self-organizing map and spiking self organizing map. In section 3, we propose the new variant, recurrent spiking self organizing map. In Section 4 we explain the principles of the new variant leaky integrators neurons model. In section 5, we illustrate experimental results of the application of SOM, RSOM, LIN, SSOM and RSSOM models in word recognition of TIMIT speech corpus.

## 2. SPIKING SELF ORGANIZING MAP

Self-organizing in networks is one of the most popular neural network fields [14], [15]. Such networks can learn to detect regularities and correlations in their input and adapt their future responses to that input accordingly [16]. The neurons of competitive networks learn to recognize groups of similar input vectors. Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors.

A self-organizing map learns to categorize input vectors. It also learns the distribution of input vectors. Feature maps allocate more neurons to recognize parts of the input space where many input vectors occur and allocate fewer neurons to parts of the input space where few input vectors occur. Self-organizing maps also learn the topology of their input vectors.

The self-organizing map output represents the result of a vector quantization algorithm that gives a fixed number of references or prototype vectors onto high dimensional data sets in an ordered fashion. A mapping from a high dimensional data space ( $\mathfrak{R}^n$ ) onto a two dimensional lattice of units is thereby defined.

An input vector  $x \in \mathfrak{R}^n$  is compared with all  $m_i$ , in any metric; in practical applications, the smallest of the Euclidian distances is usually used to define the best matching unit (BMU). The BMU is the neuron whose weight vector  $m_i$  is closest to the input vector  $x$  determined by:

$$\|x - m_c\| = \min\{\|x - m_i\|\}, \forall i \in [1..n] \quad (1)$$

Where  $n$  is the number of map units and  $\|x - m_i\|$  is a distance measure between  $x$  and  $m_i$ . After finding the BMU, his weight vector is updated so that the BMU is moved closer to the current input vector. The topological neighbors of the BMU are also updated. This adaptation procedure stretches the BMU and its topological neighbors towards the sample vector. Kohonen update rule for weight vector of the unit  $i$  in the BMU neighborhood is:

$$m_i(t+1) = m_i(t) + \alpha(t) h_{ci}(t) [x(t) - m_i(t)], \forall i \in [1..n] \quad (2)$$

$x(t)$  is the input vector randomly drawn from the input data set at time  $t$ ,  $h_{ci}(t)$  the neighborhood kernel around the winner unit  $c$  and  $\alpha(t)$  the learning rate at time  $t$  [17].

In the context of spiking neuron networks we given a set  $S$  of  $m$ -dimensional input vectors  $s = (s_1, \dots, s_m)$  and a spiking neuron network with  $m$  input neurons and  $n$  output neurons, where each output neuron  $v_j$  receives synaptic feedforward input from each input neuron  $u_i$  with weight  $w_{ij}$  and lateral synaptic input from each output neuron  $v_k$ , with weight  $w_{kj}$ . At every epoch of the learning procedure one sample is chosen and the input neurons are made fire such that they temporally encode input vectors [18], [19].

In the algorithm proposed here, the winner is selected from the subpopulation of units that fire the quickest in one simulation step. After choosing a winner, learning is applied as follows. The afferent weights of a competitive neuron  $i$  are adapted in such a way as to maximize their similarity with the current input pattern  $j$ . A measure of the similarity is the difference between the postsynaptic potential  $s_i$  that encodes the input stimulus and the connection weight  $w_{ij}$ . Furthermore, a spatial and a temporal neighborhood of the winner are created, such that only the neurons inside the  $S$  area and which have fired up until a reference time  $T_{out}$  are subject to learning. The learning rule is adapted from [29] and is given by:

$$\Delta w_{ij} = \eta \frac{T_{out} - t_j}{T_{out}} (s_i - w_{ij}) \quad (3)$$

Where  $T_j$  is the firing time of the  $j$  neuron,  $T_{out}$  is a time out limit, and  $\eta$  is the learning rate.

### 3. RECURRENT SPIKING SELF ORGANIZING MAP

The recurrent Self-Organizing Map (RSOM) [20], [21] as an extension to the Self-Organizing Map (SOM) that allows storing certain information from the past input vectors. The information is stored in the form of difference vectors in the map units. The mapping that is formed during training has the topology preservation characteristic of the SOM. Recurrent SOM differs from the SOM only in its outputs. The outputs of the normal SOM are reset to zero after presenting each input pattern and selecting best matching unit with the typical winner takes all strategy. Hence the map is sensitive only to the last input pattern. In the RSOM the sharp outputs are replaced with leaky integrator outputs, which once activated gradually lose their activity. The modeling of the outputs in RSOM is close to the behavior of natural neurons, which retain an electrical potential on their membranes with decay. The use of impulsional neuron makes it possible to improve the taking into account of the temporal parameters.

The model presented in this part has the same principle that spiking self organizing map except that the choice of the BMU is defined by a difference vector in each unit of the map. The difference vector is included in the recurring bond. Thus, the memory stores a linear sum of the preceding vectors.

The difference vector  $l(n)$  in each unit of the map is defined there according to this equation:

$$y_i(t) = (1 - \alpha)y_i(t-1) + \alpha(x(t) - m_i(t)) \quad (4)$$

Where  $y_i(n)$  is the leaked difference vector in unit  $i$ ,  $0 < \alpha \leq 1$  is the leaking coefficient.  $x(t)$  is the input vector and  $m_i(t)$  is the weight vector of the unit  $i$ .

### 4. LEAKY INTEGRATOR NEURONS

We present neural networks consisting of leaky integrator units as a universal paradigm for neural and cognitive modeling. In contrast to standard recurrent neural networks, leak integrator units are described by ordinary differential equations living in continuous time. We present an algorithm to train the temporal behavior of leaky integrator networks.

In order to use leaky integrator units to create network models for simulation experiments, a learning rule that works in continuous time is needed. The following formulation is motivated by [22] and describes how a backpropagation algorithm for leaky integrator units can be derived.

In this approach, the state of each neuron ( $i$ ) is represented by a membrane potential  $P_i(t)$ , which is a function of the input  $I_i(t)$  which measures the degree of matching between the neuron's weight vector and the current input vector.

The differential equation of a membrane potential is:

$$\frac{dP_i}{dt} = -\eta P_i(t) + I_i(t) \quad (5)$$

Where  $\eta < 0$ .

Particularly, the discrete version of the equation (5), often written as:

$$P_i(t) = \lambda P_i(t-1) + I_i(t) \quad (6)$$

LIN memorise the last activation of each neuron  $i$  by means of a Leaky Integrators potential noted  $a_i(t)$  [23], [24], [25]:

$$a_i(t) = \lambda a_i(t-1) - \frac{1}{2} P x(t) - w_i(t) P^2 \quad (7)$$

Where  $\lambda$  is a depth memory constant ( $0 \leq \lambda \leq 1$ ),  $x(t)$  is the input vector, and  $w_i(t)$  is the weight vector of neuron  $i$ . Comparing equations (6) and (7), we find that  $l_i(t) = -(\frac{1}{2}) \|x(t) - w_i(t)\|^2$ .

## 5. RESULTS AND DISCUSSIONS

We have implemented a variant of Kohonen network for continuous speech recognition. The realized system is composed of three main components [26], [27]: a pre-processor sounds and producing mel cepstrum vectors. The sound input space is composed by 12 mel cepstrum coefficients each 16 ms frame. 9 frames are selected at the middle of each word to generate data vectors. The second component is a competitive learning module. The third component is a word recognition module.

We used the DARPA TIMIT speech corpus for all experiments.

The TIMIT corpus is considered as a reference database [28]. Its broad diffusion in the international community allows an objective evaluation and shares performances of the developed systems.

The TIMIT database contains the recordings of 630 American speakers, divided into 8 regional dialects of the American English ('dr1' to 'dr8') and pronouncing each one 10 sentences. These sentences come from 3 corpus:

- Two sentences of calibration, pronounced by all the speakers are used to illustrate the variations of the dialects ('sa1' and 'sa2').
- Five sentences are taken randomly among 450 phonetically balanced and compact sentences conceived with MIT (identified 'sx3' with 'sx452') [29]. Each sentence is pronounced by 7 different speakers.
- Three sentences are selected to maximize the acoustic contexts, each sentence is marked only one time, On the hole 1890 different sentences for the 630 speakers (identified 'si453' to 'si2343') phonetically various selected with TI.

The total vocabulary of the base is 6300 sentences. The 630 speakers of the base (438 men and 192 women) are divided between the whole of training (462 speakers including 326 men and 136 women) and the whole of test (168 speakers including 112 men and 56 women). For each sentence, we have the English text, the sampled signal with 16 KHZ with a resolution of 16 bits, segmentation in words and phonemic classification in 61 classes.

In our experiments, we have used the New England dialect region (DR1) composed of 24 male and 14 female. The corpus contains 7380 word units for training. Each word unit is represented by 9 frames selected at the middle of each word to generate data vectors. Training has been made on words for ten sentences of TIMIT database. The sentences and words can be found in table 1.



Sentences	Words
SA1	{'she'} {'had'} {'your'} {'dark'} {'suit'} {'in'} {'greasy'} {'wash'} {'water'} {'all'} {'year'}
SA2	{'don't'} {'ask'} {'me'} {'to'} {'carry'} {'an'} {'oily'} {'rag'} {'like'} {'that'}
SX56	{'academic'} {'aptitude'} {'guarantees'} {'your'} {'diploma'}
SI1377	{'as'} {'these'} {'maladies'} {'overlap'} {'so'} {'must'} {'the'} {'cure'}
SX395	{'i'} {'took'} {'her'} {'word'} {'for'} {'it'} {'but'} {'is'} {'she'} {'really'} {'going'} {'with'} {'you'}
SI921	{'differences'} {'were'} {'related'} {'to'} {'social'} {'economic'} {'and'} {'educational'} {'backgrounds'}
SI1027	{'even'} {'then'} {'if'} {'she'} {'took'} {'one'} {'step'} {'forward'} {'he'} {'could'} {'catch'} {'her'}
SX159	{'the'} {'government'} {'sought'} {'authorization'} {'of'} {'his'} {'citizenship'}
SX117	{'the'} {'mango'} {'and'} {'the'} {'papaya'} {'are'} {'in'} {'a'} {'bowl'}
SI1244	{'the'} {'sculptor'} {'looked'} {'at'} {'him'} {'bugeyed'} {'and'} {'amazed'} {'angry'}

**TABLE 1:** List of words of each sentence

Table 2 shows the number of samples of training data set of TIMIT speech corpus and the size of map for each sentence.

Sentences	Number of samples of training data set	Size of map
SA1	3735	20*15
SA2	3028	19*15
SX56	42	7*5
SI1377	72	7*6
SX395	109	9*6
SI921	81	8*6
SI1027	108	9*6
SX159	59	8*5
SX117	66	8*5
SI1244	80	9*5

**TABLE 2:** Number of samples of training data set of TIMIT speech corpus and the size of map

According to table 3, RSSOM provides the best classification rate 73.70%. With RSSOM we obtained an improvement of the classification rate in order to 16 % in comparison with SOM.

It is also noticed that for the sentence 'SA1', the three variants (SSOM, RSSOM and LIN) have capacities of roughly similar recognition.

Word	SOM	RSOM	SSOM	RSSOM	LIN
she	80.11	82.16	84.21	84.21	76.02
had	73.97	83.62	87.13	88.30	86.25
your	36.47	42.85	52.88	54.71	58.53
dark	65.78	82.74	84.50	89.47	90.35
suit	60.52	75.73	80.40	79.82	71.34
in	74.08	76.52	85.06	81.09	74.08
greasy	31.28	51.16	54.97	50.87	50.87
wash	46.78	65.49	55.55	61.98	71.05
water	37.71	45.02	48.24	60.52	67.54
all	74.26	73.68	80.11	79.23	78.94
year	55.26	77.77	77.77	80.11	76.31
<b>Average</b>	<b>57.85</b>	<b>68.86</b>	<b>71.91</b>	<b>73.70</b>	<b>72.87</b>

**TABLE 3:** Sentence SA1 recognition rates

Table 4 shows that RSSOM provide the best recognition accuracy in order to 69.11%. LIN provides best rate for the word /rag/ in order to 83.49%

Word	SOM	RSOM	SSOM	RSSOM	LIN
don't	67.30	68.88	80.63	75.87	80.63
ask	46.98	51.11	57.77	60.00	56.82
me	61.05	56.84	74.03	62.10	63.85
to	44.63	57.04	62.41	76.84	65.77
carry	57.14	54.28	65.07	69.20	66.98
an	47.95	61.56	64.28	68.02	75.17
oily	58.41	67.93	68.88	76.19	75.55
rag	62.22	73.96	80.00	82.22	83.49
like	19.36	33.33	50.79	38.73	33.33
that	72.06	71.42	78.41	77.46	77.14
<b>Average</b>	<b>52.30</b>	<b>58.79</b>	<b>68.20</b>	<b>69.11</b>	<b>67.71</b>

**TABLE 4:** Sentence SA2 recognition rates

From table 5 RSSOM, LIN and SSOM provide best classification accuracy in comparison with SOM. The variant LIN reaches good classification rates in order to 97.22%. However, this model reaches good recognition rates (in the range of 90 and 100%).

We also note that for the words /step/ and /her/ the SOM provide recognition rates in the range of 10% and 30%, on the other hand the models LIN and RSSOM provide a higher value of recognition rate in order to 100%.

**TABLE 5:** Sentence S11027 recognition rates

Word	SOM	RSOM	SSOM	RSSOM	LIN
even	100	100	100	100	100
then	100	100	100	100	100
if	88.88	77.77	100	100	100
she	66.66	88.88	77.77	77.77	88.88
took	88.88	100	100	88.88	88.88
one	88.88	100	100	100	88.88
step	33.33	100	88.88	100	100
forward	100	100	100	100	100
he	88.88	88.88	100	88.88	100
could	100	88.88	100	100	100
catch	100	100	100	100	100
her	11.11	77.7	88.88	100	100
<b>Average</b>	<b>80.55</b>	<b>93.51</b>	<b>96.29</b>	<b>96.29</b>	<b>97.22</b>

From table 6 with RSSOM we obtained an improvement of the classification rate of 14 % in comparison with SOM and 21% in comparison with RSOM. RSSOM reaches good classification rates in order to 92.85% in training set.

For the word /guarantees/ RSOM has the low recognition rate in order to 10%, but with the recurrent spiking SOM (RSSOM) model we obtained a recognition rate in order to 70%, this result prove the stability and performance of the variant RSOM.

Word	SOM	RSOM	SSOM	RSSOM	LIN
academic	66.66	77.77	100	100	100
aptitude	100	100	88.88	100	100
guarantees	33.33	11.11	44.44	66.66	55.55
your	100	66.66	100	100	100
diploma	100	100	100	100	100
<b>Average</b>	<b>78.57</b>	<b>71.42</b>	<b>85.71</b>	<b>92.85</b>	<b>90.47</b>

**TABLE 6:** Sentence SX56 recognition rates

According to table 7, Leaky integrators neurons (LIN) and recurrent spiking SOM (RSSOM) provide best classification rate in order to 100% in training set.

SOM model gives the weak result for the word /so/ in order to 44%, on the other hand, all the other models provide a higher value of recognition rate in order to 100%.

Word	SOM	RSOM	SSOM	RSSOM	LIN
as	100	100	100	100	100
these	100	100	100	100	100
maladies	88.88	77.77	100	100	100
overlap	100	100	100	100	100
so	44.44	100	88.88	100	100
must	100	88.88	77.77	100	100
the	100	88.88	100	100	100
cure	100	100	100	100	100
<b>Average</b>	<b>91.66</b>	<b>94.44</b>	<b>95.83</b>	<b>100</b>	<b>100</b>

**TABLE 7:** Sentence S11377 recognition rates

According to table 8, the variants LIN and RSSOM provide best classification rate in order to 100% in training set.

Word	SOM	RSOM	SSOM	RSSOM	LIN
the	100	62.50	100	100	100
mango	100	100	100	100	100
and	66.66	44.44	100	100	100
papaya	100	88.88	100	100	100
are	100	77.77	100	100	100
in	88.88	100	100	100	100
a	100	100	75.00	100	100
bowl	100	100	100	100	100
<b>Average</b>	<b>93.93</b>	<b>84.84</b>	<b>98.48</b>	<b>100</b>	<b>100</b>

**TABLE 8:** Sentence SX117 recognition rates

According to table 9, Leaky Integrators Neurons (LIN) provides the best classification rate in order to 96.25%. With LIN we obtained an improvement of the classification rate in order to 19 % in comparison with SOM and RSOM in training set. A higher value of recognition rate for the words /looked/, /bugeyed/, /and/ and /him/ means a better performance of variant SOM.

Word	SOM	RSOM	SSOM	RSSOM	LIN
the	87.50	87.50	87.50	100	100
sculptor	100	55.55	100	100	100
looked	66.66	77.77	88.88	88.88	100
at	88.88	88.88	100	88.88	100
him	77.77	55.55	100	100	100
bugeyed	44.44	77.77	100	100	100
and	44.44	66.66	66.66	88.88	66.66
amazed	88.88	100	88.88	88.88	100
angry	100	100	100	100	100
<b>Average</b>	<b>77.50</b>	<b>78.75</b>	<b>92.50</b>	<b>95.00</b>	<b>96.25</b>

**TABLE 9:** Sentence S11244 recognition rates

According to table 10, LIN and RSSOM models provide best classification rate in order to 97.24% in training set. With RSSOM and LIN we obtained an improvement of the classification rate in order to 30 % in comparison with SOM.

With SOM model we can't recognize some words like /for/ and /is/, recognition rates (in the range of 0 and 20%), on the other hand the models LIN and RSSOM provide a higher value of recognition rate (in the range of 90 and 100%)

Word	SOM	RSOM	SSOM	RSSOM	LIN
i	100	77.77	100	100	100
took	88.88	77.77	100	100	100
her	77.77	88.88	100	100	100
word	77.77	77.77	100	100	100
for	0.00	77.77	55.55	100	88.88
it	60.00	100	80.00	100	100
but	77.77	77.77	88.88	100	100
is	20.00	100	100	80.00	80.00
she	100	100	100	100	100
really	44.44	88.88	77.77	88.88	100
going	55.55	100	100	100	100
with	55.55	88.88	88.88	88.88	88.88
you	77.77	100	88.88	100	100
<b>Average</b>	<b>66.05</b>	<b>87.15</b>	<b>90.82</b>	<b>97.24</b>	<b>97.24</b>

**TABLE 10** : Sentence SX395 recognition rates

According to table 11, the variant LIN provide best classification rate in order to 100% in training set. Table 11 shows that for words / related / and / backgrounds / classic SOM and RSOM are not able to recognize these classes (small recognition rates). However, SSOM, RSSOM and LIN reach good recognition rates (in the range of 90 and 100%).

Word	SOM	RSOM	SSOM	RSSOM	LIN
differences	100	100	100	100	100
were	100	100	100	100	100
related	55.55	44.44	88.88	100	100
to	100	100	100	100	100
social	100	100	100	100	100
economic	88.88	88.88	88.88	100	100
and	100	100	88.88	77.77	100
educational	100	100	88.88	100	100
backgrounds	66.66	100	100	100	100
<b>Average</b>	<b>90.12</b>	<b>92.59</b>	<b>95.06</b>	<b>97.53</b>	<b>100</b>

**TABLE 11** : Sentence S1921 recognition rates

Table 12 shows that LIN provides the best recognition accuracy in order to 98.30%. With LIN we obtained an improvement of the classification rate in order to 14 % in comparison with SOM. It is also noticed that for the sentence 'SX159', the three variants (SSOM, RSSOM and LIN) have capacities of roughly similar recognition.

Word	SOM	RSOM	SSOM	RSSOM	LIN
the	80.00	80.00	100	86.92	100
government	100	100	100	100	100
sought	100	100	88.88	100	100
authorization	66.66	88.88	100	100	100
of	88.88	100	100	100	100
his	55.55	44.44	66.66	77.77	88.88
citizenship	100	100	100	100	100
<b>Average</b>	<b>84.74</b>	<b>88.13</b>	<b>93.22</b>	<b>96.61</b>	<b>98.30</b>

**TABLE 12** : Sentence SX159 recognition rates

## 6. CONCLUSION

In this paper, we have proposed new variants of self organizing neural network algorithm in the unsupervised learning category, and we are interested in word recognition using sentences from TIMIT databases by means of new SOM variants with impulse neurons.

The use of impulse neurons in the SOM makes it possible to establish temporal associations between the consecutive models in a temporal order through the impulses produced according to the entry and makes it possible to improve the taking into account of the temporal parameters in the recurring SOM.

In this paper, we have presented three SOM variants, Leaky Integrators Neurons model (LIN) which consider the temporal order between the successive samples by using a mechanism called Leaky Integrators, In this approach, the state of each neuron is performed by a membrane potential which is function of the input, this potential measure the adaptation degree between the neuron weight vector and the current input vector. Then the Spiking\_SOM model (SSOM), based on the timing or the order of single spike events, the coding which represents information is through the differences in the firing times of different neurons and the recurrent Spiking\_SOM (RSSOM) has the same principle that spiking self organizing map except that the choice of the BMU is defined by a difference vector in each unit of the map.

The proposed SOM variants provide best classification rates in comparison with the basic SOM and RSOM models. The RSSOM and LIN provide best classification rates in order to 100%.

As a future work, we suggest proposing other SOM variant, the Growing Hierarchical Self-Organizing map (GHSOM) using spiking neurons. GHSOM is a network of neurons whose architecture combines two principal extensions of SOM model, the dynamic growth and the tree structure in order to reduce the complexity of the task of classification and improve classification rates.

## 7. REFERENCES

- [1] W. Maass, M. Schmitt. "On the complexity of learning for a spiking neuron". In COLT'97, Conf. on Computational Learning Theory, ACM Press, 1997. pp. 54–61.
- [2] W. Gerstner. "What's different with spiking neurons". In Henk Mastebroek and Hans Vos, editors, *Plausible Neural Networks for Biological Modelling*, Kluwer Academic Publishers;2001. pp. 23–48.
- [3] W. Maass, C. M Bishop. "Pulsed Neural Networks". MIT Press, 1999.
- [4] R. Kempter, W. Gerstner. J. L VanHemmen and H. Wagner. "Extracting oscillations: Neuronal coincidence detection with noisy periodic spike input". *Neural Comput.*, vol. 10, pp. 1987-2017, 1998.
- [5] W. R Softky, C. Koch. "The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs". *J. Neurosci.* Vol. 13, pp. 334–350, 1993.
- [6] F. Santiago , G. Alex, S. Jurgen. "Phoneme recognition with BLSIM-CIC". 2008.
- [7] S. Durand. "Réseaux neuromimétiques spatio-temporels pour l'organisation des sens. Application à la parole. Dans Actes Rencontres Nationales des Jeunes chercheurs en Intelligence Artificielle. Marseille, 1994.
- [8] S. Durand. "TOM, une architecture connexionniste de traitement de séquence. Application à la reconnaissance de la parole". PhD thesis Université Henri Poincaré, Nancy I ; 1995.
- [9] P. Danilo, Mandic, A. Jonathon. "Chambers, Recurrent Neural Networks for Prediction", John Wiley and Sons Ltd, 2001.
- [10] G Vaucher. "Un modèle de neurone artificiel conçu pour l'apprentissage non supervise de séquences d'événements asynchrones". In *Revue VALGO*, ISSN 1243-4825, Vol 1, pp. 66–107, 1993.

- [11] T. Behi, N. Arous. "Modèle auto-organisateur à composante temporelle pour la reconnaissance de la parole continue". Huitième journée scientifiques des jeunes chercheurs en génie électrique et informatique, GEI2008, Sousse-Tunisie, 2008.
- [12] T. Behi, N. Arous. "Modèles auto-organisateur à apprentissage spatio-temporels Evaluation dans le domaine de la classification phonémique". Cinquième conférence internationale JTEA2008, Hammamet-Tunisie , 2008.
- [13] R. Brette. "Modèles Impulsionnels de Réseaux de Neurones Biologiques". PhD Thesis, University of Cerveau-Cognition – Comporteme, 2003.
- [14] T. Kohonen. "Self\_Organized Formation of Topologically Correct Feature Maps". Biological Cybernetics. Vol.43, pp. 59-69,1982.
- [15] N. Arous. "Hybridation des Cartes de Kohonen par les algorithmes génétiques pour la classification phonémique. PhD Thesis Ecole Nationale d'ingénieurs de Tunis , 2003.
- [16] S. Haykin. "Neural Network A Comprehensive Foundation", Prentice Hall Upper Saddle River, New Jersey, 1999.
- [17] T. Kohonen. "Self-organizing map", third edition, Springer, 2003.
- [18] W. Maass. "Computing with spiking neurons".In Maass, W. and Bishop, C. M., editors, Pulsed Neural Networks, chapter 2, MIT-Press., pp. 55-85 (1998)
- [19] W. Maass, C. M. Bishop. "Pulsed Neural Networks". The MIT Press, 1st edition, Cambridge, 1998.
- [20] M. Varsta, J. Heikkonen and R; Milan. "A recurrent self-organizing map for temporal sequence processing". Proc. Int. Conf. on Artificial Neural Networks (ICANN'97), Lausanne, Switzerland, 1997.
- [21] T. Koskela, M. Varsta, J. Heikkonen, K. Kaski. "Time Series prediction using recurrent SOM with local linear models". International Journal of Knowledge-based Intelligent Engineering Systems,2(1), : 60-68, 1998.
- [22] J.G. Taylor. "Temporal patterns and leaky integrator neurons". Proc. Int. Conference. Neural Networks (ICNN90), Paris, 1990.
- [23] T. koskela, M. varsta. "Recurrent SOM with local linear models in time series prediction", PhD Thesis, University of Helsinki university of technologie-labo of computational engineering-Finland, April 1998.
- [24] M. Varsta. "Temporal sequence processing using recurrent SOM", PhD Thesis, University of Helsinki university of technologie-labo of computational engineering-Finland, 1998.
- [25] T. Voegtlin. "Réseaux de neurones et autoréférence", PhD Thesis, University of lumière lyon II, 2004.
- [26] N. Arous, N. Ellouze. "Phoneme classification accuracy improvements by means of new variants of unsupervised learning neural networks", 6th World Multiconference on Systematics, Cybernetics and Informatics, Floride, USA, 2002.
- [27] N. Arous, N. Ellouze. "Cooperative supervised and unsupervised learning algorithm for phoneme recognition in continuous speech and speaker-independent context", Elsevier Science, Neurocomputing, Special Issue on Neural Pattern Recognition, vol. 51, pp. 225 – 235, 2003.
- [28] J. Garofalo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren and V. Zue. "TIMIT acoustic-phonetic continuous speech corpus", Linguistic Data Consort, 2005.

- [29] V. Zue, S. Seneff, J. Glass. "Speech database development at MIT, TIMIT, and beyond", *Speech Commun*, vol. 9, pp. 351–356, 1990.



## INSTRUCTIONS TO CONTRIBUTORS

The *International Journal of Signal Processing (SPIJ)* lays emphasis on all aspects of the theory and practice of signal processing (analogue and digital) in new and emerging technologies. It features original research work, review articles, and accounts of practical developments. It is intended for a rapid dissemination of knowledge and experience to engineers and scientists working in the research, development, practical application or design and analysis of signal processing, algorithms and architecture performance analysis (including measurement, modeling, and simulation) of signal processing systems.

As SPIJ is directed as much at the practicing engineer as at the academic researcher, we encourage practicing electronic, electrical, mechanical, systems, sensor, instrumentation, chemical engineers, researchers in advanced control systems and signal processing, applied mathematicians, computer scientists among others, to express their views and ideas on the current trends, challenges, implementation problems and state of the art technologies.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for SPIJ.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, SPIJ appears in more focused issues. Besides normal publications, SPIJ intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

### **SPIJ LIST OF TOPICS**

The realm of Signal Processing: An International Journal (SPIJ) extends, but not limited, to the following:

- Biomedical Signal Processing
- Communication Signal Processing
- Detection and Estimation
- Earth Resources Signal Processing
- Industrial Applications
- Optical Signal Processing
- Radar Signal Processing
- Signal Filtering
- Signal Processing Technology
- Software Developments
- Spectral Analysis
- Stochastic Processes
- Acoustic and Vibration Signal Processing
- Data Processing
- Digital Signal Processing
- Geophysical and Astrophysical Signal Processing
- Multi-dimensional Signal Processing
- Pattern Recognition
- Remote Sensing
- Signal Processing Systems
- Signal Theory
- Sonar Signal Processing
- Speech Processing

**CALL FOR PAPERS**

---

**Volume: 6 - Issue: 2 - April 2012**

**i. Paper Submission:** January 31, 2012

**ii. Author Notification:** March 15, 2012

**iii. Issue Publication:** April 2012

## **CONTACT INFORMATION**

### **Computer Science Journals Sdn Bhd**

B-5-8 Plaza Mont Kiara, Mont Kiara  
50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6207 1607  
006 03 2782 6991

Fax: 006 03 6207 1697

Email: [cscpress@cscjournals.org](mailto:cscpress@cscjournals.org)

CSC PUBLISHERS © 2011  
COMPUTER SCIENCE JOURNALS SDN BHD  
M-3-19, PLAZA DAMAS  
SRI HARTAMAS  
50480, KUALA LUMPUR  
MALAYSIA

PHONE: 006 03 6207 1607  
006 03 2782 6991

FAX: 006 03 6207 1697  
EMAIL: [cscpress@cscjournals.org](mailto:cscpress@cscjournals.org)