RADAR PULSE REPETITION INTERVAL TRACKING WITH KALMAN FILTER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SONER AVCU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Kerim DEMİRBAŞ
Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU	(METU, EE)	
Prof. Dr. Kerim DEMİRBAŞ	(METU, EE)	
Assoc. Prof. Dr. Çağatay CANDAN	(METU, EE)	
Assoc. Prof. Dr. Fahrettin ARSLAN	(Ankara Unv., STAT)	
İrfan OKŞAR (M.S.)	(ASELSAN)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Soner AVCU

Signature :

ABSTRACT

RADAR PULSE REPETITION INTERVAL TRACKING
WITH KALMAN FILTER

Avcu, Soner
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Kerim Demirbaş

September 2006, 94 pages

In this thesis, the radar pulse repetition interval (PRI) tracking with Kalman Filter problem is investigated. The most common types of PRIs are constant PRI, step (jittered) PRI, staggered PRI, sinusoidally modulated PRI. This thesis considers the step (this type of PRI agility is called as constant PRI when the jitter on PRI values is eliminated) and staggered PRI cases. Different algorithms have been developed for tracking step and staggered PRIs cases. Some useful simplifications are obtained in the algorithm developed for step PRI sequence. Two different algorithms robust to the effects of missing pulses obtained for staggered PRI sequence are compared according to estimation performances. Both algorithms have two parts: detection of the period part and a Kalman filter model. The advantages and disadvantages of these algorithms are presented. Simulations are implemented in MATLAB.

Keywords: Radar Pulse Repetition Interval, Kalman Filter, Discrete Fourier Transform, Performance Analysis

ÖZ

KALMAN SÜZGECİ İLE
RADAR DARBESİ TEKRAR ARALIĞI İZLEME

Avcu, Soner
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Kerim Demirbaş

Eylül 2006, 94 sayfa

Bu tezde, Kalman süzgeci ile radar darbesi tekrar aralığı (PRI) izleme problemi incelendi. En yaygın PRI çeşitleri sabit, kademeli, periyodik değişken değerli (staggered), sinüsoidal modüleli PRI'dır. Bu çalışma kademeli (PRI değerleri üzerindeki gürültü kaldırıldığı zaman Sabit PRI olarak adlandırılır) ve periyodik değişken değerli (staggered) PRI durumlarını dikkate almaktadır. Kademeli ve periyodik değişken değerli PRI'ları izlemek için farklı algoritmalar geliştirilmiştir. Kademeli PRI dizileri için oluşturulan algoritmada önemli birtakım sadeleştirmeler elde edilmiştir. Periyodik değişken değerli PRI dizisi için oluşturulan kayıp darbelerin etkilerine karşı korumalı olan iki farklı algoritma, tahmin performanslarına göre karşılaştırılmıştır. Her iki algoritma, periyot saptama ve Kalman süzgeci modeli olmak üzere iki bölümden oluşmaktadır. Algoritmaların avantajlı ve dezavantajlı yönleri gösterilmiştir. Simülasyonlar MATLAB ortamında gerçekleştirilmiştir.

Anahtar Kelimeler: Radar Darbesi Tekrar Aralığı, Kalman Süzgeci, Ayrık Fourier Dönüşümü, Performans analizi

ACKNOWLEDGMENTS

I would like to express my thanks and gratitude to Prof. Dr. Kerim Demirbaş for his valuable supervision, advice and criticism throughout the development and improvement of this thesis.

I would also like to extend my special appreciation to my parents for their encouragement they have given me not only throughout my thesis but also throughout my life.

Lastly, I thank Selvi Kaya with deepest appreciation for her incessant encouragement and motivation.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The problem of estimation and tracking the radar pulse repetition interval (PRI) has many civilian and military applications which include neurobiological signal processing, the radar intercept problem, etc... In this work, the radar intercept problem meaningly the PRI tracking is considered. A PRI tracking algorithm is needed as the basis for a deceptive countermeasures technique. Filtering and estimation are two important tools in this area. Whenever the state of a system must be estimated from received jittered information, some types of estimators are used for accurate estimation. When the system and observation models are linear, the minimum mean squared error (MMSE) estimate can be implemented using Kalman filter. What is a “Kalman Filter”? The Kalman filter is an estimator for the linear-quadratic problem, which is the problem of estimating the instantaneous “state” of a linear dynamic system perturbed by white noise [7]. The Kalman filter is one of the most used methods for tracking due to its simplicity and optimality.

In this thesis, the radar pulse repetition interval (PRI) tracking with Kalman Filter problem is investigated. The most common types of PRI agilities are step (jittered) PRI, constant PRI (constant PRI is a special case of step PRI, step PRI is called as constant PRI when the jitter on PRI values is eliminated), staggered PRI and sinusoidally modulated PRI [4]. This thesis considers the step and staggered PRI cases. One algorithm has been developed for step PRI case and two different algorithms have been developed for staggered PRI case. One of the algorithms developed for staggered PRI case uses discrete Fourier transform of the staggered PRI sequence but the other one does not use; it takes directly the PRI values to form the state vector. Estimation performances of the algorithms developed for staggered PRI case are investigated. Both algorithms are developed in MATLAB environment and some simulations are performed in order to evaluate PRI estimation performances of the methods used in algorithms.

In Chapter 2, Kalman filter is explained. Since a staggered PRI sequence can be viewed as a discrete time series, detailed information is given about Discrete Kalman filter. One cycle in the state estimation of a linear system is explained by giving time update (prediction) and measurement update equations. Also, the meanings of the parameters used in Kalman filter are given.

In Chapter 3, one algorithm has been developed for step PRI case and two different algorithms have been developed for staggered PRI case. In the algorithm developed for step PRI sequence, some useful simplifications are obtained. So, the computational complexity of the matrices is reduced. Also the use of validation region (gate) [18, 19] in the presence of missing pulses is explained. On the other hand, two different algorithms developed for staggered PRI sequence. These algorithms have two parts: detection of the period part and Kalman filter predictor part. Detection of the period part is used to determine the period of staggered PRI sequence. Since a staggered PRI sequence can be viewed as a discrete time series and a staggered sequence is periodic, a staggered PRI sequence can be expressed as a linear combination of sine and cosine terms plus a constant (Fourier representation theorem [15]). So one of the algorithms uses discrete Fourier transform of the staggered PRI sequence but the other one does not use. Robustness to the effect of missing pulses is also considered for both algorithms.

In Chapter 4, the simulation results for the comparison of the two algorithms developed for staggered PRI sequence are given. The PRI estimation performances for both algorithms are presented.

In Chapter 5, the conclusions are given in which the advantages and disadvantages of both algorithms are discussed by considering the simulation results.

In Appendix, the computer programs written for the staggered PRI case in MATLAB environment are given

CHAPTER 2

KALMAN FILTER

The Kalman filter was developed by Rudolph Emil Kalman, although Peter Swerling developed a very similar algorithm in 1958. The filter is named after Kalman because he published his results in a more prestigious journal and his work was more general and complete [3].

The Kalman filter, which is based on the use of state-space techniques and recursive algorithms, revolutionized the field of estimation. Since that time, Kalman filter has been applied in the fields of aerospace, military, nuclear plant, economics, radar, sonar, biomedical signal processing, etc [8]. Kalman filter became one of the greatest discoveries in the history of statistical estimation theory. The lifelines of important names in the history of estimation theory are shown in Figure 1.

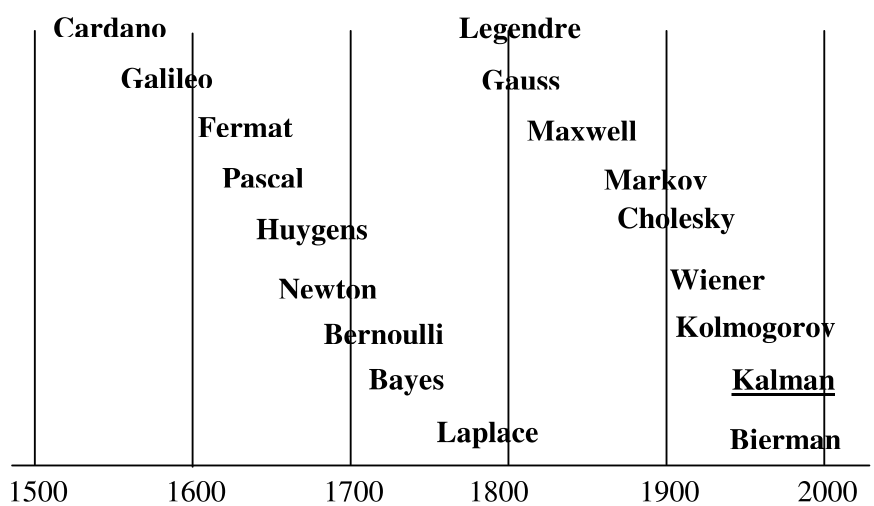

Figure 1. Founder's Lifelines of the Estimation Theory [7]

The Kalman filter is a multiple-input, multiple-output digital filter that can optimally estimate, in real time, the states of a system based on its noisy outputs [1]. Mathematically, the Kalman filter estimates the states of a linear system. It is very attractive in theory and practice because it minimizes the variance of estimation error effectively. It does not require storing all previous data in memory, because it is a recursive filter which makes it easier to implement Kalman filter. The Kalman filter is just a computer algorithm for processing discrete measurements (the input) into optimal estimates (the output) [6].

In order to use a Kalman filter to remove noise from a signal, the process that we are measuring must be able to be described by a linear system [3]. The Kalman filter, which assumes linear systems, has found its greatest application to nonlinear systems. It is generally used in these problems by assuming knowledge of an approximate solution (as Gauss proposed) and by describing the deviations from the reference by linear equations. The approximate linear model that is obtained forms the basis for the Kalman filter utilization [9]. Many physical processes, such as a satellite orbiting the earth, a motor shaft driven by winding currents, a vehicle driving along a road or a sinusoidal radio-frequency carrier signal, can be described by a linear system.

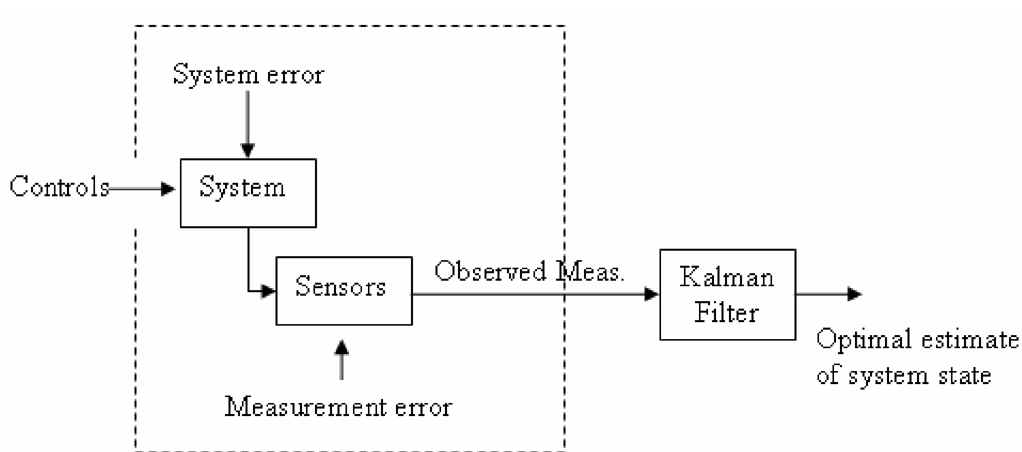

Figure 2. Typical Kalman Filter Application [5]

Let us assume that the random process to be estimated can be modeled as follows:

$$X_k = F_{k-1} X_{k-1} + G_{k-1} u_{k-1} + w_{k-1} \tag{2.1}$$

with an observation (measurement) written as:

$$Y_k = H_k X_k + v_k \tag{2.2}$$

Where

$\boldsymbol{X_k}$ = (n x 1) process state vector at time t_k, $\boldsymbol{X_k} \in \Re^n$

$\boldsymbol{F_k}$ = (n x n) transition matrix

$\boldsymbol{w_k}$ = (n x 1) system noise vector – assumed to be a white sequence with known covariance structure

$\boldsymbol{Y_k}$ = (m x 1) measurement vector, $\boldsymbol{Y_k} \in \Re^m$

$\boldsymbol{H_k}$ = (m x n) measurement matrix

$\boldsymbol{v_k}$ = (m x 1) measurement noise vector – assumed to be a white sequence with known covariance structure and having zero cross-correlation with the $\boldsymbol{w_k}$ sequence

$\mathbf{u_k}$ = (l x 1) known input vector, $u_k \in \Re^l$

$\mathbf{G_k}$ = $(n\ x\ l)$ control matrix of input $u_k \in \Re^l$.

2.1 Discrete Kalman Filter

Let us write the discrete Kalman filter equations for the system defined by Equations 2.1 and 2.2. The random variables w_k and v_k in Equations 2.1 and 2.2 represents the system and measurement noise vectors respectively. They are assumed to be independent (of each other), white and with normal probability distributions as follows:

$$w_k \sim N(0, Q_k) \tag{2.3}$$

$$v_k \sim N(0, R_k) \tag{2.4}$$

The process noise covariance Q_k and measurement noise covariance R_k matrices might change with each time step or measurement.

The covariance matrices for the w_k and v_k noise vectors are given as follows:

$$E\{w_k w_i'\} = \begin{cases} Q_k, & i = k \\ 0 & i \neq k \end{cases} \tag{2.5}$$

$$E\{v_k v_i'\} = \begin{cases} R_k, & i = k \\ 0 & i \neq k \end{cases} \tag{2.6}$$

$$E\{w_k v_i'\} = \{0, \quad for\ all\ \ k\ \ and\ i \tag{2.7}$$

Let's say that $\hat{X}_k^- \in \Re^n$ (note that "super minus") to be our *a priori* or *predicted* state estimate at step k given knowledge of the process prior to step k, and $\hat{X}_k \in \Re^n$ to be our *a posteriori* or *updated* state estimate at step k given measurement Y_k [14]. A priori state estimate at step k "$\hat{X}_k^-$" can be written by:

$$\hat{X}_k^{\ -} = F_{k-1}\hat{X}_{k-1} + G_{k-1}u_{k-1} \tag{2.8}$$

Now, we can define *a priori* and *a posteriori* estimate errors as follows:

$$e_k^- \equiv X_k - \hat{X}_k^-, \text{ a priori estimate error} \tag{2.9}$$

$$e_k \equiv X_k - \hat{X}_k, \text{ a posteriori estimate error} \tag{2.10}$$

So, the *a priori* estimate error covariance is given by:

$$P_k^{\ -} = E\left[e_k^- e_k^{-T}\right]. \tag{2.11}$$

And the *a posteriori* estimate error covariance is given by:

$$P_k = E\left[e_k e_k^{\ T}\right]. \tag{2.12}$$

The *a posteriori* state estimate $\hat{X}_k$ is a linear combination of an *a priori* estimate $\hat{X}_k^-$ and a weighted difference between an actual measurement Y_k and a measurement prediction $H_k \hat{X}_k^-$:

$$\hat{X}_k = \hat{X}_k^- + K_k \left(Y_k - H_k \hat{X}_k^- \right) \tag{2.13}$$

The difference $\left(Y_k - H_k \hat{X}_k^- \right)$ in Equation 2.13 is called the measurement *innovation* or the *residual*. It reflects the discrepancy between the predicted measurement $H_k \hat{X}_k^-$ and the actual measurement Y_k. A measurement innovation of zero means that the two are in complete agreement.

Now, let's find the *n x m* matrix K_k (Equation 2.13) which is said to be the *gain* or *blending factor* that minimizes the a posteriori error covariance (Equation 2.13). The minimization can be done by first substituting Equation 2.13 into the Equation 2.12, doing the indicated expectations, taking the derivative with respect to K_k, setting that result equal to zero, and solving for K_k [2].
As a result, one form of the resulting K_k that minimizes Equation 2.12 is given by:

$$K_k = P_k^- H_k^{\ T} \left(H_k P_k^- H_k^{\ T} + R_k \right)^{-1} \tag{2.14}$$

From Equation 2.14, we can see that as the measurement error covariance R_k approaches zero, the gain K_k weights the residual more heavily. Specifically,

$$\lim_{R_k \to 0} K_k = H_k^{\ -1} , \tag{2.15}$$

Or, as the a priori estimate error covariance $P_k^{\ -}$ approaches zero, the gain K_k weights the residual less heavily:

$$\lim_{P_k^- \to 0} K_k = 0 . \tag{2.16}$$

Actually, the Kalman filter works by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. So the equations for the Kalman filter fall into two groups: *time update or prediction* equations and *measurement update* equations.

The time update equations can also be the thought of as predictor equations and measurement update equations can be thought of as corrector equations. So, the algorithm looks like a predictor-corrector algorithm.

Now, we can write the time update equations and the measurement update equations as in Table 1 and Table 2.

Table 1. Prediction or Time Update Equations

Predicted state estimate	$\hat{X}_k^{-} = F_{k-1}\hat{X}_{k-1} + G_{k-1}u_{k-1}$	(2.17)
Predicted error covariance	$P_k^{-} = F_{k-1}P_{k-1}F_{k-1}^{T} + Q_k$	(2.18)

Table 2. Measurement Update Equations

Innovation or residual	$Inn = Y_k - H_k \hat{X}_k^{-}$	(2.19)
Innovation covariance	$S_k = H_k P_k^{-} H_k^{T} + R_k$	(2.20)
Kalman Gain	$K_k = P_k^{-} H_k^{T} S_k^{-1}$	(2.21)
Updated state estimate	$\hat{X}_k = \hat{X}_k^{-} + K_k\left(Y_k - H_k \hat{X}_k^{-}\right)$	(2.22)
Updated error covariance	$P_k = P_k^{-} - K_k S_k K_k^{T} \ or \ P_k = \left(I - K_k H_k\right)P_k^{-}$	(2.23)

Figure 3 shows one cycle in the state estimation of a linear system. The left column of Figure 3 shows the true system's evolution from time k-1 to time k with the input u_{k-1} and the system noise w_{k-1}; the measurement follows from the new state and the measurement noise v_k. Estimation of the state is done in the middle column of the figure and it consists of:

(i) state and measurement **prediction** (time update),

(ii) state **update** (measurement update).

The known input is used by the state estimator to obtain the state prediction for the next time. The state update requires the filter gain, obtained in the course of the covariance calculations in the right column. Also, updating the state covariance is done in the right column of the Figure 3.

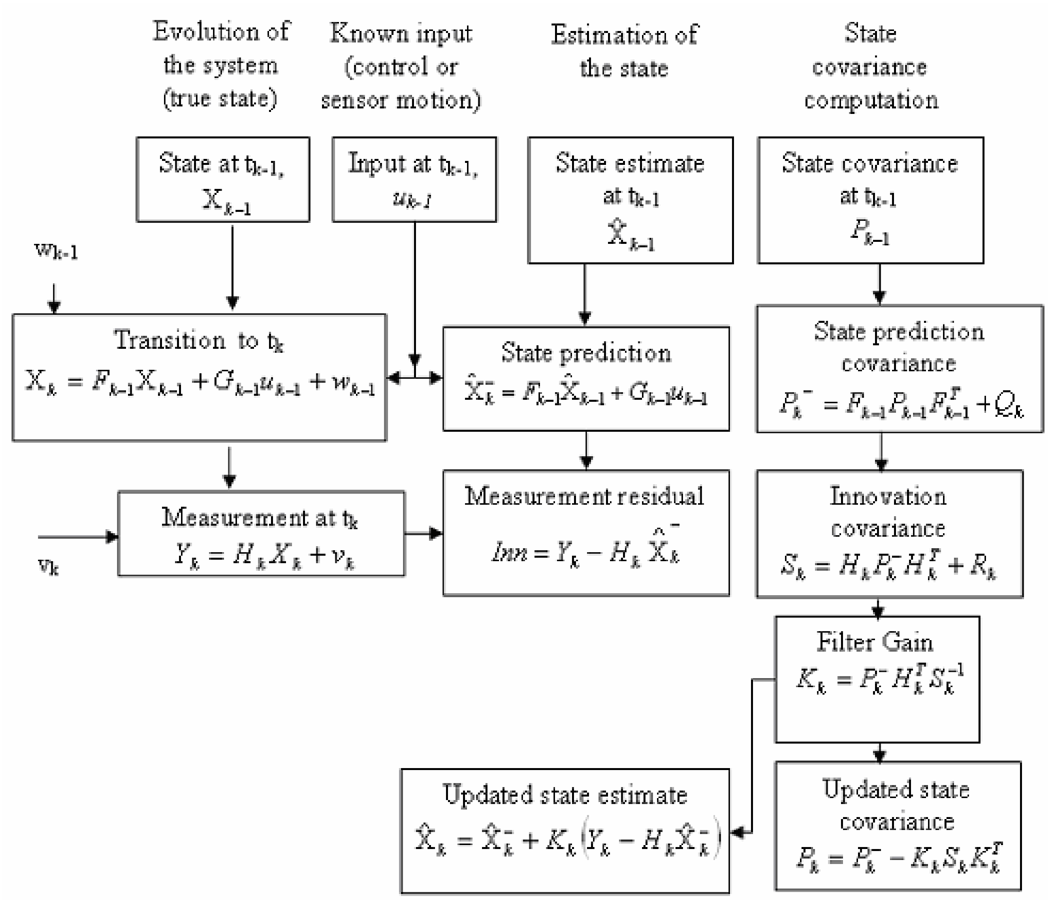

Figure 3. One Cycle in the State Estimation of a Linear System [18]

CHAPTER 3

SYSTEM MODELING AND KALMAN FILTER PREDICTOR ALGORITHMS

The issue in this thesis is radar pulse repetition interval (PRI) tracking. The time interval between two pulses emitted by radar is called PRI. In Figure 4, a radar pulse is shown. Radar may have a constant PRI, or it may have some form of PRI agility, in other words the time interval between pulses varies on a pulse-to-pulse basis. Today, a large number of anti-ship missiles use radar as the homing device. Some currently have PRI agility. In the future, radars are expected to have both some form of PRI and frequency agility. The most known types of PRI agility are staggered PRIs, PRIs with random jitter (Step-PRI), and sinusoidally modulated PRIs [4].

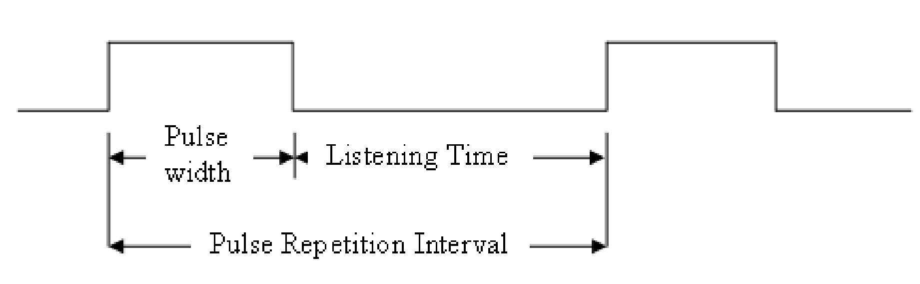

Figure 4. Pulse Repetition Interval-PRI

In electronic warfare (EW), PRI tracking algorithm is needed as the basis for a deceptive countermeasures technique. After selecting an emitter for deceptive countermeasures, the PRI tracking algorithm should predict the PRI. Predicting PRI is equivalent to predicting the pulse time of arrival (TOA) of the next pulse. So jammer can be gated on at that time. Another important subject is that the algorithm should also generate a measure of the variance of the prediction. The variance of the prediction, along with the measured pulse width, is used to control the width of the jamming pulse. Keeping the jamming pulse length as short as possible is very important. When the variance of the prediction is small, the jamming pulse length approaches the received pulse width. This allows two significant operational advantages. First, the electronic counter measure (ECM) transmitter can be time multiplexed to handle multiple incoming threats simultaneously. Second, own ship RFI problems are minimized [4]. The step-PRI and staggered-PRI forms are considered in this chapter.

3.1 Step Pulse Repetition Interval (Step-PRI) Sequence

3.1.1 System and observation models for Step-PRI sequence

3.1.1.1 Step-PRI sequence models in the case of no missing pulse

Many radars emit a sequence of pulses whose times of arrival (TOAs) are contaminated by jitter (TOA jitter is represented by White Gaussian noise "$w_j \sim N(0,\sigma_w^2)$") as shown in Figure 5.

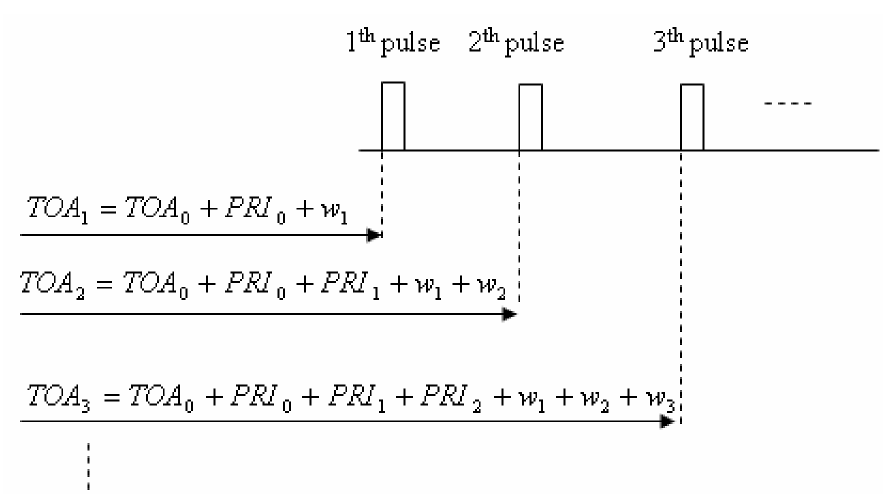

Figure 5. TOA Values Written for Step-PRI Case

A step PRI emition can be modeled according to cumulative model [10, 11] as follows (see Figure 5):

$$TOA_1 = TOA_0 + PRI_0 + w_1$$

$$TOA_2 = TOA_1 + PRI_1 + w_2$$

$$\vdots$$

$$TOA_{j+1} = TOA_j + PRI_j + w_{j+1} \qquad w_j \sim N\left(0, \sigma_w^2\right) \quad \text{j=0, 1…} \tag{3.1}$$

The sequence $\{w_j\}$ corresponds to cumulative jitter (CJ) [10, 11] component with distribution $N\left(0, \sigma_w^2\right)$.

A white Gaussian noise "$\beta_j \sim N\left(0, \sigma_\beta^2\right)$" is intentionally added to PRI values by radar system. So, the equation for the PRI values is written as:

$$PRI_{j+1} = PRI_j + \beta_j \qquad \beta_j \sim N\left(0, \sigma_{\beta_j}^2\right) \quad \text{j=0, 1 …} \tag{3.2}$$

Where PRI_j is assumed to be a slowly varying parameter and β_j is a Gaussian random variable with distribution $N\left(0, \sigma_{\beta_j}^2\right)$.

Measurements (measured TOA values are denoted by Y_j) are done such that:

$$Y_j = TOA_j + v_j \qquad v_j \sim N\left(0, \sigma_j^2\right) \qquad \text{j=0, 1…} \tag{3.3}$$

Where v_j is the measurement noise with distribution $N\left(0, \sigma_j^2\right)$. So, the equations 3.1, 3.2 and 3.3 can be rewritten as follows:

System equation:

$$X_{k+1} = F_k X_k + u_k \qquad u_k = \begin{pmatrix} w_{k+1} \\ \beta_k \end{pmatrix}, \quad w_k \sim N\left(0, \sigma_w^2\right) \textit{ and } \beta_k \sim N\left(0, \sigma_\beta^2\right) \tag{3.4}$$

Observation equation: $Y_k = H\ X_k + v_k \qquad v_k \sim N\left(0, \sigma_v^2\right)$ (3.5)

Where,

$$X_k = \begin{bmatrix} TOA_k \\ PRI_k \end{bmatrix} \text{ is the state vector,}$$

$H = [1 \quad 0]$ is the measurement vector,

$F_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ is the state transition matrix if there is no missing pulse at that time and it will be denoted by F at the points when there is no missing pulse,

Y_k is the measured time of arrival (TOA),

$u_k = \begin{bmatrix} w_{k+1} \\ \beta_k \end{bmatrix}$ is the system noise vector,

v_k is the measurement noise.

It is important to note that constant PRI is a special case of Step-PRI agility. In other words, Step-PRI is called as constant PRI when there is no jitter (β_j) on the PRI values.

In these equations, the measured data set $Y_0, Y_1, ..., Y_k, \ldots$, the variance of β_j (σ_β^2), the variance of TOA jitter (σ_w^2) and the variance of measurement noise (σ_v^2) are all known.

The covariance matrix of u_k is denoted by Q and is given by $Q = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_\beta^2 \end{bmatrix}$.

3.1.1.1.1 Kalman filter equations

In the algorithm, Kalman filter is initiated at time 2, initial TOA and PRI values at time 2 are calculated as follows:

To find initial value (predicted value according to the previous data) of the state vector, we must use the previous data (measured TOA values). So, measured TOA values Y_0 and Y_1 are used to find a prediction for the state vector X_2 which is written as follows:

$T\hat{O}A_2 = Y_1 + Y_1 - Y_0 = 2Y_1 - Y_0$: *predicted TOA value based on the previous measured TOA values* Y_1 *and* Y_0.

$P\hat{R}I_2 = Y_1 - Y_0$: *predicted PRI value based on the previous measured TOA values* Y_1 *and* Y_0.

So, initial value of the state vector is $\hat{X}_2 = \begin{bmatrix} 2Y_1 - Y_0 \\ Y_1 - Y_0 \end{bmatrix}$.

The state prediction error related to this initial estimation is found as:

$$X(2) - \hat{X}(2) = \begin{bmatrix} TOA_2 \\ PRI_2 \end{bmatrix} - \begin{bmatrix} 2TOA_1 - TOA_0 + 2v_1 - v_0 \\ TOA_1 - TOA_0 + v_1 - v_0 \end{bmatrix} \tag{3.6}$$

From the Equations 3.1 and 3.2 we know that;

$TOA_2 = TOA_1 + PRI_1 + w_2$ and $PRI_2 = PRI_1 + \beta_1$,

$$TOA_1 = TOA_0 + PRI_0 + w_1 \text{ and } PRI_1 = PRI_0 + \beta_0. \tag{3.7}$$

So, the value of TOA_2 and PRI_2 can be rewritten as:

$$TOA_2 = TOA_0 + 2PRI_0 + \beta_0 + w_1 + w_2 \text{ and } PRI_2 = PRI_0 + \beta_0 + \beta_1. \tag{3.8}$$

Using Equations 3.7 and 3.8, the state prediction error (Equation 3.6) can be written as:

$$X(2) - \hat{X}(2) = \begin{bmatrix} w_2 - w_1 + v_0 - 2v_1 + \beta_0 \\ v_0 - v_1 - w_1 + \beta_0 + \beta_1 \end{bmatrix}. \tag{3.9}$$

So, the initial covariance matrix of the state prediction error can be written as:

$$P(2) = \begin{bmatrix} 5\sigma_v^2 + 2\sigma_w^2 + \sigma_\beta^2 & \sigma_w^2 + 3\sigma_v^2 + \sigma_\beta^2 \\ \sigma_w^2 + 3\sigma_v^2 + \sigma_\beta^2 & 2\sigma_v^2 + \sigma_w^2 + 2\sigma_\beta^2 \end{bmatrix}. \tag{3.10}$$

Using the time update and measurement update equations (Equations 2.21-2.23), the Kalman gain K_k, optimal estimates of the state vector after the measurement and updated error covariance matrix (see Equations 2.21, 2.22, 2.23) are written as follows:

$\hat{X}_k^- = F\hat{X}_{k-1}$ $\qquad\qquad$ $\hat{X}_k^-$: predicted state

$\hat{P}_k^- = F\hat{P}_{k-1}F^T + Q$ $\hat{P}_k^-$: State prediction error covariance

Kalman gain is given by $K_k = \hat{P}_k^- H^T S_k^{-1} = \hat{P}_k^- H^T \left\{H\hat{P}_k^- H^T + R\right\}^{-1}$

$$= \left(F\hat{P}_{k-1}F^T + Q\right)H^T\left(H\left(F\hat{P}_{k-1}F^T + Q\right)H^T + R\right)^{-1}$$

Where R is the measurement covariance with a covariance of σ_v^2.

So, Kalman gain for step k+1 which is denoted by "K_{k+1}" can be written as:

$$K_{k+1} = \left(F\hat{P}_k F^T + Q\right)H^T\left(H\left(F\hat{P}_k F^T + Q\right)H^T + R\right)^{-1} \tag{3.11}$$

The updated state estimate at step k+1 ($\hat{X}_{k+1}$) is given by:

$$\hat{X}_{k+1} = F\hat{X}_k + K_{k+1}\left\{Y_{k+1} - H(F\hat{X}_k)\right\} \tag{3.12}$$

The updated error covariance matrix at step k+1 ($\hat{P}_{k+1}$) is given by:

$$\hat{P}_{k+1} = (I - K_{k+1}H)\left(F\hat{P}_k F^T + Q\right), \tag{3.13}$$

It's known that state prediction error covariance matrix is 2x2 matrix, also the Kalman gain is 2x1 matrix. Let us denote these matrices as follows:

$$\hat{P}_k \overset{\Delta}{=} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}, \quad K_k \overset{\Delta}{=} \begin{bmatrix} f_k \\ g_k \end{bmatrix}$$

Now, let us evaluate $\hat{P}_k$ and K_k. Substituting Equation 3.11 into the Equation 3.13, following equation is obtained for state prediction error covariance matrix at step k+1 ($\hat{P}_{k+1}$):

$$\hat{P}_{k+1} = \left\{I - \left(\left(F\hat{P}_k F^T + Q\right)H^T\left(H\left(F\hat{P}_k F^T + Q\right)H^T + R\right)^{-1}H\right)\right\}\left(F\hat{P}_k F^T + Q\right) \tag{3.14}$$

Since the dimensions of H, $\hat{P}_k$ and R are known as to be 1x2, 2x2, 1x1 respectively, the dimension of the equation $\left\{H\left(F\hat{P}_k F^T + Q\right)H^T + R\right\}$ is found to be 1x1. The result of $\left(F\hat{P}_k F^T + Q\right)$ is given as:

$$\left(F\hat{P}_k F^T + Q\right) = \begin{bmatrix} a_k + b_k + c_k + d_k + \sigma_u^2 & b_k + d_k \\ c_k + d_k & d_k + \sigma_\beta^2 \end{bmatrix}.$$

So, the result of $\left\{H\left(F\hat{P}_k F^T + Q\right)H^T + R\right\}$ is found as:

$$\left\{H\left(F\hat{P}_k F^T + Q\right)H^T + R\right\}^{-1} = \frac{1}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} \tag{3.15}$$

Substituting the Equation 3.15 into the Equation 3.14 and using the actual values of the terms given in Equation 3.14, following equation is obtained:

$$\hat{P}_{k+1} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{\left(\left(F\hat{P}_k F^T + Q\right)\begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\right)}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} \right) \left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_\beta^2 \end{bmatrix} \right)$$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{\left[\left(F\hat{P}_k F^T + Q\right)\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\right]}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} \right) \left(\begin{bmatrix} a_k + b_k + c_k + d_k & b_k + d_k \\ c_k + d_k & d_k \end{bmatrix} + \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_\beta^2 \end{bmatrix} \right)$$

Let us substitute the result of $\left(F\hat{P}_k F^T + Q\right) = \begin{bmatrix} a_k + b_k + c_k + d_k + \sigma_w^2 & b_k + d_k \\ c_k + d_k & d_k + \sigma_\beta^2 \end{bmatrix}$ into the equation above as follows:

$$= \begin{bmatrix} 1 - \frac{a_k + b_k + c_k + d_k + \sigma_w^2}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} & 0 \\ -\frac{c_k + d_k}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} & 1 \end{bmatrix} \left(\begin{bmatrix} a_k + b_k + c_k + d_k & b_k + d_k \\ c_k + d_k & d_k \end{bmatrix} + \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_\beta^2 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 1 - \dfrac{a_k + b_k + c_k + d_k + \sigma_w^2}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} & 0 \\ -\dfrac{c_k + d_k}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2} & 1 \end{bmatrix} \begin{bmatrix} a_k + b_k + c_k + d_k + \sigma_w^2 & b_k + d_k \\ c_k + d_k & d_k + \sigma_\beta^2 \end{bmatrix} \quad (3.16)$$

From the Equation 3.10, it is seen that $b_k = c_k$ for k=2, this equality will also be valid for the iterations with $k \geq 3$. So using this result in the Equation 3.16, following equation is obtained:

$$\hat{P}_{k+1} = \begin{bmatrix} a_{k+1} & b_{k+1} \\ b_{k+1} & d_{k+1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 - \dfrac{a_k + 2b_k + d_k + \sigma_w^2}{a_k + 2b_k + d_k + \sigma_w^2 + \sigma_v^2} & 0 \\ -\dfrac{b_k + d_k}{a_k + 2b_k + d_k + \sigma_w^2 + \sigma_v^2} & 1 \end{bmatrix} \begin{bmatrix} a_k + 2b_k + d_k + \sigma_w^2 & b_k + d_k \\ b_k + d_k & d_k + \sigma_\beta^2 \end{bmatrix} \quad (3.17)$$

So, Kalman gain (K_{k+1}) can be written as follows:

$$K_{k+1} = \left(F\hat{P}_k F^T + Q\right)H^T \left(H\left(F\hat{P}_k F^T + Q\right)H^T + R\right)^{-1}$$

$$\begin{bmatrix} f_{k+1} \\ g_{k+1} \end{bmatrix} = \begin{bmatrix} a_k + b_k + c_k + d_k + \sigma_w^2 & b_k + d_k \\ c_k + d_k & d_k + \sigma_\beta^2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{1}{a_k + b_k + c_k + d_k + \sigma_w^2 + \sigma_v^2}$$

$$\begin{bmatrix} f_{k+1} \\ g_{k+1} \end{bmatrix} = \begin{bmatrix} \dfrac{a_k + 2b_k + d_k + \sigma_w^2}{a_k + 2b_k + d_k + \sigma_w^2 + \sigma_v^2} \\ \dfrac{b_k + d_k}{a_k + 2b_k + d_k + \sigma_w^2 + \sigma_v^2} \end{bmatrix}, \text{ since } b_k = c_k \text{ for all k.} \quad (3.18)$$

Using Equations 3.17 and 3.18, the equations obtained are as follows:

$$\hat{P}_{k+1} = \begin{bmatrix} a_{k+1} & b_{k+1} \\ b_{k+1} & d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 - f_{k+1} & 0 \\ -g_{k+1} & 1 \end{bmatrix} \begin{bmatrix} a_k + 2b_k + d_k + \sigma_w^2 & b_k + d_k \\ b_k + d_k & d_k + \sigma_\beta^2 \end{bmatrix}$$

$$a_{k+1} = (1 - f_{k+1})(a_k + 2b_k + d_k + \sigma_w^2) = (1 - f_{k+1})(a_k + \sigma_w^2 + b_k) + (1 - f_{k+1})(b_k + d_k),$$

$$b_{k+1} = (1 - f_{k+1})(b_k + d_k), \quad (3.19)$$

So, $a_{k+1} = (1 - f_{k+1})(a_k + \sigma_w^2 + b_k) + b_{k+1}$, (3.20)

$$d_{k+1} = -g_{k+1}(b_k + d_k) + d_k + \sigma_\beta^2 \tag{3.21}$$

Where

$$f_{k+1} = \frac{a_k + 2b_k + d_k + \sigma_w^2}{a_k + 2b_k + d_k + \sigma_w^2 + \sigma_v^2} \text{ and } g_{k+1} = \frac{b_k + d_k}{a_k + 2b_k + d_k + \sigma_w^2 + \sigma_v^2} \tag{3.22}$$

Using Equation 3.12, we obtain the updated state estimate as follows:

$$\begin{bmatrix} T\hat{O}A_{k+1} \\ P\hat{R}I_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} T\hat{O}A_k \\ P\hat{R}I_k \end{bmatrix} + \begin{bmatrix} f_{k+1} \\ g_{k+1} \end{bmatrix}\left\{Y_{k+1} - \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} T\hat{O}A_k \\ P\hat{R}I_k \end{bmatrix}\right\}$$

$$T\hat{O}A_{k+1} = T\hat{O}A_k + P\hat{R}I_k + f_{k+1}\left(Y_{k+1} - T\hat{O}A_k - P\hat{R}I_k\right) \tag{3.23}$$

$$P\hat{R}I_{k+1} = P\hat{R}I_k + g_{k+1}\left(Y_{k+1} - T\hat{O}A_k - P\hat{R}I_k\right) \tag{3.24}$$

3.1.1.2 Step-PRI sequence models in the case of missing pulse

While deriving the Equations 3.19 - 3.24 it is assumed that there is no missing pulse in the measured data. Let us rewrite the equations if there are missing pulses in the measured data. In Figure 6, (a) shows the pulses with their indices in the case of no missing pulse. Now, assume that, the $(k+1)^{th}$ pulse has been missed in the measurement, so the next pulse ($(k+2)^{th}$ pulse) is renamed as $(k+1)^{th}$ pulse, the $(k+3)^{th}$ pulse is renamed as $(k+2)^{th}$ pulse and so on as shown in Figure 6 (b):

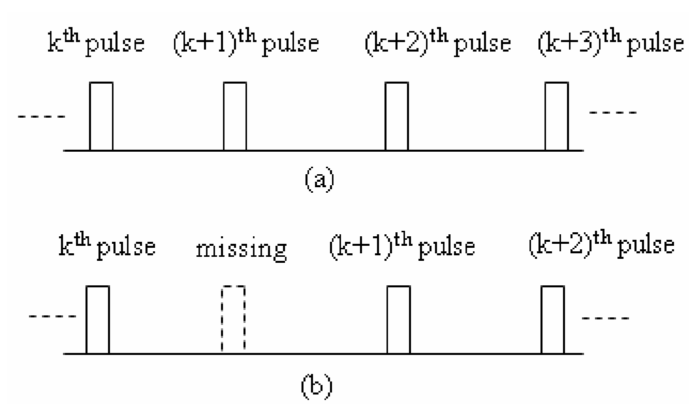

Figure 6. Missing Pulse case in the Measured Data

So, according to the system equation (Equation 3.4) let write the system equations for both case as follows:

<u>No missing pulse - Figure 6 (a)</u>

$$X_k = F\ X_{k-1} + u_{k-1}$$

$$X_{k+1} = F\ X_k + u_k$$

$$X_{k+2} = F\ X_{k+1} + u_{k+1}$$

$$\vdots$$

<u>(k+1)th pulse is missing – Figure 6 (b)</u>

$$X_k = F\ X_{k-1} + u_{k-1}$$

~~$X_{k+1} = F\ X_k + u_k$~~

$$X_{k+1} = F\ \left(F\ X_{k-1} + u_{k-1}\right) + u_k$$

$$X_{k+1} = F^2 X_{k-1} + F u_{k-1} + u_k$$

$$X_{k+2} = F X_{k+1} + u_{k+1}$$

$$\vdots$$

So, at the missing pulse point, we obtain the following system equation:

$$X_{k+1} = F^2 X_{k-1} + F u_{k-1} + u_k \qquad (3.25)$$

From Equation 3.25, we can say that transition matrix and system noise change at the missing pulse points.

So, at the missing pulse points TOA and PRI values are calculated as follows:

$$\begin{bmatrix} T\hat{O}A_{k+1} \\ P\hat{R}I_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T\hat{O}A_k \\ P\hat{R}I_k \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T\hat{O}A_k \\ P\hat{R}I_k \end{bmatrix}$$

$$T\hat{O}A_{k+1} = T\hat{O}A_k + 2P\hat{R}I_k$$

$$P\hat{R}I_{k+1} = P\hat{R}I_k$$

In Equation 3.25, if we denote the noise "$F u_{k-1} + u_k$" by $u_k^{'}$ ($u_k^{'} \overset{\Delta}{=} F u_{k-1} + u_k$) then the noise vector $u_k^{'}$ can be calculated as:

$$u_k^{'} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_k \\ \beta_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k+1} \\ \beta_k \end{bmatrix} = \begin{bmatrix} \beta_{k-1} + w_k + w_{k+1} \\ \beta_{k-1} + \beta_k \end{bmatrix}.$$

So, the covariance matrix ($Q^{'}$) of the noise vector $u_k^{'}$ is given by:

$$Q^{'} = \begin{bmatrix} 2\sigma_w^2 + \sigma_\beta^2 & \sigma_\beta^2 \\ \sigma_\beta^2 & 2\sigma_\beta^2 \end{bmatrix}$$

Also, at missing pulse points, the state prediction error covariance matrix is calculates as:

$\hat{P}_{k+1} = F^2 P_k F^{2^T} + Q^{'}$ $\quad \hat{P}_{k+1}$: estimate error covariance matrix at the missing pulse point.

$$\hat{P}_{k+1} = \begin{bmatrix} a_{k+1} & b_{k+1} \\ b_{k+1} & d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 2\sigma_w^2 + \sigma_\beta^2 & \sigma_\beta^2 \\ \sigma_\beta^2 & 2\sigma_\beta^2 \end{bmatrix}, \; b_k = c_k \text{ for all k}$$

$$\hat{P}_{k+1} = \begin{bmatrix} a_{k+1} & b_{k+1} \\ b_{k+1} & d_{k+1} \end{bmatrix} = \begin{bmatrix} a_k + 4b_k + 4d_k + 2\sigma_w^2 + \sigma_\beta^2 & b_k + 2d_k + \sigma_\beta^2 \\ b_k + 2d_k + \sigma_\beta^2 & d_k + 2\sigma_\beta^2 \end{bmatrix} \tag{3.26}$$

So,

$$a_{k+1} = a_k + 4b_k + 4d_k + 2\sigma_w^2 + \sigma_\beta^2, \tag{3.27}$$

$$b_{k+1} = b_k + 2d_k + \sigma_\beta^2, \tag{3.28}$$

$$d_{k+1} = d_k + 2\sigma_\beta^2. \tag{3.29}$$

3.1.1.2.1 Detection of missing pulses

As it can be seen from the Equations 3.26 - 3.29, the observation at the missing pulse point is not used for updating the existing tracks or for initiating new tracks. The use of standard Kalman filtering when there are missing pulses in the measured data can lead to divergence because the covariance matrix may not reflect the increased error due to miscorrelation. Thus, the detection of missing pulses or in other words selection of the measurements to be incorporated into the filter is done using validation or association region (gate) [18, 19]. Validation region is defined as an area of the measurement space where the observation will be found with some high probability. A measurement in the gate is a valid candidate to for Kalman filter update equations [21].

It's assumed that the true measurement conditioned on the past is normally (Gaussian) distributed with its probability function (pdf) given by [18]:

$$p[Y_{k+1} \mid Y_{1:k}] = N[Y_{k+1}; \hat{Y}_{k+1}^{-}, S_{k+1}]$$

Where, $\hat{Y}_{k+1}^{-}$ ($\hat{Y}_{k+1}^{-} = H\hat{X}_{k+1}^{-} = HF\hat{X}_{k}$) is the predicted value (mean) of the measurement and S_{k+1} is the associated innovation covariance. The validation region is related to the inverse of the innovation covariance matrix S_{k+1} and the innovation (Equation 2.19). It describes an ellipse in the measurement space and it is the minimum volume that contains a given probability mass under the Gaussian assumption. The validation gate is defined as:

$$V(k+1, \gamma) = \left\{ Y : [Y_{k+1} - \hat{Y}_{k+1}^{-}]^{T} S_{k+1}^{-1} [Y_{k+1} - \hat{Y}_{k+1}^{-}] \leq \gamma \right\} \tag{3.30}$$

with probability determined by the gate threshold γ. The region defined by the Equation 3.30 is called as the ***validation gate*** or ***ellipsoidal gate***. The semi-axes of the ellipsoid (Equation 3.30) are the square roots of the eigenvalues of γS_{k+1}. The innovation $Y_{k+1} - \hat{Y}_{k+1}^{-}$ that defines the validation region is chi-square distributed with number of degrees of freedom equal to the dimension of the measurement Y_{k+1} which is 1. The threshold γ can be obtained from the standard chi-square tables and is chosen based on the confidence level required. Selecting a too small gate size may lead to miss the true measurements; whereas selecting too large gate size is computationally expensive [21]. For example, with dim(Y_{k+1})=1 and setting $\gamma = 4$ results in 95% of the probability mass inside the validation gate [18].

Let us rewritten Equation 3.30 as:

$$(Y_{k+1} - HF\hat{X}_{k})^{T} S_{k+1}^{-1} (Y_{k+1} - HF\hat{X}_{k}) \leq \gamma$$

Also, using the Equation 2.20, the measurement innovation covariance at step 'k+1' can be written as follows:

$$S_{k+1} = HP_{k+1}H^T + R = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} a_{k+1} & b_{k+1} \\ b_{k+1} & d_{k+1} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma_v^2 = a_{k+1} + \sigma_v^2 \tag{3.31}$$

So,

$$\left(Y_{k+1} - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T\hat{O}A_k \\ P\hat{R}I_k \end{bmatrix} \right)^2 \leq \gamma S_{k+1}$$

Using the result found for innovation covariance S_{k+1}:

$$\left(Y_{k+1} - T\hat{O}A_k - P\hat{R}I_k \right)^2 \leq \gamma \left(a_{k+1} + \sigma_v^2 \right) \tag{3.32}$$

Following equation is obtained by taking square roots of the Equation 3.32:

$$\left| Y_{k+1} - T\hat{O}A_k - P\hat{R}I_k \right| \leq \sqrt{\gamma \left(a_{k+1} + \sigma_v^2 \right)}$$

Finally, the validation region or gate equation can be written as:

$$T\hat{O}A_k + P\hat{R}I_k - \phi\sqrt{S_{k+1}} \leq Y_{k+1} \leq T\hat{O}A_k + P\hat{R}I_k + \phi\sqrt{S_{k+1}}, \qquad \phi = \sqrt{\gamma}$$

Where, the square root $\phi = \sqrt{\gamma}$ is referred to as the "number of sigma's" (standard deviations) of the gate.

3.2 Staggered Pulse Repetition Interval (Staggered PRI) Sequence

Several adaptive measures may be assumed by radar to lessen its susceptibility to electronic counter measures (ECM); one which will make the job of a repeater jammer more difficult is the incorporation of staggered pulse trains.

However, the same basic laws of nature apply to exotic pulse train generation (*i.e.*, the elapsed time between any group of pulses cannot be less than the desired maximum range of the radar). The staggered pulse repetition frequency (PRF) also enhances associated radar features such as Moving Target indication by reducing the effects of blind spots in the radar.

A staggered pulse [4, 20] sequence is fundamentally a basic PRF with this same PRF impressed upon itself one or more times. Each level of impression (stagger) utilizes a different start time or reference which will preclude the generation of concurrent pulses or pulses shadowing one another. The number of levels (or positions) is the number of times the basic PRF/IPP (inter-pulse period) is integrated in the pulse train. As mentioned above, each level has the same characteristic PRF and pulse width (PW), but the Time to First Event (TFE) for each level is different. The PRF of the radar is the sum of all the pulse trains so that if a radar warning receiver (RWR) operated on PRF, the additional identification inherent in the stagger pattern would not be useful. This problem is overcome by measuring PRI rather than PRF so that the RWR measures the basic PRI a number of times equal to the number of stagger levels [20].

So, a staggered PRI sequence can be defined as a sequence of several different pulse intervals in a repeating pattern (periodic). For example, the staggered PRI sequence {240, 290, 350, 410, 240, 240, 290, 350, 410, 240, 240 …} has four distinct pulse intervals and a period of five. It is referred to as a 4-element, 5-position staggered PRI sequence with stagger elements of 240, 290, 350 and 410. Figure 7 illustrates the time relationship involved in the generation of this 4-element, 5-position staggered PRI sequence [4].

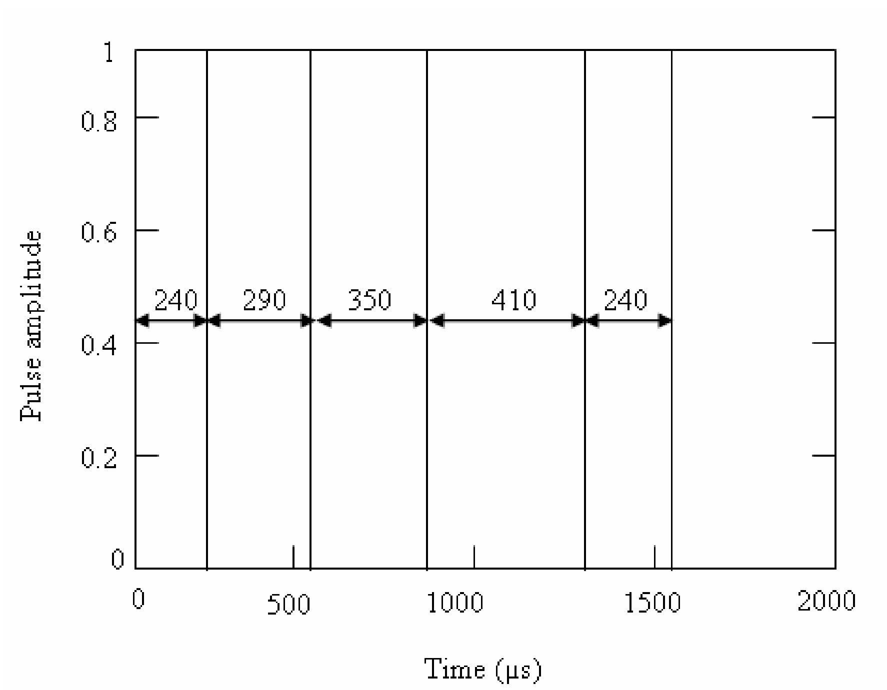

Figure 7. Four-Element, Five-Position Staggered PRI Sequence

An algorithm is presented for determining the period of the staggered PRI sequence and then two prediction algorithms are presented for PRI estimation. While designing the algorithms, the following assumptions were made [4]:

1. Emitters which we deal with are specified by the electronic attack system for tracking.
2. A group of pulses related to each emitter is accumulated in a buffer for tracking.
3. An electronic support receiver has sorted PRI measurements so that all measurements in a given buffer will come from the same emitter. However since all ES systems make mistakes, the sorting process is not perfect. As a result, missing pulses [12] as well as jitter and measurement noise may corrupt the PRI sequence in the buffer. Missing pulses occur due to the failure of the measuring apparatus to detect or receive a pulse.
4. The received data stream contains pulses from a single emitter, but missing pulses, jitter and measurement noise may corrupt the data stream.

All of the data used was simulated. This data consist of TOAs and PRIs written to a file. The received data corrupted by missing pulses and jitter is modeled in the simulated data. We assume that our data consists of an ordered sequence measured time of arrival values "τ_j", j=1, 2… Following cumulative model [10, 11] is used while simulating data (see Figure 8):

TOA_j : The time of arrival of the jth pulse

τ_j : The measured time of arrival (TOA)

$$TOA_j = TOA_{j-1} + PRI_j + w_j \qquad w_j \sim N\left(0, \sigma_w^2\right) \tag{3.33}$$

$$\tau_j = TOA_j + v_j \qquad v_j \sim N\left(0, \sigma_v^2\right) \tag{3.34}$$

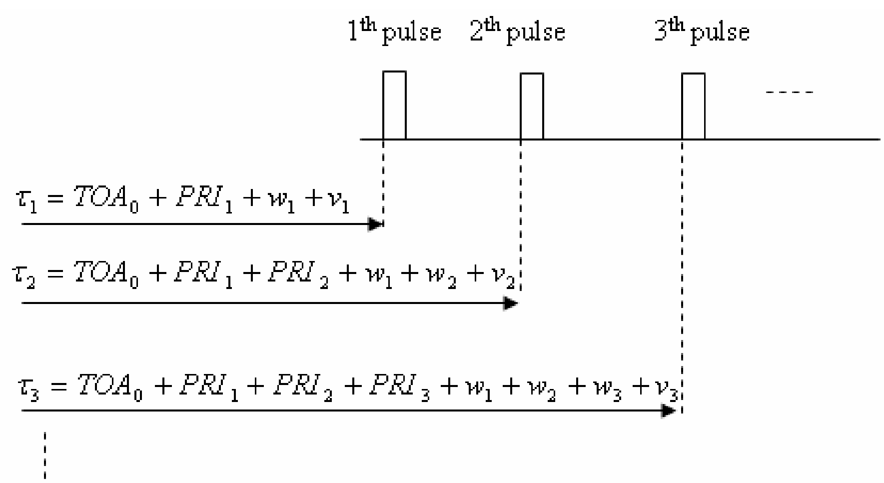

Figure 8. Measured-TOA Values Written for Staggered-PRI Case

The independent, zero mean, Gaussian random variable w_j is responsible for the effects of oscillator instability and deceptive jitter and it is added to the data at the emitter. The independent, zero mean, Gaussian random variable v_j is responsible for the effect of measurement noise and it is added to the data at the receiver. To simulate the effect of missing pulses, different data sets were generated with or without missing pulses. In the simulated data sets, the occurrence of a single missing pulse is represented by a large PRI value. The large value is the sum of the missing PRI value and the following PRI value.

3.2.1 Period detection for Staggered PRI sequence

An algorithm is needed to determine the period (the number of pulses per period) of the staggered PRI sequence, so we can obtain the PRI values in one period of staggered PRI sequence. We assumed that a set of PRI measurements is available to determine the period. Our aim is to find the number of pulses per period equivalently the position number of the staggered PRI sequence in one complete period.

If the number of pulses per period is *N,* every PRI in a data set will repeat after *N* PRI values if there is no missing pulse. The repetition number *N* for each PRI can change according to the number of missing pulses and existence of the same PRI, because the same PRI can exist in the data set more than once in a period. Algorithm starts successively at each PRI in the measured data set and estimates are obtained by searching forward for similar PRI values in the measured data set. While doing this, algorithm takes into account the effects of jitter (see Equations 2.1 and 2.2) on PRI values, so forward searching for similar PRI values is done by opening a gate according to the jitter levels and searching the similar PRI is done in this gate for each PRI value. Let us define the gate for each PRI value;

The variance of the measured PRI is found to be as (see Equation 3.37):

Variance of the measured PRI= $\sigma_w^2 + 2\sigma_v^2$

Where σ_w^2 and σ_v^2 are the variances of the system noise w_k and measurement noise v_k respectively. So for each measured PRI value, a gate is opened to find the same PRI values, the similar PRI value must be the -/+ 4 standard deviations of the referenced PRI value. If this is true, an estimate of *N* is obtained according to index of the pulses. Finally, similar estimates of *N* are grouped together in bins. The bin with the maximum number of estimates contains the correct value of *N.* Figure 9 shows this search procedure.

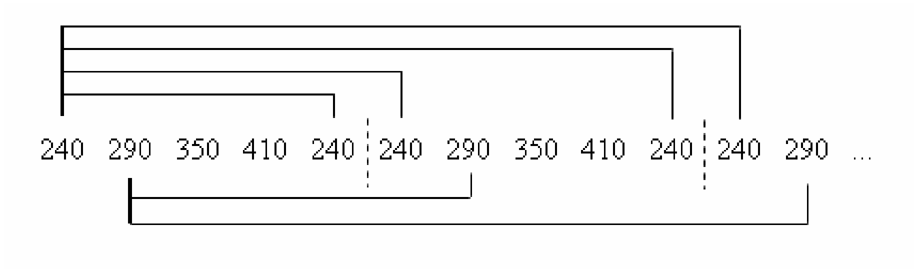

Figure 9. Search Procedure for Possible Periods

3.2.1.1 Detection of period in the case of no missing pulse

Let us assume that the actual (true) staggered PRI sequence is given as in Figure 9 and the measured PRI values (total number of measured PRI values is 20, sigma_w=0.6 and sigma_v=1) of this sequence are given as follows:

241.3275 288.5622 348.7725 409.6608 240.6697 239.8278 290.7683 347.7291 408.2143 240.2345 240.8556 291.2208 350.0216 409.3564 242.2221 239.6129 289.9353 351.2272 409.2255 239.3145

Figure 10 shows the results of search procedure applied to this staggered PRI sequence (5-position, 4-level staggered PRI sequence with no missing pulse) to detect the period of the sequence. In Figure 10, number of hits represents total repetition number of similar PRI values for each possible period. For example, no PRI value repeats after 2 PRI values, so number of hits for 2 is zero as shown in Figure 10. Also, 5 PRI values repeat after 15 PRI values, so number of hits for 15 is 5 as shown in Figure 10. According to the Figure 10, maximum number of hits is 15 so period of the staggered PRI sequence is found to be 5. Note that the integer multiples of period can be seen easily from Figure 10 (i.e. 10, 15 …). This is expected because any sequence with period *N* also has period *2N, 3N*, etc.

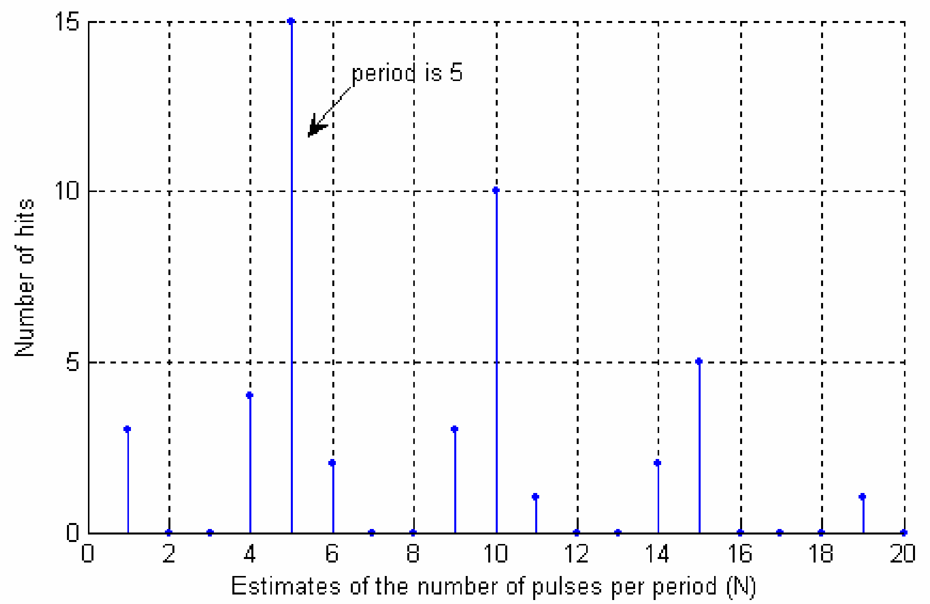

Figure 10. Period Detection for the Sequence with no Missing Pulse (5-Position, 4-Level Staggered PRI Sequence)

3.2.1.2 Detection of period in the case of missing pulse

Now, let us assume that the measurements for the pulses with indices 10, 12, 14 and 17 have been missed and the measured PRI values (total number of measured PRI values is 20, sigma_w=0.6 and sigma_v=1) of this sequence are given as follows:

241.3275 288.5622 348.7725 409.6608 240.6697 239.8278 290.7683 347.7291
408.2143 480.0582 639.9633 649.3091 239.6129 639.2652 409.2255 239.3145

Figure 11 shows the results of search procedure applied to this staggered PRI sequence (5-position, 4-level staggered PRI sequence with four missing pulses) to detect the period of the sequence. In Figure 11, number of hits represents total repetition number of similar PRI values for each possible period. For example, no PRI value repeats after 9 PRI values, so number of hits for 9 is zero as shown in Figure 11. According to the Figure 11, maximum number of hits is 4 so period of the staggered PRI sequence is found to be 5. The change in the number of hits due to missing pulses can be seen easily from Figure 10 and Figure 11.

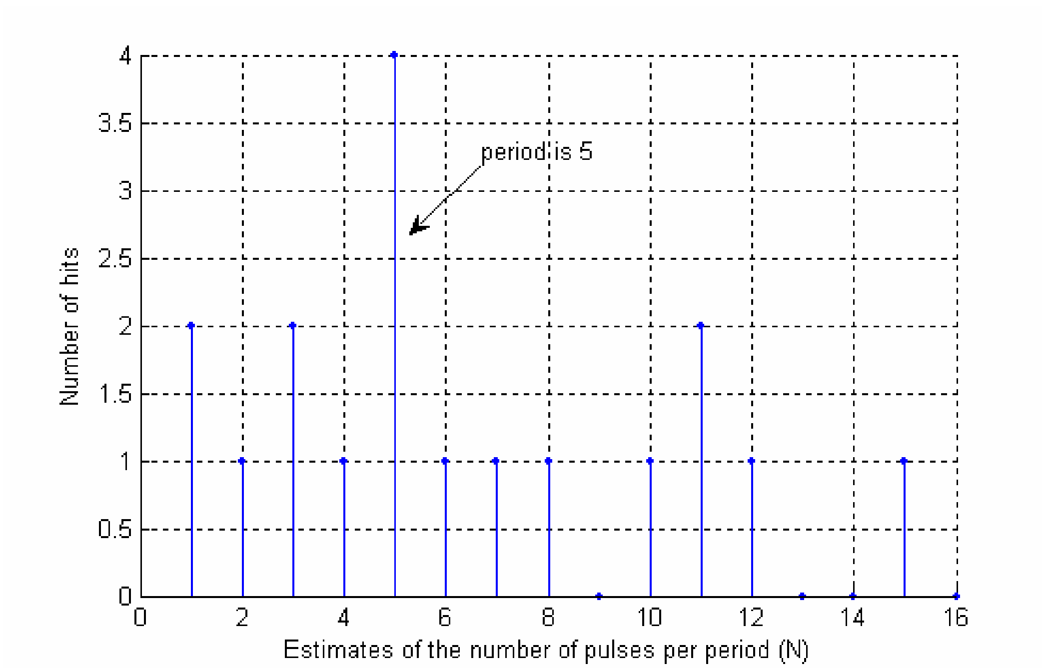

Figure 11. Period Detection for the Sequence with Missing Pulses (5-Position, 4-Level Staggered PRI Sequence, 4 missing pulses)

3.2.2 System and observation models for Staggered PRI sequence in time domain - (Algorithm I)

3.2.2.1 Staggered PRI sequence models in time domain in the case of no missing pulse

First, we have to find system and observation models for the staggered PRI sequence. The TOA values contain integrated system noise ($w_j \sim N(0, \sigma_w^2)$). If we use cumulative model [10, 13] for a staggered PRI sequence with a period of N, the TOA values can be written as:

$$TOA_j = TOA_{j-1} + PRI_j + w_j \qquad w_j \sim N(0, \sigma_w^2)$$

$$TOA_1 = TOA_0 + PRI_1 + w_1$$

$$TOA_2 = TOA_1 + PRI_2 + w_2$$

$$\vdots$$

$$TOA_N = TOA_{N-1} + PRI_N + w_N$$

$$TOA_{N+1} = TOA_N + PRI_1 + w_{N+1} \quad \text{(Because, } PRI_i = PRI_{i+N} \text{ and } N \text{ is the period.)}$$

$$TOA_{N+2} = TOA_{N+1} + PRI_2 + w_{N+2}$$

$$\vdots$$

Equivalently,

$$TOA_1 = TOA_0 + PRI_1 + w_1$$

$$TOA_2 = TOA_0 + PRI_1 + PRI_2 + w_1 + w_2$$

$$TOA_3 = TOA_0 + PRI_1 + PRI_2 + PRI_3 + w_1 + w_2 + w_3$$

$$TOA_4 = TOA_0 + PRI_1 + PRI_2 + PRI_3 + PRI_4 + w_1 + w_2 + w_3 + w_4$$

$$\vdots$$

$$TOA_{N-1} = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_{N-1} + w_1 + w_2 + \ldots + w_{N-1}$$

$$TOA_N = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_{N-1} + PRI_N + w_1 + w_2 + \ldots + w_{N-1} + w_N$$

$$TOA_{N+1} = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_{N-1} + PRI_N + PRI_1 + w_1 + w_2 \ldots + w_{N+1}$$

$$TOA_{N+2} = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_N + PRI_1 + PRI_2 + w_1 + w_2 + \ldots + w_{N+2}$$

$$\vdots$$

$$TOA_n = TOA_0 + \sum_{i=1}^{n} PRI_i + \sum_{i=1}^{n} w_i \,. \tag{3.35}$$

In Equation 3.35, for a staggered PRI sequence with period *N*, $PRI_i = PRI_{i+N}$.

If an observation is done at time *t* in a Kalman filter, it contains system noise which is integrated from time zero to time t

So, the measured TOA values (starting at TOA_0 =0) τ_j at the receiver are in the form of:

$$\tau_j = TOA_j + v_j \qquad v_j \sim N(0, \sigma_v^2),$$

$$\tau_1 = TOA_1 + v_1 = PRI_1 + w_1 + v_1$$

$$\tau_2 = TOA_2 + v_2 \;=\; PRI_1 + PRI_2 + w_1 + w_2 + v_2$$

$$\tau_3 = TOA_3 + v_3 \;=\; PRI_1 + PRI_2 + PRI_3 + w_1 + w_2 + w_3 + v_3$$

$$\vdots$$

$$\tau_N = TOA_{N-1} + v_N \;=\; PRI_1 + PRI_2 + \ldots + PRI_N + w_1 + w_2 + \ldots + w_N + v_N$$

$$\tau_{N+1} = TOA_N + v_{N+1} = 2PRI_1 + PRI_2 + \ldots + PRI_N + w_1 + w_2 + \ldots + w_{N+1} + v_{N+1}$$

$$\vdots$$

$$\tau_n = \sum_{i=1}^{n} PRI_i + \sum_{i=1}^{n} w_i + v_n . \tag{3.36}$$

Since staggered PRI sequence is periodic with N, so PRI values can be written as: $PRI_i = PRI_{i+N}$.

In this application, our aim is to find the PRI values rather than TOAs. PRIs are computed as the difference between successive TOAs:

measured_PRI_n = measured_TOA_n - measured_TOA_{n-1}

$$= PRI_n + w_n + v_n - v_{n-1}$$

Var(measured_PRI_n)= $\sigma_w^2 + 2\sigma_v^2$. (3.37)

Now, let's rewrite the equations 2.1 and 2.2 with white Gaussian random noises $w_k \sim N(0, \sigma_w^2)$ and $v_k \sim N(0, \sigma_v^2)$:

System equation : $X_k = X_{k-1}$ (3.38)

Observation equation: $Y_k = H_k X_k + u_k$ (3.39)

Where,

X_k is the state vector,

k is the period index,

H_k is the measurement matrix,

Y_k is the measured TOA vector,

u_k is the measurement noise vector defined by $\begin{bmatrix} \sum_{i=1}^{Nk-(N-1)} w_i + v_{Nk-(N-1)} \\ \sum_{i=1}^{Nk-(N-2)} w_i + v_{Nk-(N-2)} \\ \vdots \\ \sum_{i=1}^{Nk} w_i + v_{Nk} \end{bmatrix}$

where N is the period of the periodic staggered sequence.

X_k, Y_k and H_k are defined as below:

The period of the PRI sequence determines the length of the sate vector. So, a N-position stagger sequence requires a Kalman filter of order N. The state vector contains the PRI values of one period is written as:

$X_k^T = [PRI_1\ PRI_2 \ldots PRI_N]$.

N is the period of the sequence or equivalently the number of pulses per period.

The length of the measured TOA vector Y_k depends on the period of the PRI sequence. Again for a sequence with period N and no missing pulses:

For k=1, $Y_1^T = [\tau_1 \quad \tau_2 \quad \ldots \quad \tau_N]$

For k=2, $Y_2^T = [\tau_1 \quad \tau_2 \quad \ldots \quad \tau_N]$

$\vdots$

So, $Y_k^T = [\tau_{(k-1)N+1} \quad \tau_{(k-1)N+2} \quad \ldots \quad \tau_{(k-1)N+N}]$

The value and the length of the measurement matrix H_k depends on the period of the sequence N and the period index k. From the TOA equations and measured TOA equations with no missing pulses:

For k=1, $$H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots & 0 \\ \vdots & & & & & \\ 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}_{NxN}$$

For k=2, $$H_2 = \begin{bmatrix} 2 & 1 & 1 & 1 & \dots & 1 \\ 2 & 2 & 1 & 1 & \dots & 1 \\ 2 & 2 & 2 & 1 & \dots & 1 \\ \vdots & & & & & \\ 2 & 2 & 2 & 2 & \dots & 2 \end{bmatrix}_{NxN}$$

$\vdots$

So, $$H_k = \begin{bmatrix} k & k-1 & k-1 & k-1 & \dots & k-1 \\ k & k & k-1 & k-1 & \dots & k-1 \\ k & k & k & k-1 & \dots & k-1 \\ \vdots & & & & & \\ k & k & k & k & \dots & k \end{bmatrix}_{NxN}$$

Now, let us write the system and observation equations in matrix form, for simplicity a 3-position staggered PRI sequence with no missing pulse will be used while deriving the following equations:

From Equation 3.38, system equation is written as $X_k = X_{k-1} = \begin{bmatrix} PRI_1 \\ PRI_2 \\ PRI_3 \end{bmatrix}$.

So for the first period of the data set, the observation equation is written as:

$$\underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}}_{Y_1} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{H_1} \underbrace{\begin{bmatrix} PRI_1 \\ PRI_2 \\ PRI_3 \end{bmatrix}}_{X_1} + \begin{bmatrix} w_1 + v_1 \\ w_1 + w_2 + v_2 \\ w_1 + w_2 + w_3 + v_3 \end{bmatrix}.$$

For the second period, the observation equation is written as;

$$\underbrace{\begin{bmatrix}\tau_4\\ \tau_5\\ \tau_6\end{bmatrix}}_{Y_2}=\underbrace{\begin{bmatrix}2&1&1\\ 2&2&1\\ 2&2&2\end{bmatrix}}_{H_2}\underbrace{\begin{bmatrix}PRI_1\\ PRI_2\\ PRI_3\end{bmatrix}}_{X_2}+\begin{bmatrix}w_1+w_2+w_3+w_4+v_4\\ w_1+w_2+w_3+w_4+w_5+v_5\\ w_1+w_2+w_3+w_4+w_5+w_6+v_6\end{bmatrix} \tag{3.40}$$

Similarly, for the third period, the observation equation is written as:

$$\underbrace{\begin{bmatrix}\tau_7\\ \tau_8\\ \tau_9\end{bmatrix}}_{Y_3}=\underbrace{\begin{bmatrix}3&2&2\\ 3&3&2\\ 3&3&3\end{bmatrix}}_{H_3}\underbrace{\begin{bmatrix}PRI_1\\ PRI_2\\ PRI_3\end{bmatrix}}_{X_3}+\begin{bmatrix}w_1+w_2+w_3+w_4+w_5+w_6+w_7+v_7\\ w_1+w_2+w_3+w_4+w_5+w_6+w_7+w_8+v_8\\ w_1+w_2+w_3+w_5+w_6+w_7+w_8+w_9+v_9\end{bmatrix}.$$

$$\vdots$$

So, for step k, the observation equation can be written as:

$$Y_k=\underbrace{\begin{bmatrix}k&k-1&k-1\\ k&k&k-1\\ k&k&k\end{bmatrix}}_{H_k}\underbrace{\begin{bmatrix}PRI_1\\ PRI_2\\ PRI_3\end{bmatrix}}_{X_k}+\underbrace{\begin{bmatrix}\sum_{i=1}^{3k-2}w_i+v_{3k-2}\\ \sum_{i=1}^{3k-1}w_i+v_{3k-1}\\ \sum_{i=1}^{3k}w_i+v_{3k}\end{bmatrix}}_{u(k)}$$

$E\{m(k)m(k)'\}=R_k$, where R_k is called as the measurement covariance matrix calculated as follows:

For k=1, $R_1=\begin{bmatrix}\sigma_w^2+\sigma_v^2&\sigma_w^2&\sigma_w^2\\ \sigma_w^2&2\sigma_w^2+\sigma_v^2&2\sigma_w^2\\ \sigma_w^2&2\sigma_w^2&3\sigma_w^2+\sigma_v^2\end{bmatrix}$

For k=2, $R_2=\begin{bmatrix}4\sigma_w^2+\sigma_v^2&4\sigma_w^2&4\sigma_w^2\\ 4\sigma_w^2&5\sigma_w^2+\sigma_v^2&5\sigma_w^2\\ 4\sigma_w^2&5\sigma_w^2&6\sigma_w^2+\sigma_v^2\end{bmatrix}$

For k=3, $R_3 = \begin{bmatrix} 7\sigma_w^2 + \sigma_v^2 & 7\sigma_w^2 & 7\sigma_w^2 \\ 7\sigma_w^2 & 8\sigma_w^2 + \sigma_v^2 & 8\sigma_w^2 \\ 7\sigma_w^2 & 8\sigma_w^2 & 9\sigma_w^2 + \sigma_v^2 \end{bmatrix}$

$$\vdots$$

So general form of the measurement covariance matrix R_k related to staggered sequence with a period of 3 is given by:

$$R_k = \begin{bmatrix} (3k-2)\sigma_w^2 + \sigma_v^2 & (3k-2)\sigma_w^2 & (3k-2)\sigma_w^2 \\ (3k-2)\sigma_w^2 & (3k-1)\sigma_w^2 + \sigma_v^2 & (3k-1)\sigma_w^2 \\ (3k-2)\sigma_w^2 & (3k-1)\sigma_w^2 & (3k)\sigma_w^2 + \sigma_v^2 \end{bmatrix} \quad \textit{k: period index}$$

According to the result which has found above, the general form of the measurement covariance matrix R_k related to staggered sequence with a period of N is given by:

$$R_k = \begin{bmatrix} (Nk-N+1)\sigma_w^2 + \sigma_v^2 & (Nk-N+1)\sigma_w^2 & \dots & (Nk-N+1)\sigma_w^2 \\ (Nk-N+1)\sigma_w^2 & (Nk-N+2)\sigma_w^2 + \sigma_v^2 & \dots & (Nk-N+2)\sigma_w^2 \\ \vdots & \vdots & & \vdots \\ (Nk-N+1)\sigma_w^2 & (Nk-N+2)\sigma_w^2 & \dots & (Nk)\sigma_w^2 + +\sigma_v^2 \end{bmatrix}_{NxN}$$

Let us find the initial values for state vector X_k and error covariance P_k by using Equation 3.36:

$$\hat{X}_0(1) = Y_1 - Y_0 = PRI_1 + w_1 + v_1 - v_0$$

$$\hat{X}_0(2) = Y_2 - Y_1 = PRI_2 + w_2 + v_2 - v_1$$

$$\hat{X}_0(3) = Y_3 - Y_2 = PRI_3 + w_3 + v_3 - v_2$$

So, initial value of the state vector is written as:

$$\hat{X}_0 = \begin{bmatrix} PRI_1 + w_1 + v_1 - v_0 \\ PRI_2 + w_2 + v_2 - v_1 \\ PRI_3 + w_3 + v_3 - v_2 \end{bmatrix}.$$

The state prediction error can be found as:

$$X_0 - \hat{X}_0 = \begin{bmatrix} PRI_1 \\ PRI_2 \\ PRI_3 \end{bmatrix} - \begin{bmatrix} PRI_1 + w_1 + v_1 - v_0 \\ PRI_2 + w_2 + v_2 - v_1 \\ PRI_3 + w_3 + v_3 - v_2 \end{bmatrix} = \begin{bmatrix} -w_1 - v_1 + v_0 \\ -w_2 - v_2 + v_1 \\ -w_3 - v_3 + v_2 \end{bmatrix}$$

So, the initial value of the state prediction error covariance matrix is:

$$P_0 = \begin{bmatrix} \sigma_w^2 + 2\sigma_v^2 & -\sigma_v^2 & 0 \\ -\sigma_v^2 & \sigma_w^2 + 2\sigma_v^2 & -\sigma_v^2 \\ 0 & -\sigma_v^2 & \sigma_w^2 + 2\sigma_v^2 \end{bmatrix}.$$

3.2.2.2 Staggered PRI sequence models in time domain in the case of missing pulse

If there are missing pulses in the data set (measured TOA values), then the measurement matrix H_k and the noise vector u_k change. For example, if the 5th pulse is missing in the second period (again the same sequence as in part 3.2.2.1 is used for simplicity), then the matrix model of the observation equation (equation 3.40) will be the following:

$$\begin{bmatrix} \tau_4 \\ \tau_5 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} PRI_1 \\ PRI_2 \\ PRI_3 \end{bmatrix} + \begin{bmatrix} w_1 + w_2 + w_3 + w_4 + v_4 \\ w_1 + w_2 + w_3 + w_5 + w_6 + v_5 \end{bmatrix}$$

In the equations above, w_k represents white Gaussian cumulative TOA jitter and v_k represents white Gaussian non-cumulative jitter. Now, let us find the general form of the observation equation in the case of missing pulse.

We have found general form of the observation equation in the case of no missing pulse as:

$$\underbrace{\begin{bmatrix} \tau_{(k-1)N+1} \\ \tau_{(k-1)N+2} \\ \tau_{(k-1)N+3} \\ \vdots \\ \tau_{(k-1)N+N} \end{bmatrix}}_{Y_k} = \underbrace{\begin{bmatrix} k & k-1 & k-1 & k-1 & \dots & k-1 \\ k & k & k-1 & k-1 & \dots & k-1 \\ k & k & k & k-1 & \dots & k-1 \\ \vdots & & & & & \\ k & k & k & k & \dots & k \end{bmatrix}}_{H_k}{}_{NxN} \underbrace{\begin{bmatrix} PRI_1 \\ PRI_2 \\ PRI_3 \\ \vdots \\ PRI_N \end{bmatrix}}_{X_k} + \underbrace{\begin{bmatrix} \sum_{i=1}^{Nk-(N-1)} w_i + v_{Nk-(N-1)} \\ \sum_{i=1}^{Nk-(N-2)} w_i + v_{Nk-(N-2)} \\ \sum_{i=1}^{Nk-(N-3)} w_i + v_{Nk-(N-3)} \\ \vdots \\ \sum_{i=1}^{Nk} w_i + v_{Nk} \end{bmatrix}}_{u_k}$$

If there is a missing pulse, then the rows of the measurement vector Y_k, measurement matrix H_k and measurement noise vector u_k are deleted according to the index of the missing pulse. For example, if the pulse with index of "(k-1)N+2" is missing then the observation equation is rewritten as:

$$\underbrace{\begin{bmatrix} \tau_{(k-1)N+1} \\ \tau_{(k-1)N+3} \\ \vdots \\ \tau_{(k-1)N+N} \end{bmatrix}}_{Y_k} = \underbrace{\begin{bmatrix} k & k-1 & k-1 & k-1 & \cdots & k-1 \\ k & k & k & k-1 & \cdots & k-1 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\ k & k & k & k & k & k \end{bmatrix}}_{H_k}{}_{NxN} \underbrace{\begin{bmatrix} PRI_1 \\ PRI_2 \\ PRI_3 \\ \vdots \\ PRI_N \end{bmatrix}}_{X_k} + \underbrace{\begin{bmatrix} \sum_{i=1}^{Nk-(N-1)} w_i + v_{Nk-(N-1)} \\ \sum_{i=1}^{Nk-(N-3)} w_i + v_{Nk-(N-3)} \\ \vdots \\ \sum_{i=1}^{Nk} w_i + v_{Nk} \end{bmatrix}}_{u_k}$$

Missing pulses results in PRIs that are much larger than the actual PRIs in the sequence. To find the missing pulses in the sequence, a gate is opened for each measured TOA value. If the standard deviations of the uncorrelated white Gaussian noises are known, then taking +/-3 standard deviation includes %99,74 of the Gaussian distribution. So from the Equation 3.36, measured TOA τ_n has jitter value of $\sum_{i=1}^{n} w_i + v_n$ with a variance of "$n\sigma_w^2 + \sigma_v^2$". Also at each iteration, Kalman filter makes a prediction with an associated variance. As a result the gate can be opened for each measured TOA value.

For example let's open the gate (including %99, 74 of the distribution) for measured TOA value "τ_2" which is equal to the $PRI_1 + PRI_2 + w_1 + w_2 + v_2$:

$$PRI_1 + PRI_2 - 3\sqrt{2\sigma_w^2 + \sigma_v^2} \le \tau_2 \le PRI_1 + PRI_2 + 3\sqrt{2\sigma_w^2 + \sigma_v^2}$$

3.2.3 System and observation models for Staggered-PRI sequence with discrete Fourier transform (DFT) - (Algorithm II)

3.2.3.1 Staggered PRI sequence models with DFT in the case of no missing pulse

First, we have to find system and observation models for the staggered PRI sequence. It is important to say that a staggered PRI sequence can be represented by a discrete time series. A staggered PRI sequence has a period, and each period contains an integer number of pulses. So, we can say that the sequence repeats in a deterministic manner.

Any periodic function can be expressed as a linear combination of sine and cosine terms plus a constant according to the Fourier representation theorem [4, 15]. Actually, the frequencies of the sine and cosine terms represent different harmonics. A discrete time series can be represented by a finite number of harmonics. If a discrete time series has a period of *N,* then the first harmonic has the frequency of 1/*N.* The second harmonic has the frequency of 2/*N.* If *N* is even, at most *N*/2 harmonics are needed to represent the discrete time series because the period corresponding to the (*N*/2)th harmonic is 2, which is the shortest possible cycle length. In the other hand, if *N* is odd, at most (*N*-1)/2 harmonics are required [16]. Let us show why this is true:

First of all, Fourier series representation of a periodic discrete time sequence $\tilde{x}[n]$ can be written as follows:

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\left(\frac{2\pi}{N}\right)kn} \tag{3.41}$$

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}[k]e^{j\left(\frac{2\pi}{N}\right)kn} \tag{3.42}$$

Where N is the period of the sequence $\tilde{x}[n]$ and $\tilde{X}[k]$ is the Fourier coefficients of the sequence $\tilde{x}[n]$.

So, for a N-position staggered PRI sequence, Discrete Fourier Transform (DFT) equations can be written as:

$$PRI_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} C_k e^{j\left(\frac{2\pi}{N}\right)kn}, \qquad n = 0,1,\ldots,N-1$$

From the equation above it is found that $PRI_n = PRI_{n+N}$. In other words, Discrete Fourier Transform can be considered as Fourier series representation for periodic sequences. The coefficients C_k 's are found as:

$$C_k = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} PRI_n e^{-j\left(\frac{2\pi}{N}\right)kn}, \qquad k = 0,1,\ldots,N-1$$

Also we know that a staggered PRI sequence has only real values. So the complex DFT coefficients are defined as:

$$C_{N-k}^{*} = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} PRI_n e^{j\left(\frac{2\pi}{N}\right)(N-k)n} = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} PRI_n e^{-j\left(\frac{2\pi}{N}\right)kn} = C_k$$

In other words, complex DFT coefficients for a real valued periodic staggered sequence have symmetry, this symmetry can be written as:

$$C_{N-k}^{*} = C_k \qquad k = 1,2,\ldots,\frac{N-1}{2}. \tag{3.43}$$

What happens if <u>N is even</u>? Now, let us assume that N is even:

$$PRI_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} C_k e^{j\left(\frac{2\pi}{N}\right)kn}, \qquad n = 0,1,\ldots,N-1$$

$$= \frac{1}{\sqrt{N}}\left(C_0 + C_{\frac{N}{2}}e^{j(\pi)n} + \sum_{k=1}^{\frac{N-2}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=\frac{N+2}{2}}^{N-1} C_k e^{j\left(\frac{2\pi}{N}\right)kn}\right), \qquad n = 0,1,\ldots,N-1$$

Now, if *N-k* is substituted for *k* in second summation ($k \rightarrow N-k$):

$$= \frac{1}{\sqrt{N}}\left(C_0 + C_{\frac{N}{2}} e^{j(\pi)n} + \sum_{k=1}^{\frac{N-2}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=1}^{\frac{N-2}{2}} C_{N-k} e^{j\left(\frac{2\pi}{N}\right)(N-k)n} \right)$$

$$= \frac{1}{\sqrt{N}}\left(C_0 + C_{\frac{N}{2}} e^{j(\pi)n} + \sum_{k=1}^{\frac{N-2}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=1}^{\frac{N-2}{2}} C_{N-k} e^{-j\left(\frac{2\pi}{N}\right)kn} \right)$$

It is found that $C_{N-k}^{*} = C_k \ or \ C_{N-k} = C_k^{*} \ for \quad k = 1,2,...,\frac{N-1}{2}$ (equation 3.43). If this result is substituted into the Equation 3.43:

$$= \frac{1}{\sqrt{N}}\left(C_0 + C_{\frac{N}{2}} e^{j(\pi)n} + \sum_{k=1}^{\frac{N-2}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=1}^{\frac{N-2}{2}} C_k^{*} e^{-j\left(\frac{2\pi}{N}\right)kn} \right).$$

$$= \frac{1}{\sqrt{N}}\left(C_0 + C_{\frac{N}{2}} e^{j(\pi)n} + \sum_{k=1}^{\frac{N-2}{2}} \left(C_k e^{j\left(\frac{2\pi}{N}\right)kn} + C_k^{*} e^{-j\left(\frac{2\pi}{N}\right)kn} \right) \right) \tag{3.44}$$

I

n Equation 3.44, the value of the coefficient $C_{\frac{N}{2}}$ can be found as below:

$$C_{\frac{N}{2}} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} PRI_n e^{-j(\pi)n} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} PRI_n \cos(\pi n)$$

From the equation above, it is seen that the imaginary part of $C_{\frac{N}{2}}$ is zero as C_0.

If DFT coefficients C_k 's are written in the form of $C_k \stackrel{\Delta}{=} \alpha_k + j\beta_k$:

$$PRI_n = \frac{1}{\sqrt{N}}\left(C_0 + \alpha_{\frac{N}{2}} \cos(\pi n) + \sum_{k=1}^{\frac{N-2}{2}} \left(2\alpha_k \cos\left(\frac{2\pi kn}{N}\right) - 2\beta_k \sin\left(\frac{2\pi kn}{N}\right) \right) \right) \tag{3.45}$$

What happens if <u>*N* is odd</u>? Now, let us assume that *N* is odd:

$$PRI_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} C_k e^{j\left(\frac{2\pi}{N}\right)kn}, \qquad n = 0,1,...,N-1$$

$$= \frac{1}{\sqrt{N}} \left(C_0 + \sum_{k=1}^{\frac{N-1}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=\frac{N+1}{2}}^{N-1} C_k e^{j\left(\frac{2\pi}{N}\right)kn} \right), \qquad n = 0,1,\ldots,N-1$$

As we did for even *N*, *N-k* is substituted for *k* in second summation ($k \rightarrow N-k$):

$$= \frac{1}{\sqrt{N}} \left(C_0 + \sum_{k=1}^{\frac{N-1}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=1}^{\frac{N-1}{2}} C_{N-k} e^{j\left(\frac{2\pi}{N}\right)(N-k)n} \right), \qquad n = 0,1,\ldots,N-1$$

$$= \frac{1}{\sqrt{N}} \left(C_0 + \sum_{k=1}^{\frac{N-1}{2}} C_k e^{j\left(\frac{2\pi}{N}\right)kn} + \sum_{k=1}^{\frac{N-1}{2}} C_{N-k} e^{-j\left(\frac{2\pi}{N}\right)kn} \right)$$

$$= \frac{1}{\sqrt{N}} \left(C_0 + \sum_{k=1}^{\frac{N-1}{2}} \left(C_k e^{j\left(\frac{2\pi}{N}\right)kn} + C_k^* e^{-j\left(\frac{2\pi}{N}\right)kn} \right) \right)$$

If we substitute the result $C_k = \alpha_k + j\beta_k$ into the equation above, PRI_n is found as:

$$PRI_n = \frac{1}{\sqrt{N}} \left(C_0 + \sum_{k=1}^{\frac{N-1}{2}} \left(2\alpha_k \cos\left(\frac{2\pi kn}{N}\right) - 2\beta_k \sin\left(\frac{2\pi kn}{N}\right) \right) \right). \qquad (3.46)$$

From the Equation 3.46, following results can be found:

$$\overline{PRI} = \frac{1}{\sqrt{N}} C_0 = \frac{1}{N} \sum_{n=0}^{N-1} PRI_n \qquad \overline{PRI} : \textit{mean value of staggered PRI sequence}$$

$$C_m = \alpha_m + j\beta_m \qquad m = 1,2,\ldots,\left[\frac{N-1}{2}\right].$$

$$a_m = \frac{2}{\sqrt{N}} \operatorname{Re}\{C_m\} = \frac{2}{\sqrt{N}} \alpha_m = \frac{2}{N} \sum_{n=0}^{N-1} PRI_n \cos\left(\frac{2\pi mn}{N}\right)$$

$$b_m = -\frac{2}{\sqrt{N}} \operatorname{Im}\{C_m\} = -\frac{2}{\sqrt{N}} \beta_m = \frac{2}{N} \sum_{n=0}^{N-1} PRI_n \sin\left(\frac{2\pi mn}{N}\right)$$

If *N* is <u>even</u>:

$$a_{\frac{N}{2}} = \frac{1}{\sqrt{N}} \operatorname{Re}\left\{C_{\frac{N}{2}}\right\} = \frac{1}{\sqrt{N}} \alpha_{\frac{N}{2}} = \frac{1}{N} \sum_{n=0}^{N-1} PRI_n \cos(\pi n)$$

$b_{\frac{N}{2}} = 0$ Imaginary part is zero because $\sin(\pi n) = 0$ *for all n.*

So, we obtain the following result [4]:

$$m = \begin{cases} N/2 & for\ N\ even \\ (N-1)/2 & for\ N\ odd. \end{cases} \tag{3.47}$$

Now, let us remember TOA equations of periodic staggered PRI sequence:

$TOA_j = TOA_{j-1} + PRI_j + w_j \qquad w_j \sim N(0, \sigma_w^2)$ White Gaussian system noise

$TOA_1 = TOA_0 + PRI_1 + w_1$

$TOA_2 = TOA_1 + PRI_2 + w_2$

$\vdots$

$TOA_N = TOA_{N-1} + PRI_N + w_N$

$TOA_{N+1} = TOA_N + PRI_1 + w_{N+1}$ (Because, $PRI_i = PRI_{i+N}$ and N is the period.)

$TOA_{N+2} = TOA_{N+1} + PRI_2 + w_{N+2}$

$\vdots$

Equivalently,

$TOA_1 = TOA_0 + PRI_1 + w_1$

$TOA_2 = TOA_0 + PRI_1 + PRI_2 + w_1 + w_2$

$TOA_3 = TOA_0 + PRI_1 + PRI_2 + PRI_3 + w_1 + w_2 + w_3$

$TOA_4 = TOA_0 + PRI_1 + PRI_2 + PRI_3 + PRI_4 + w_1 + w_2 + w_3 + w_4$

$\vdots$

$TOA_{N-1} = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_{N-1} + w_1 + w_2 + \ldots + w_{N-1}$

$TOA_N = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_{N-1} + PRI_N + w_1 + w_2 + \ldots + w_{N-1} + w_N$

$TOA_{N+1} = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_{N-1} + PRI_N + PRI_1 + w_1 + w_2 \ldots + w_{N+1}$

$TOA_{N+2} = TOA_0 + PRI_1 + PRI_2 + \ldots + PRI_N + PRI_1 + PRI_2 + w_1 + w_2 + \ldots + w_{N+2}$

So, $$TOA_n = TOA_0 + \sum_{i=1}^{n} PRI_i + \sum_{i=1}^{n} w_i\ . \tag{3.48}$$

And let us write the measured TOA values with a non-cumulative white Gaussian observation noise of $v_j \sim N(0, \sigma_v^2)$ and using Equation 3.48.

So, the measured TOA values (starting at TOA_0 =0) τ_j at the receiver are in the form of:

$$\tau_1 = TOA_1 + v_1 \;=\; PRI_1 + w_1 + v_1$$

$$\tau_2 = TOA_2 + v_2 \;=\; PRI_1 + PRI_2 + w_1 + w_2 + v_2$$

$$\tau_3 = TOA_3 + v_3 \;=\; PRI_1 + PRI_2 + PRI_3 + w_1 + w_2 + w_3 + v_3$$

$$\vdots$$

$$\tau_n = TOA_n + v_n = PRI_1 + PRI_2 + \ldots + PRI_n + w_1 + w_2 + \ldots + w_n + v_n$$

$$\vdots$$

$$\tau_n = \sum_{i=1}^{n} PRI_i + \sum_{i=1}^{n} w_i + v_n . \tag{3.49}$$

Measured PRI values (Y_n) can be found using Equation 3.49 as below:

$$Y_n = \tau_n - \tau_{n-1} = \sum_{i=1}^{n} PRI_i + \sum_{i=1}^{n} w_i + v_n - \sum_{i=1}^{n-1} PRI_i + \sum_{i=1}^{n-1} w_i + v_{n-1}$$
$$= PRI_n + w_n + v_n - v_{n-1} \tag{3.50}$$

If the PRI value is represented in DFT form, following observation equation is obtained [4]:

$$Y_n = H_n X_n + w_n + v_n - v_{n-1} \tag{3.51}$$

Where H_n is the measurement matrix and described as:

$$H_n = \left[1 \quad \cos\left(2\pi \frac{1}{N} n\right) \quad \sin\left(2\pi \frac{1}{N} n\right) \quad \ldots \quad \cos\left(2\pi \frac{m}{N} n\right) \quad \sin\left(2\pi \frac{m}{N} n\right)\right] \tag{3.52}$$

And X_n is the state vector described as:

$$X_n = \left[\overline{PRI} \quad a_1 \quad b_1 \quad a_2 \quad b_2 \quad \ldots \quad a_m \quad b_m\right]^T \tag{3.53}$$

Where, m is defined as in Equation 3.47.

And $w_n + v_n - v_{n-1}$ is the observation noise, if we say $u_n = w_n + v_n - v_{n-1}$ then covariance of u_n can be written as:

$E\left(u(n)u(n)'\right) = R_n = \sigma_w^2 + 2\sigma_v^2$ with a mean of zero.

State space equation can be written as below:

$X_n = X_{n-1}$ Because, state is constant for a periodic staggered PRI sequence.

So, if the period of the staggered PRI sequence can be determined, then the following Kalman filter model can be used to predict PRIs:

$$\text{System equation:} \quad X_k = X_{k-1} \tag{3.54}$$

$$\text{Observation equation:} \quad Y_k = H_k X_k + u_k \qquad u_k \sim N\left(0, \sigma_u^2\right) \tag{3.55}$$

Where,

X_k is the state vector given by Equation 3.53,

H_k is the measurement vector given by Equation 3.52,

Y_k is the measured PRI value,

$u_k = w_k + v_k - v_{k-1}$ is the measurement noise with a mean of zero and variance of $\sigma_u^2 = \sigma_w^2 + 2\sigma_v^2$.

From the model described above, we can write the Kalman equations as follows:

The covariance of the process noise is $Q_k = 0$ (state is constant).

Measurement covariance matrix is $R_k = E\left(u_k u_k{}^T\right) = \sigma_w^2 + 2\sigma_v^2$

$\hat{X}_k^- = \hat{X}_{k-1}$ $\quad \hat{X}_k^-$: predicted state for k at k-1.

$\hat{P}_k^- = P_{k-1}$ $\quad \hat{P}_k^-$: State prediction error covariance

$S_k = H_k \hat{P}_k^- H_k{}^T + \sigma_w^2 + 2\sigma_v^2$ $\quad S_k$: Innovation covariance

$K_k = \hat{P}_k^- H_k^T S_k{}^{-1} = \dfrac{\hat{P}_k^- H_k^T}{H_k \hat{P}_k^- H_k{}^T + \sigma_w^2 + 2\sigma_v^2}$ $\quad K_k$: Kalman filter gain

$\hat{X}_k = \hat{X}_k^- + \dfrac{\hat{P}_k^- H_k^T}{H_k \hat{P}_k^- H_k{}^T + \sigma_w^2 + 2\sigma_v^2}\left\{Y_k - H_k \hat{X}_k^-\right\}$ $\quad \hat{X}_k$: Updated state estimate

So, $\hat{X}_k = \hat{X}_k^- + \{Y_k - H_k \hat{X}_k^-\}$

$\hat{P}_k = \hat{P}_k^- - K_k S_k K_k^T \quad \hat{P}_k$: Updated state covariance

So, $\hat{P}_k = \hat{P}_k^- \left(I - \left(\dfrac{H_k^T H_k \hat{P}_k^{-T}}{H_k \hat{P}_k^- H_k^T + \sigma_w^2 + 2\sigma_v^2} \right) \right)$

To find the initial value of the state vector X, periodic staggered PRI sequence is represented in DFT form and then mean PRI value "$\overline{PRI}$", the coefficients a_m and b_m are obtained.

If period of staggered sequence is <u>odd</u>:

$$\overline{PRI} = \frac{1}{N}\sum_{n=0}^{N-1} PRI_n \quad \overline{PRI} : mean\ value\ of\ staggered\ PRI\ sequence\ with\ period\ N.$$

$$a_m = \frac{2}{N}\sum_{n=0}^{N-1} PRI_n \cos\left(\frac{2\pi mn}{N}\right)$$

$$b_m = \frac{2}{N}\sum_{n=0}^{N-1} PRI_n \sin\left(\frac{2\pi mn}{N}\right)$$

Let us write these results in vector form for initial state:

$$\underbrace{\begin{bmatrix} \overline{PRI} \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ \vdots \\ a_m \\ b_m \end{bmatrix}}_{State\ vector\ X} = \underbrace{\frac{1}{N}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2\cos\left(2\pi\frac{1}{N}\right) & \cdots & 2\cos\left(2\pi\frac{N-1}{N}\right) \\ 0 & 2\sin\left(2\pi\frac{1}{N}\right) & \cdots & 2\sin\left(2\pi\frac{N-1}{N}\right) \\ 2 & 2\cos\left(2\pi 2\frac{1}{N}\right) & \cdots & 2\cos\left(2\pi 2\frac{N-1}{N}\right) \\ 0 & 2\sin\left(2\pi 2\frac{1}{N}\right) & \cdots & 2\sin\left(2\pi 2\frac{N-1}{N}\right) \\ \vdots & \vdots & \cdots & \vdots \\ 2 & 2\cos\left(2\pi m\frac{1}{N}\right) & \cdots & 2\cos\left(2\pi m\frac{N-1}{N}\right) \\ 0 & 2\sin\left(2\pi m\frac{1}{N}\right) & \cdots & 2\sin\left(2\pi m\frac{N-1}{N}\right) \end{bmatrix}}_{\Omega} \underbrace{\begin{bmatrix} PRI_0 \\ PRI_1 \\ \vdots \\ PRI_{N-1} \end{bmatrix}}_{\lambda} \tag{3.56}$$

If period of staggered sequence is <u>even</u>:

$$a_{\frac{N}{2}} = \frac{1}{\sqrt{N}} \operatorname{Re}\left\{ C_{\frac{N}{2}} \right\} = \frac{1}{\sqrt{N}} \alpha_{\frac{N}{2}} = \frac{1}{N} \sum_{n=0}^{N-1} PRI_n \cos(\pi n)$$

$$b_{\frac{N}{2}} = 0$$

Again let us write this result in vector form for initial state (period is even):

$$\underbrace{\begin{bmatrix} \overline{PRI} \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ \vdots \\ a_m \\ b_{m-1} \end{bmatrix}}_{State\ vector\ X} = \underbrace{\frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2\cos\left(2\pi \frac{1}{N}\right) & \cdots & 2\cos\left(2\pi \frac{N-1}{N}\right) \\ 0 & 2\sin\left(2\pi \frac{1}{N}\right) & \cdots & 2\sin\left(2\pi \frac{N-1}{N}\right) \\ 2 & 2\cos\left(2\pi 2 \frac{1}{N}\right) & \cdots & 2\cos\left(2\pi 2 \frac{N-1}{N}\right) \\ 0 & 2\sin\left(2\pi 2 \frac{1}{N}\right) & \cdots & 2\sin\left(2\pi 2 \frac{N-1}{N}\right) \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 1\cos\left(2\pi m \frac{1}{N}\right) & \cdots & 1\cos\left(2\pi m \frac{N-1}{N}\right) \\ 0 & 2\sin\left(2\pi m \frac{1}{N}\right) & \cdots & 2\sin\left(2\pi m \frac{N-1}{N}\right) \end{bmatrix}}_{\Omega} \underbrace{\begin{bmatrix} PRI_0 \\ PRI_1 \\ \vdots \\ PRI_{N-1} \end{bmatrix}}_{\lambda} \tag{3.57}$$

As a result the state vector *X* can be written as multiplication of the vectors Ω and λ which are shown as in the Equations 3.56 and 3.57:

$$X = \Omega\lambda \tag{3.58}$$

Now, let us find the covariance matrix (C) of the state vector *X*, the covariance matrix can be written by using Equation 3.58 as:

$$C = E\left\{ \Omega(\lambda - \bar{\lambda})(\Omega(\lambda - \bar{\lambda}))^T \right\} = E\left\{ \Omega(\lambda - \bar{\lambda})(\lambda - \bar{\lambda})^T \Omega^T \right\}$$
$$= \Omega E\left\{ (\lambda - \bar{\lambda})(\lambda - \bar{\lambda})^T \right\} \Omega^T \tag{3.59}$$

In Equation 3.59, the vector λ is composed of first measured PRI values and the vector $\bar{\lambda}$ is the first PRI values without noise (measurement noise u_k, Equation 3.55) in other words actual PRI values.

So the difference of $\lambda - \overline{\lambda}$ can be written as:

$\lambda - \overline{\lambda} = \begin{bmatrix} w_0 + v_0 - v_{-1} \\ w_1 + v_1 - v_0 \\ \vdots \\ w_{N-1} + v_{N-1} - v_{N-2} \end{bmatrix}$, where *N* is the period the staggered PRI sequence.

So, the covariance matrix of λ "$E\left\{(\lambda - \overline{\lambda})(\lambda - \overline{\lambda})^T\right\}$" can be found as:

$$= \begin{bmatrix} 2\sigma_v^2 + \sigma_w^2 & -\sigma_v^2 & 0 & 0 & 0 & \dots & 0 \\ -\sigma_v^2 & 2\sigma_v^2 + \sigma_w^2 & -\sigma_v^2 & 0 & 0 & \dots & 0 \\ 0 & -\sigma_v^2 & 2\sigma_v^2 + \sigma_w^2 & -\sigma_v^2 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & -\sigma_v^2 & 2\sigma_v^2 + \sigma_w^2 & -\sigma_v^2 \\ 0 & 0 & 0 & \dots & 0 & -\sigma_v^2 & 2\sigma_v^2 + \sigma_w^2 \end{bmatrix} \quad (3.60)$$

Using the result found for $E\left\{(\lambda - \overline{\lambda})(\lambda - \overline{\lambda})^T\right\}$ (Equation 3.60), the covariance matrix (C) of the state vector *X* can be written by:

$$C = \Omega E\left\{(\lambda - \overline{\lambda})(\lambda - \overline{\lambda})^T\right\}\Omega^T \quad (3.61)$$

While calculating the covariance matrix 'C' (see Equation 3.61), the first staggered PRI sequence $(PRI_0, PRI_1, ..., PRI_{N-1})$ is used. The covariance matrix (C) can be taken as $P(N \mid N-1)$ in other words $\hat{P}_N^-$.

3.2.3.2 Staggered PRI sequence models with DFT in the case of missing pulse

To represent the missing pulses that occur when the probability of detection is less then one or when the electronic support deinterleaver [17] makes a mistake, data sets were generated with missing pulses. A missing pulse is represented by a missing PRI value followed by a large PRI value. The large PRI value is the sum of the missing PRI and the PRI that follows it. If there are missing pulses in the received data, the following procedure is used:

The Kalman filter makes a prediction ($H_k X_k$) with an associated variance at each iteration. If the next received PRI is not in the four standard deviations (because, taking +/-3 standard deviation includes %99, 74 of the Gaussian distribution) of the prediction then we can say that there is a missing PRI. This false value is not used to update the filter. Also, the next prediction is adjusted to compensate for the false PRI. So, the next prediction is the sum of two PRIs.

As a result, a gate is opened for each measured (received) PRI as follows:

$$H_k X_k - 4\sqrt{\sigma_w^2 + 2*\sigma_v^2} \leq Next\,received\,PRI \leq H_k X_k + 4\sqrt{\sigma_w^2 + 2*\sigma_v^2} \qquad (3.62)$$

In equation 3.62 "k" is the index of the k^{th} pulse.

Let us write the system equations (Equation 3.54) and observation equations (Equation 3.55) for each pulse in the case of <u>no missing pulse</u> as follows:

System Equation:	Observation Equation:
$X_k = X_{k-1}$	$Y_k = H_k X_k + u_k$
$X_{k+1} = X_k$	$Y_{k+1} = H_{k+1} X_{k+1} + u_{k+1}$
$X_{k+2} = X_{k+1}$	$Y_{k+2} = H_{k+2} X_{k+2} + u_{k+2}$
$\vdots$	$\vdots$

Assume that the pulse with index "k+1" is missing. Since the pulse with index "k+1" is missing, the indices of the next pulses (i.e. "k+2", "k+3","k+4",..) must be renamed. Also, the indices of the measurement vector H must be increased by 1 for each missing pulse after missing pulse points to obtain the correct PRI values. So, let us rewrite the system and observation equations, if the pulse with index "k+1" is missing, as follows:

System Equation:	Observation Equation:
$X_k = X_{k-1}$	$Y_k = H_k X_k + u_k$
~~$X_{k+1} = X_k$~~	~~$Y_{k+1} = H_{k+1} X_{k+1} + u_{k+1}$~~
$X_{k+1} = X_{k-1}$	$Y_{k+1} = H_{k+2} X_{k+1} + u_{k+1}$

$X_{k+2} = X_{k+1}$ $\qquad\qquad$ $Y_{k+2} = H_{k+3} X_{k+2} + u_{k+2}$

$\vdots$ $\qquad\qquad$ $\vdots$

Also, at the missing pulse points, state prediction error covariance is written by:

$$\hat{P}_{k+1} = P_k \tag{3.63}$$

CHAPTER 4

SIMULATIONS

Matlab is chosen as the simulation tool, because of its flexibility in mathematical environment. The algorithms developed in section 3.2 are implemented and Monte Carlo simulation method is used for testing. The results collected from filter output are displayed with graphics. Gaussian Normal error distributions with different variances are used in the simulations to see the effect of the error on the received data (measured time of arrival values). Also, the effect of the missing pulses is analyzed in the simulation and robustness of the algorithms to missing pulses is discussed. In order to check the performance of the algorithms, the received data is needed. In other words, measured TOA values must be generated. The "*randn(.)*" command of Matlab is used to generate the Normal gaussian distributed errors according the corresponding variances. The TOA data is simulated according to the scenarios discussed in section 3.2.

The simulation results given below show the performance of the algorithms described in section 3.2. Results obtained from different parameter values are plotted on the same graph to increase the comprehension. Measurement error is defined as the error between true PRI values and measured PRI values. Estimation error is defined as the error between true PRI values and estimated PRI values.

4.1 Simulation Results for Algorithm I

Four different simulations were implemented for Algorithm I. In the simulations, a staggered PRI sequence with a period of 7 was used. Simulations 1 and 2 show the performance of the Algorithm I in the case of no missing pulse, simulations 3 and 4 show the performance of the Algorithm I in the case of missing pulses. The implemented simulations are as follows:

Simulation 1:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. So this sequence is called as a 6-level, 7-position staggered PRI sequence. It is assumed that there is no missing pulse in the received data. Number of pulses in the received data is 280. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 0.6.

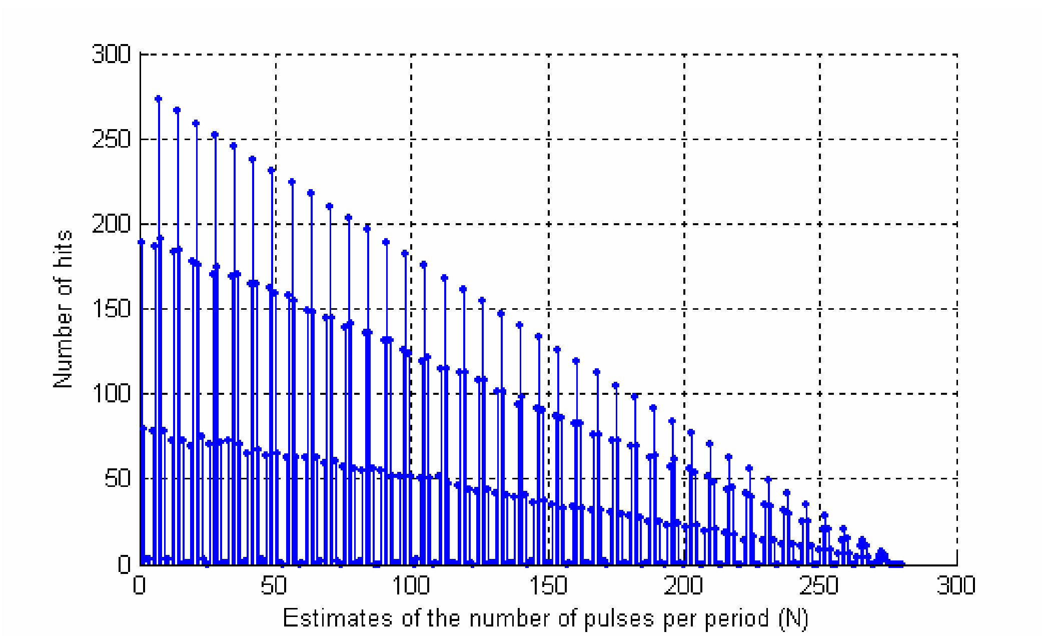

Figure 12. Finding Period of the Sequence

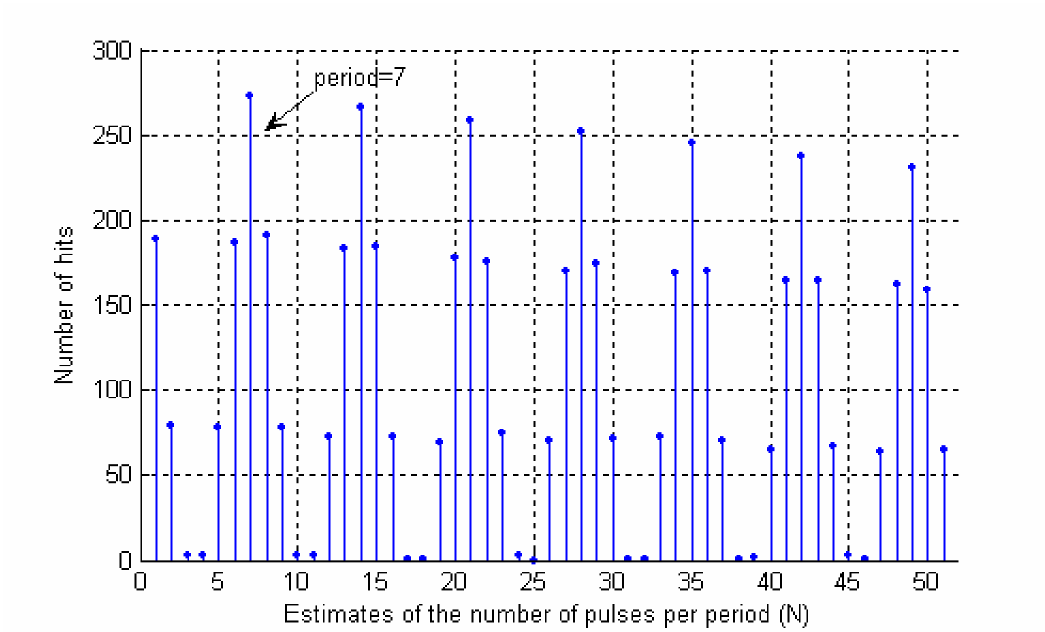

Figure 13. Enlargement of "Figure 12"

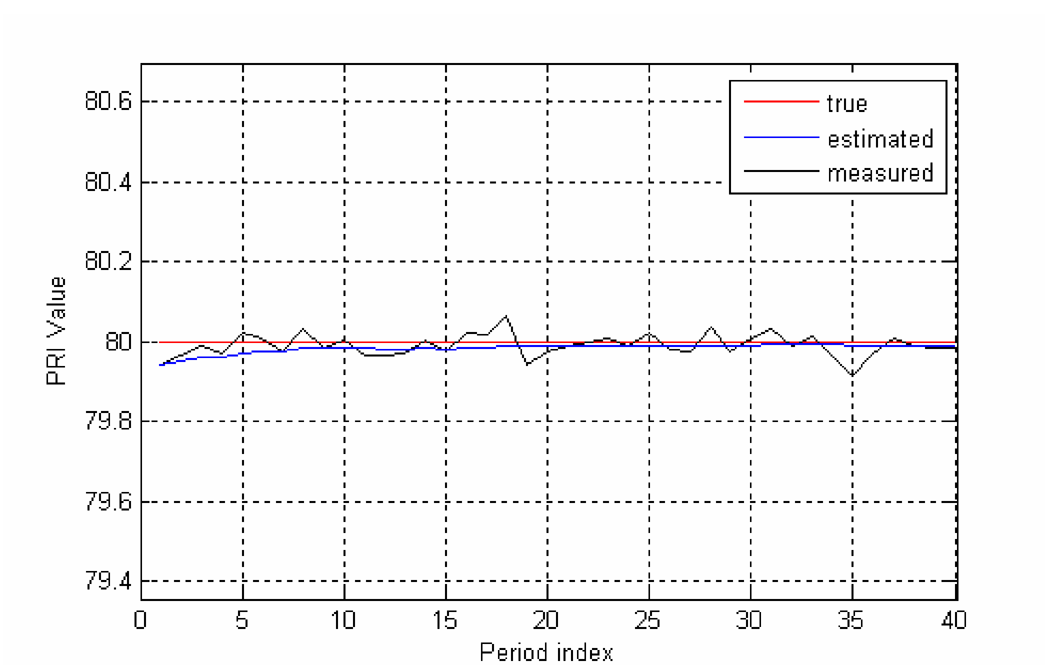

Figure 14. Measured and Estimated PRI Values for the PRI value of 80

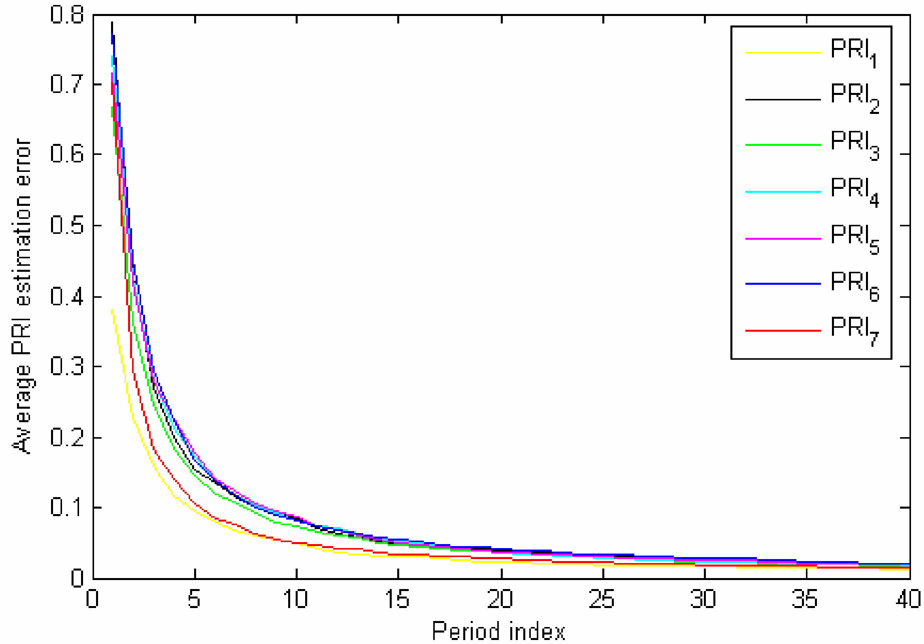

Figure 15. Average PRI Estimation Error

Simulation 2:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. So this sequence is called as a 6-level, 7-position staggered PRI sequence. It is assumed that there is no missing pulse in the received data. Number of pulses in the received data is 280. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 1.

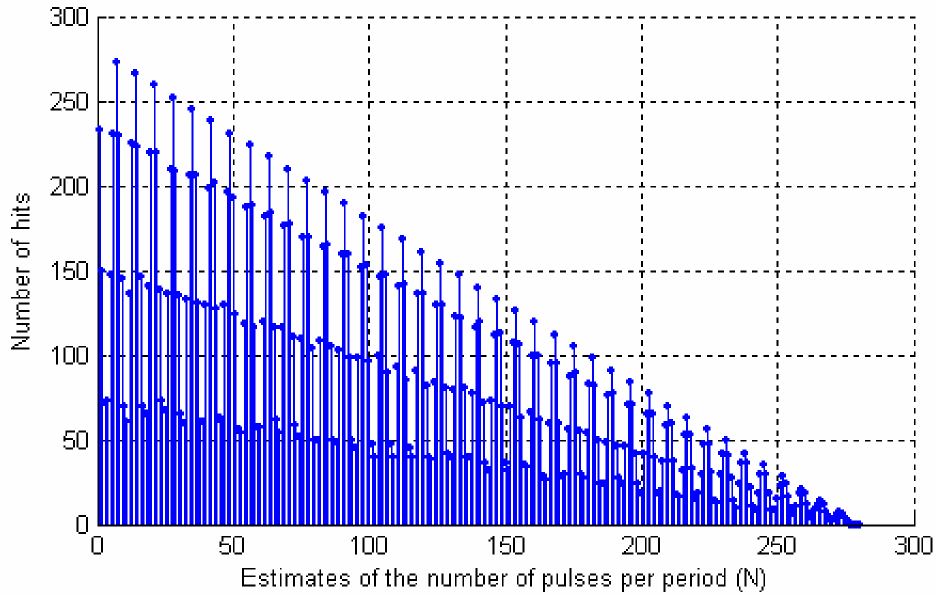

Figure 16. Finding Period of the Sequence

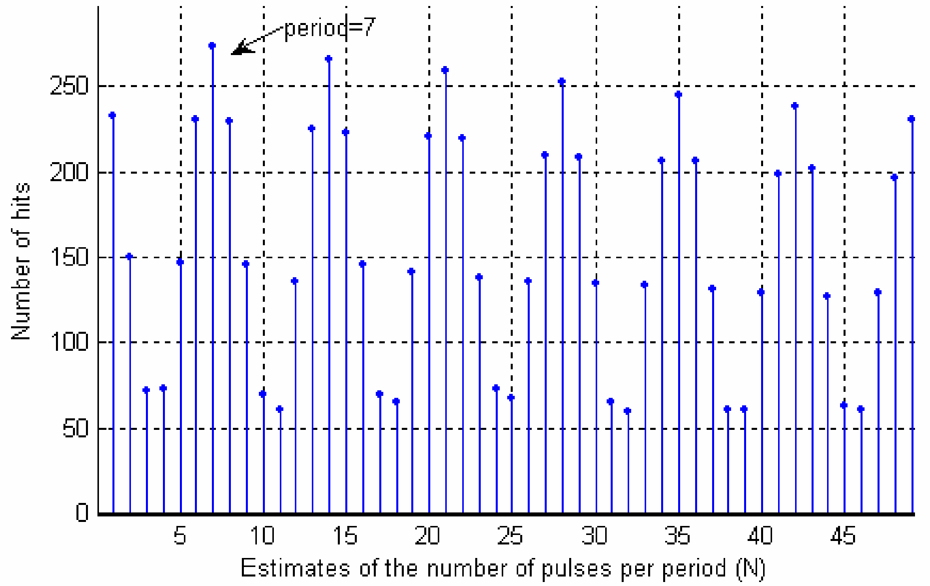

Figure 17. Enlargement of "Figure 16"

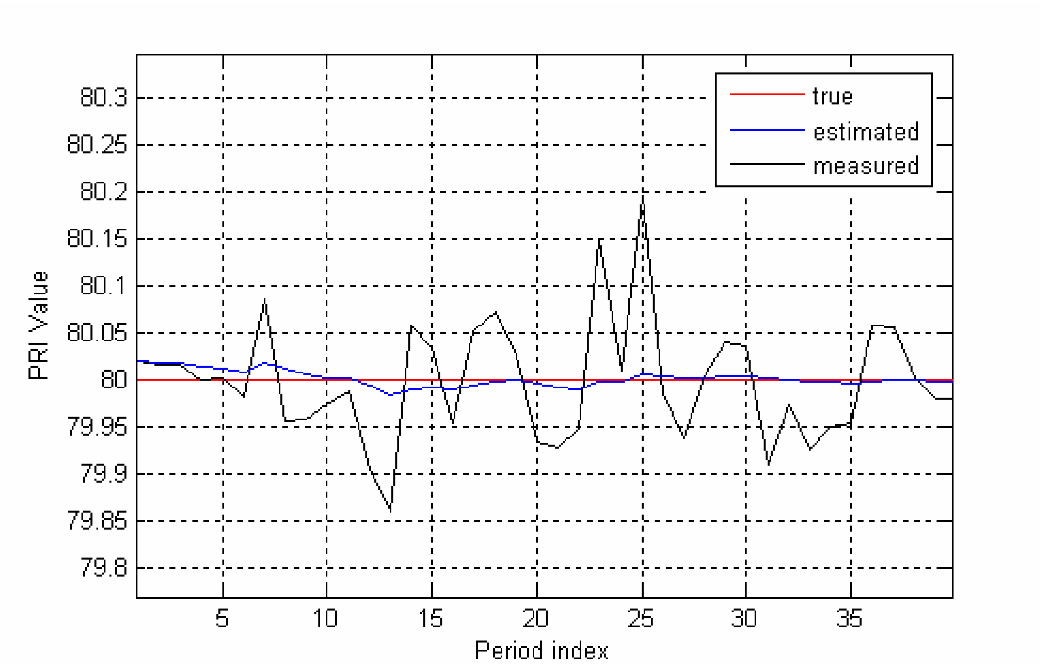

Figure 18. Measured and Estimated PRI Values for the PRI value of 80

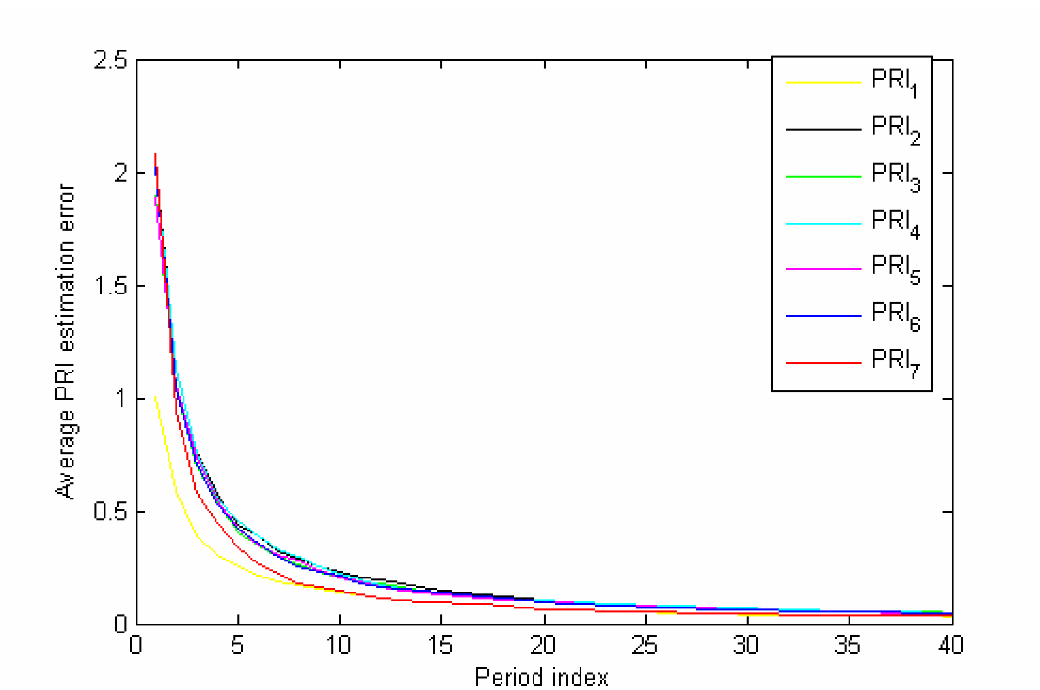

Figure 19. Average PRI Estimation Error

Simulation 3:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. So this sequence is called as a 6-level, 7-position staggered PRI sequence. It is assumed that there are 8 missing pulses in the received data with indices 11, 61, 97, 169, 193, 229, 243 and 251. The number of pulses in the received data is 272. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 1.

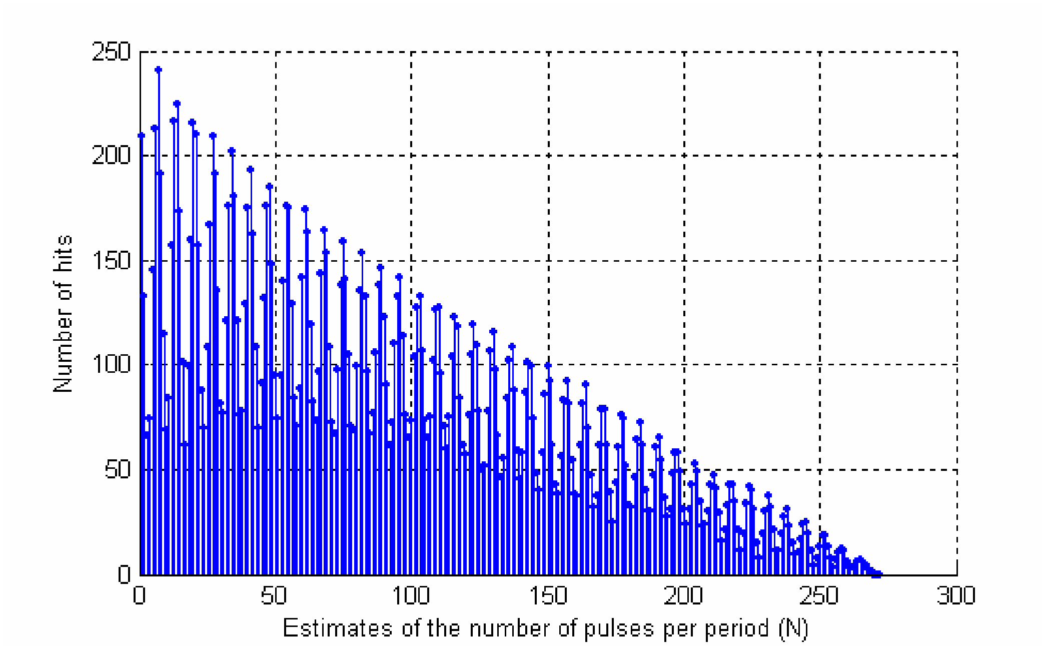

Figure 20. Finding Period of the Sequence

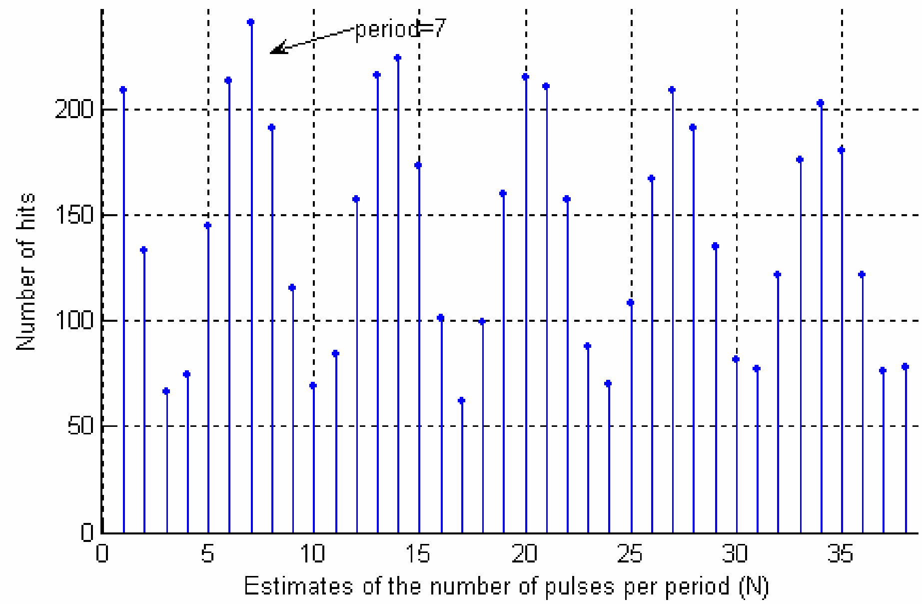

Figure 21. Enlargement of "Figure 20"

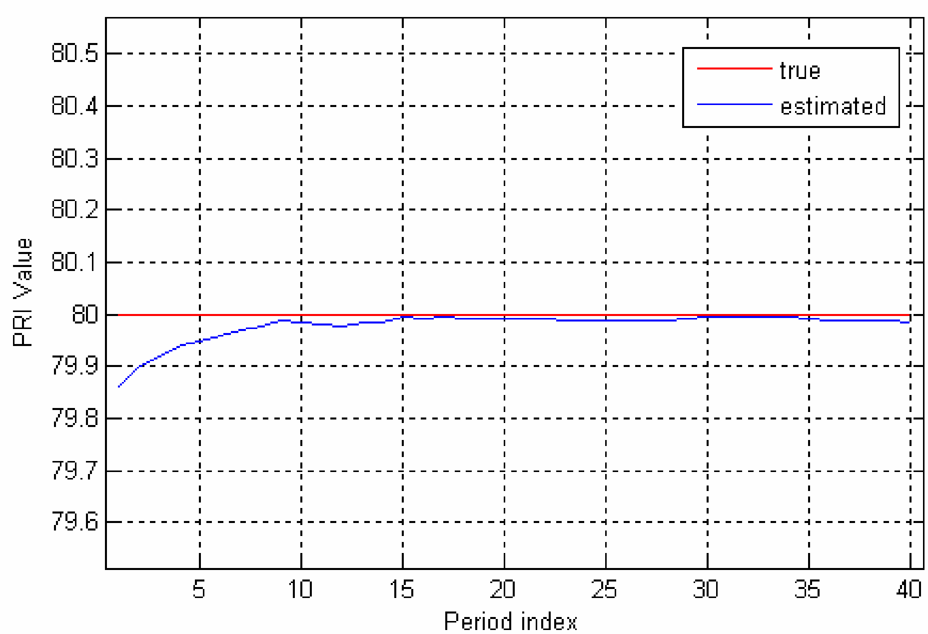

Figure 22. Estimated PRI Values for the PRI value of 80

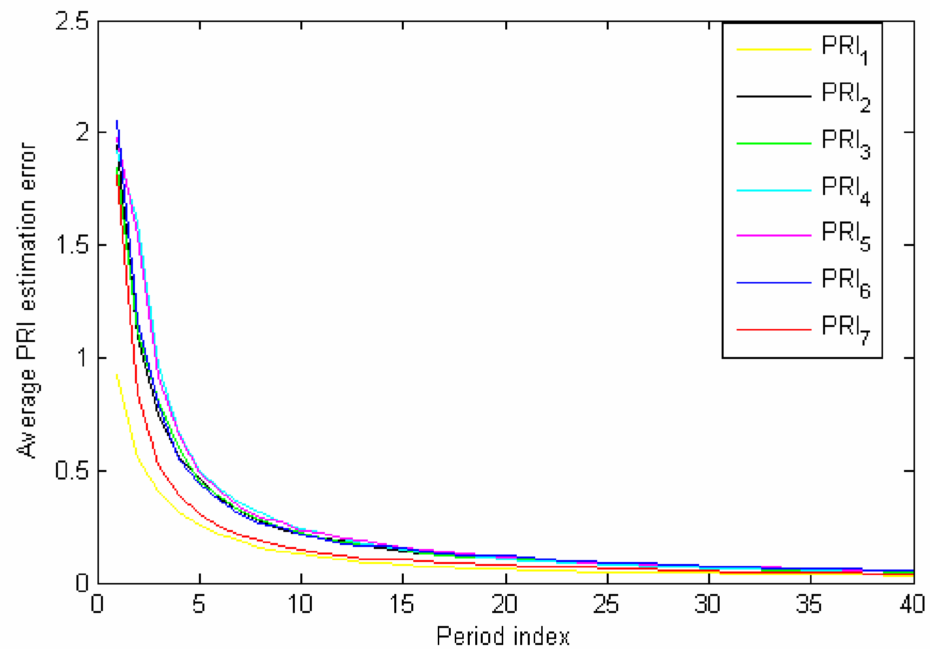

Figure 23. Average PRI Estimation Error

Simulation 4:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. So this sequence is called as a 6-level, 7-position staggered PRI sequence. It is assumed that there are 8 missing pulses in the received data with indices 11, 61, 97, 169, 193, 229, 243 and 251. The number of pulses in the received data is 272. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 2.

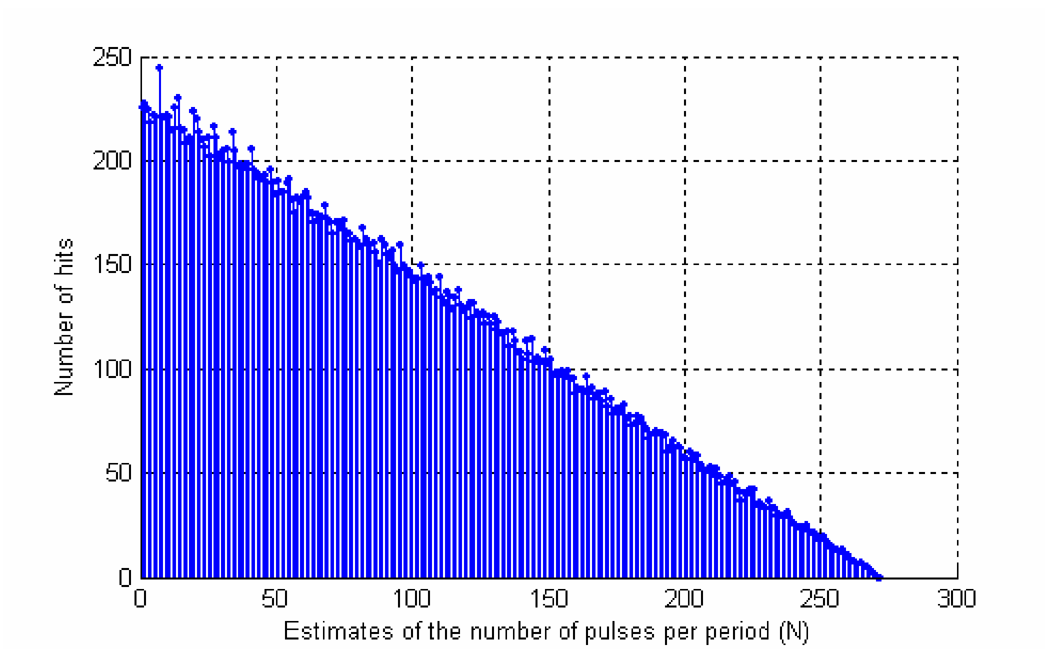

Figure 24. Finding Period of the Sequence

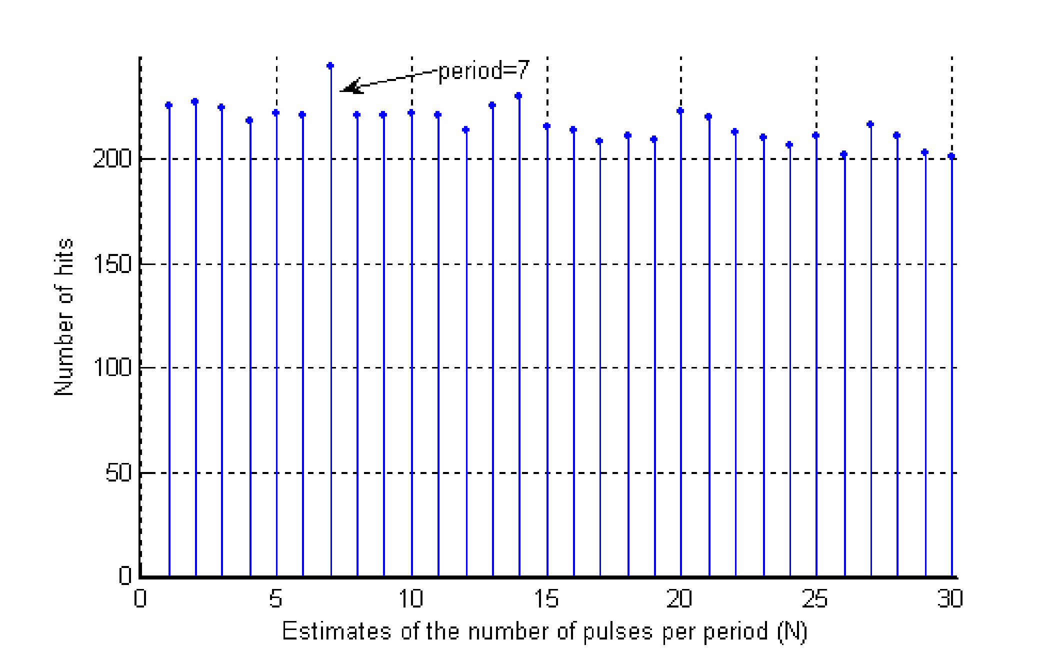

Figure 25. Enlargement of "Figure 24"

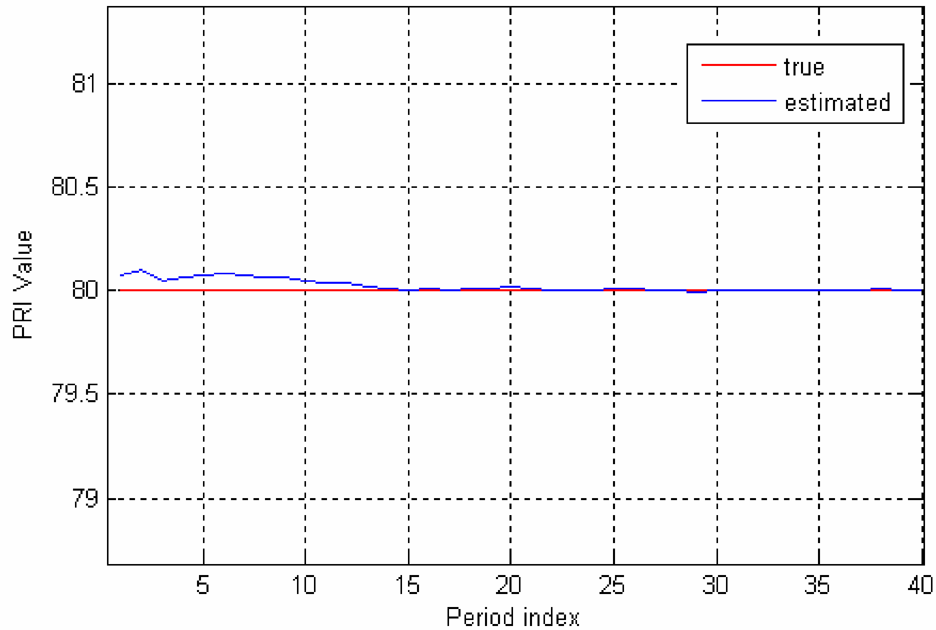

Figure 26. Estimated PRI Values for the PRI value of 80

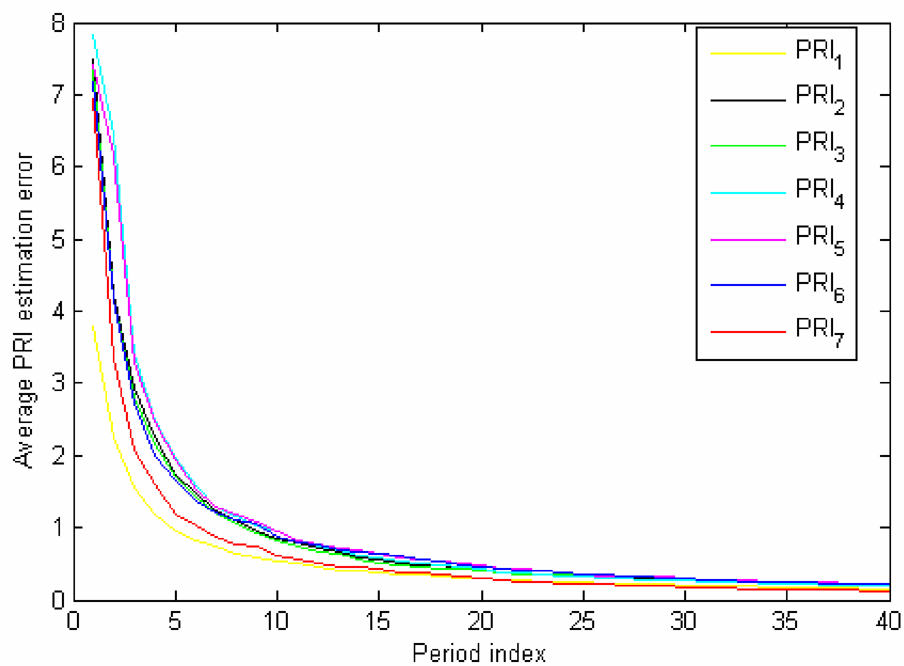

Figure 27. Average PRI Estimation Error

Comment on simulations:

It can be observed from the results of the Simulations 1, 2, 3 and 4 (Figures 12, 13, 16, 17, 20, 21, 24, and 25) that the detection of period of the staggered PRI sequence performed well even if there are missing pulses in the received data and repetitions (existence of the same PRI values) in the staggered PRI sequence. The convergence property of the algorithm is shown by plotting the true, estimated and measured PRI values for the PRI value of 80 on the same graph (Figures 14, 18). It is obvious from the Figures 14, 18, 22 and 26 that the estimated PRI values obtained from noisy measured PRI values converges to the true PRI values after some number of PRI estimations. According to the Figure 15, 19, 23 and 27, the average estimation error for each PRI value in one period of the staggered PRI sequence goes to zero as period index increases. The PRI values in one period of staggered PRI sequence are denoted by PRI_1, PRI_2, PRI_3, PRI_4, PRI_5, PRI_6 and PRI_7 as shown in Figure 15, 19, 23 and 27. From Figure 23 and Figure 27, it can be observed that algorithm performs well even in the case of missing pulses, because the algorithm is robust to the effect of the missing pulses.

4.2 Simulation Results for Algorithm II

Four different simulations were implemented for Algorithm II. In the simulations, a staggered PRI sequence with a period of 7 was used. Simulations 1 and 2 show the performance of the Algorithm II in the case of no missing pulse, simulations 3 and 4 show the performance of the Algorithm II in the case of missing pulses. The implemented simulations are as follows:

Simulations 1:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. It is assumed that there is no missing pulse in the received data. Number of pulses in the received data is 280. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 0.6.

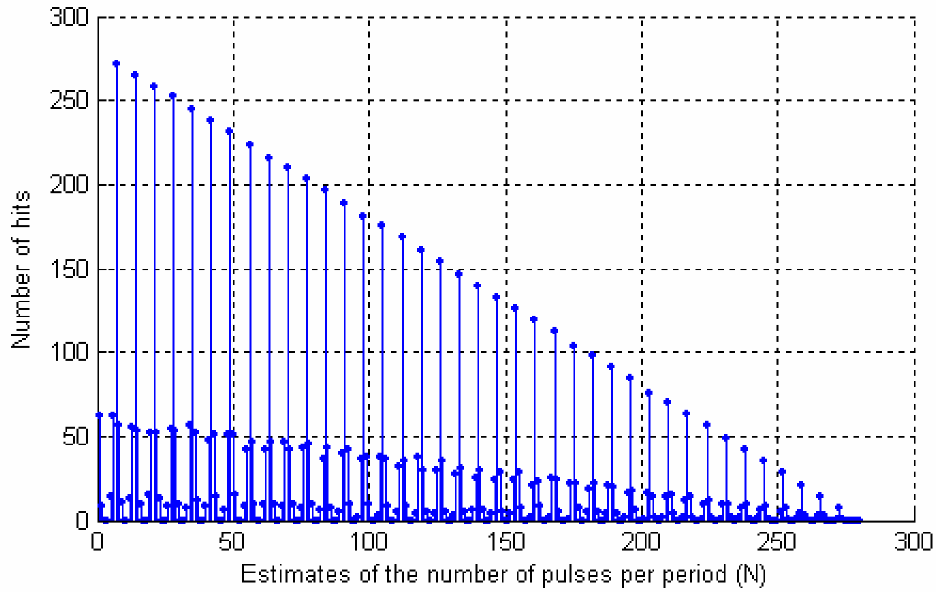

Figure 28. Finding Period of the Sequence

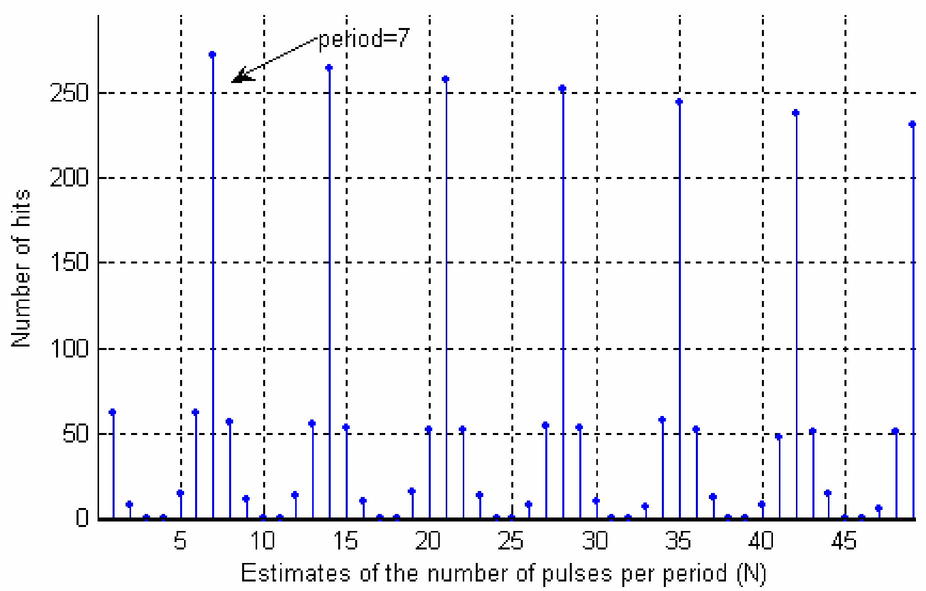

Figure 29. Enlargement of "Figure 28"

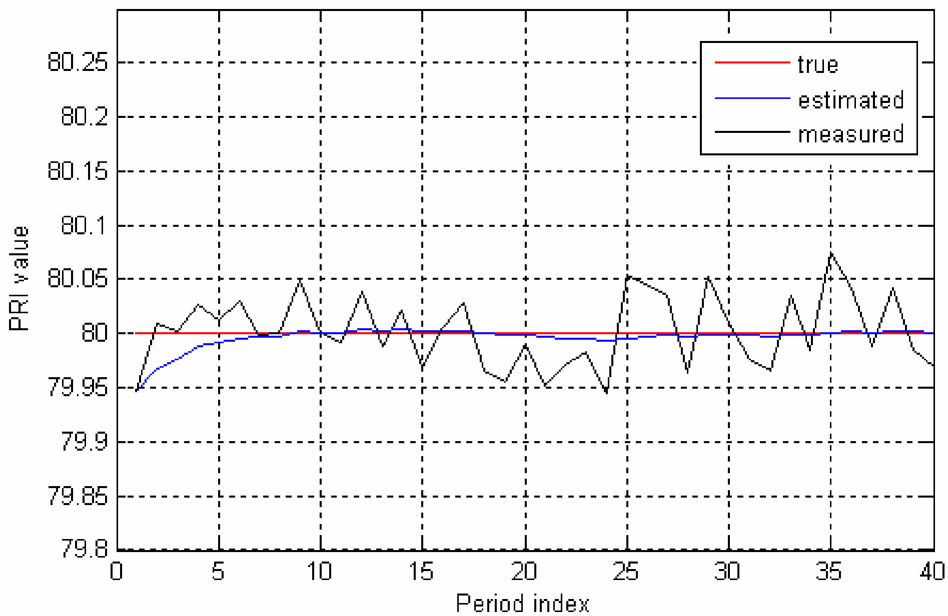

Figure 30. Measured and Estimated PRI Values for the PRI value of 80

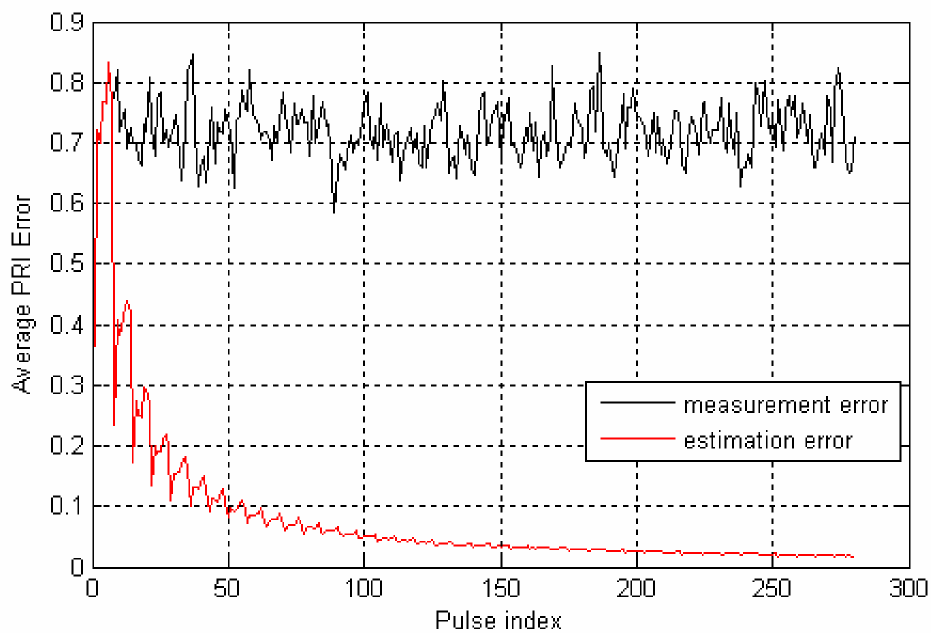

Figure 31. Average PRI Error

Simulations 2:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. It is assumed that there is no missing pulse in the received data. Number of pulses in the received data is 280. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 1.

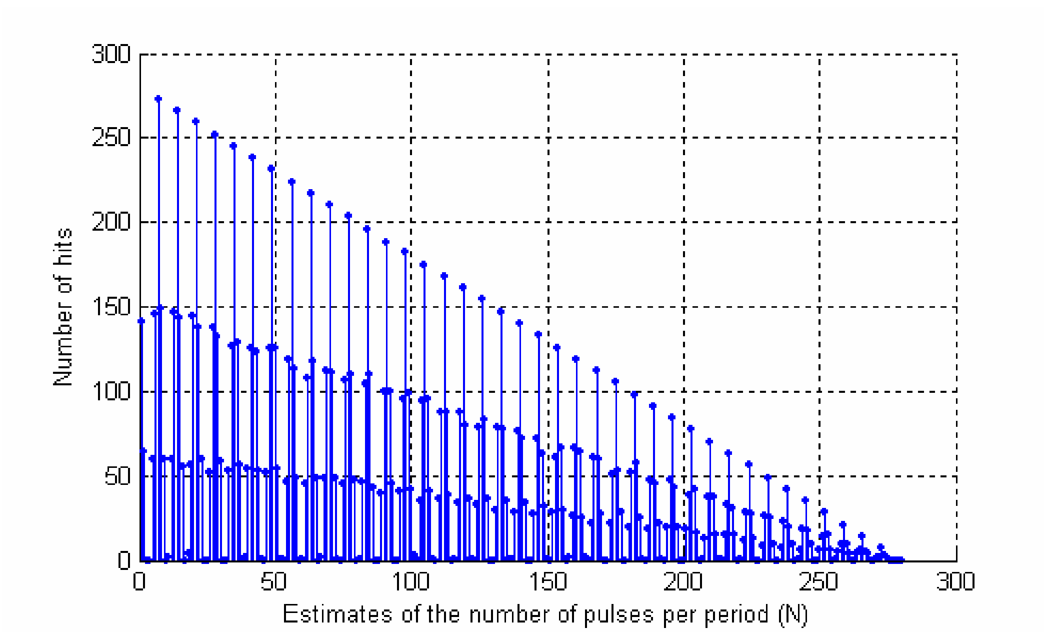

Figure 32. Finding Period of the Sequence

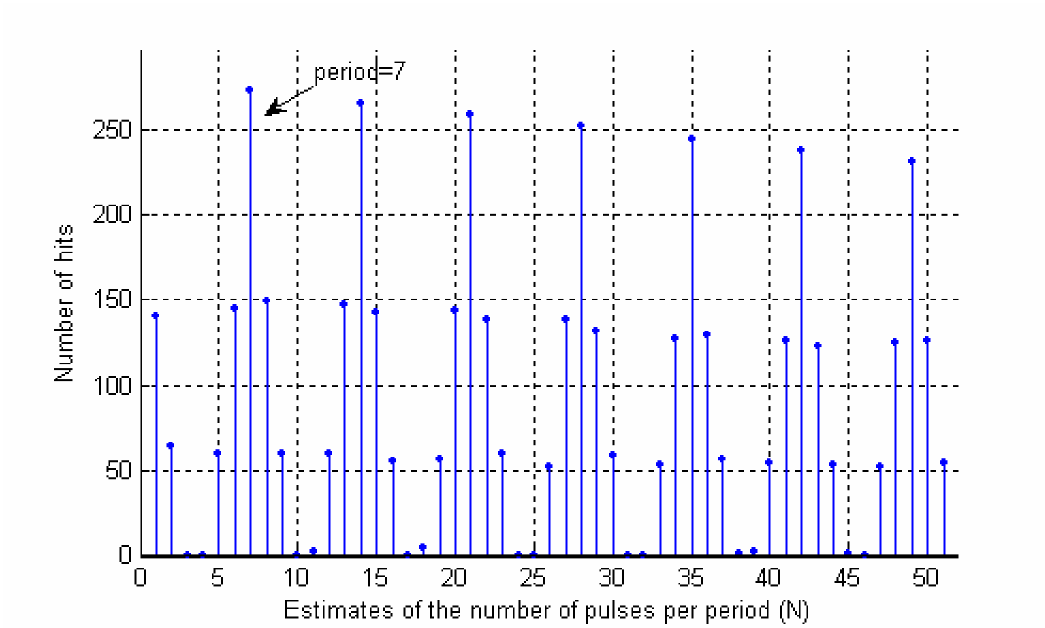

Figure 33. Enlargement of "Figure 32"

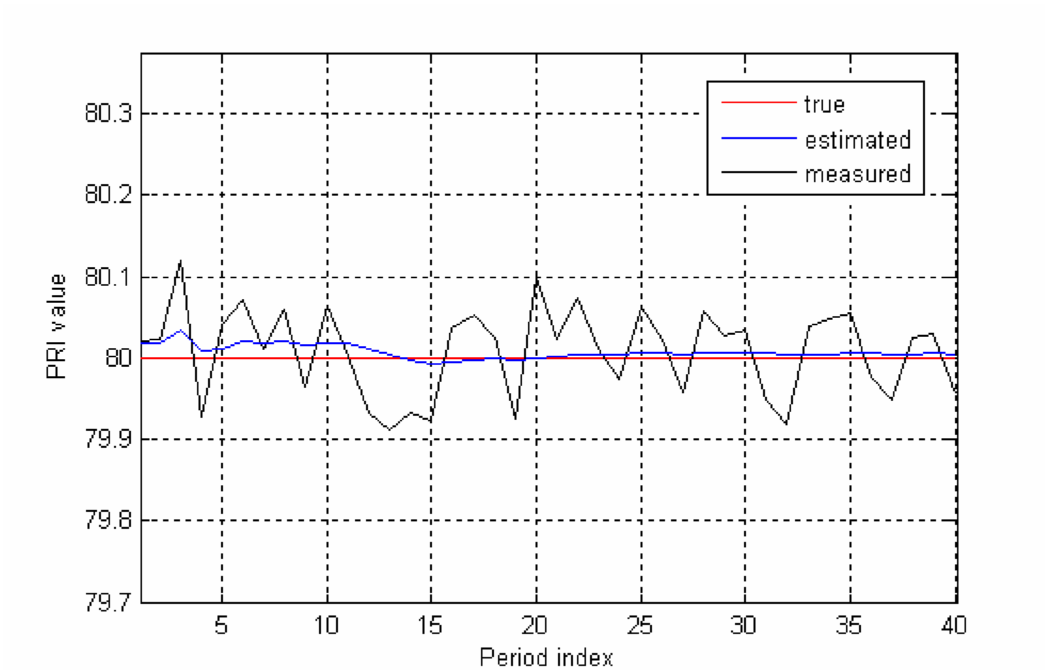

Figure 34. Measured and Estimated PRI Values for the PRI value of 80

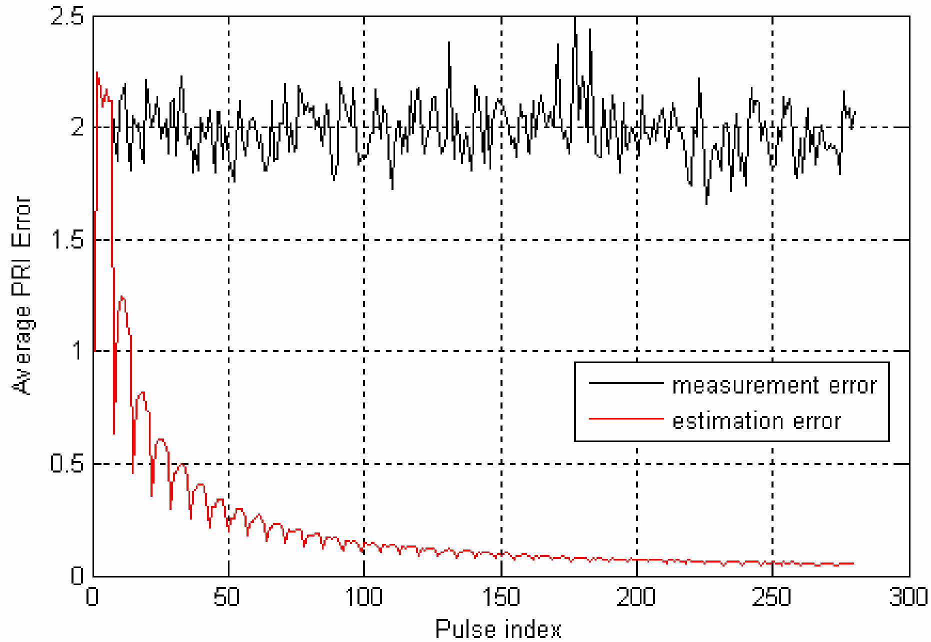

Figure 35. Average PRI Error

Simulation 3:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. It is assumed that there are 8 missing pulses in the received data with indices 11, 61, 97, 169, 193, 229, 243 and 251. So the number of pulses in the received data is 272. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 1.

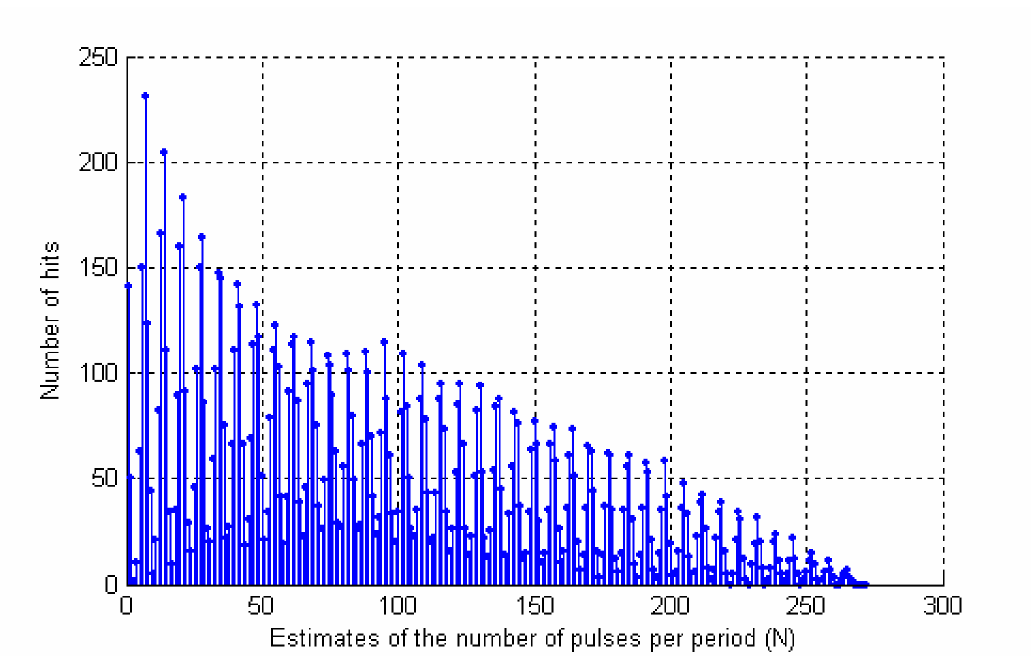

Figure 36. Finding Period of the Sequence

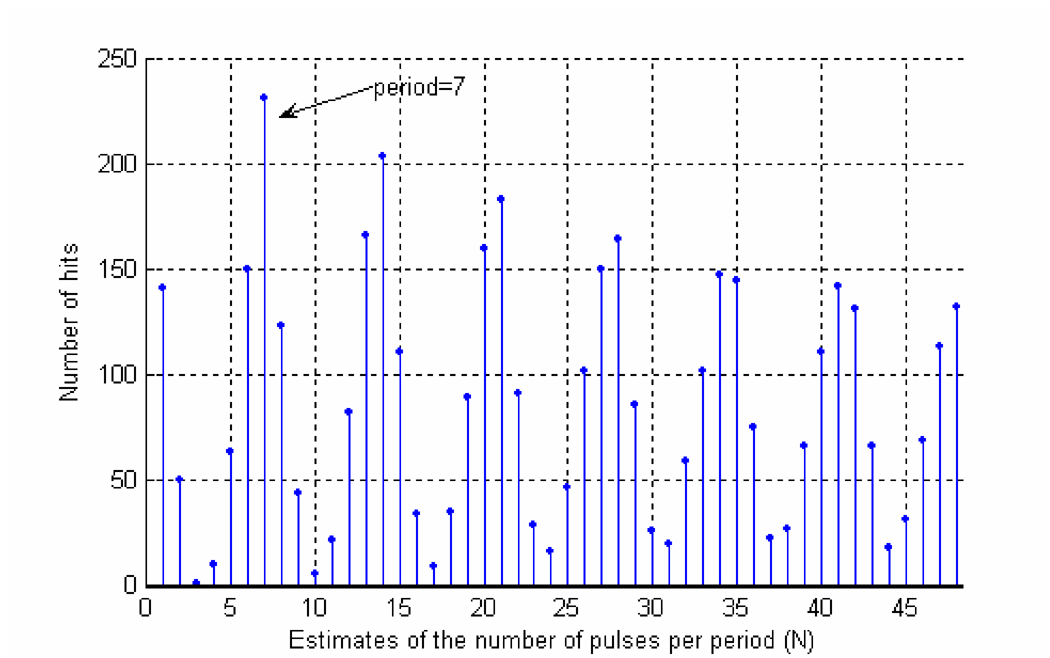

Figure 37. Enlargement of "Figure 36"

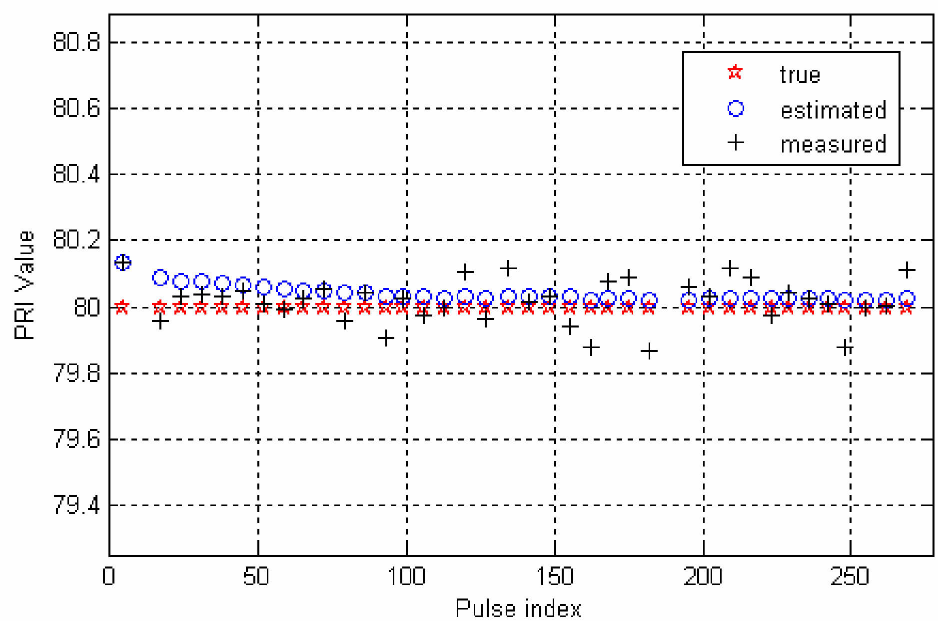

Figure 38. Measured and Estimated PRI Values for the PRI value of 80

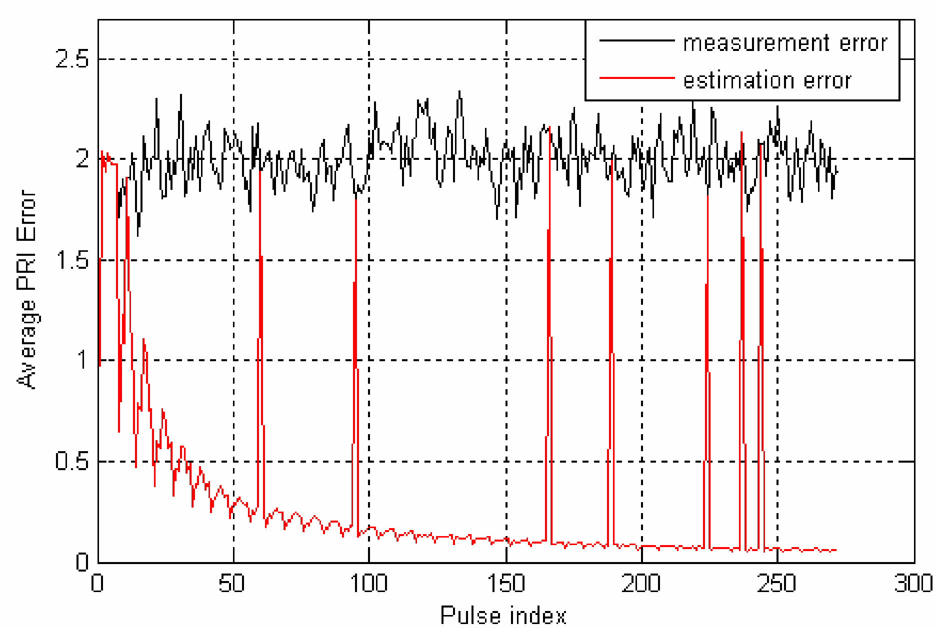

Figure 39. Average PRI Error

Simulation 4:

Staggered PRI sequence with a period of 7 was used in the simulation. One period of the Staggered PRI sequence is [60 70 75 80 85 85 90]. It is assumed that there are 8 missing pulses in the received data with indices 11, 61, 97, 169, 193, 229, 243 and 251. So the number of pulses in the received data is 272. Monte Carlo simulations with 500 runs were implemented. Standard deviation of noise w (with zero mean) is 0.01 and standard deviation of the noise v (zero mean) is 2.

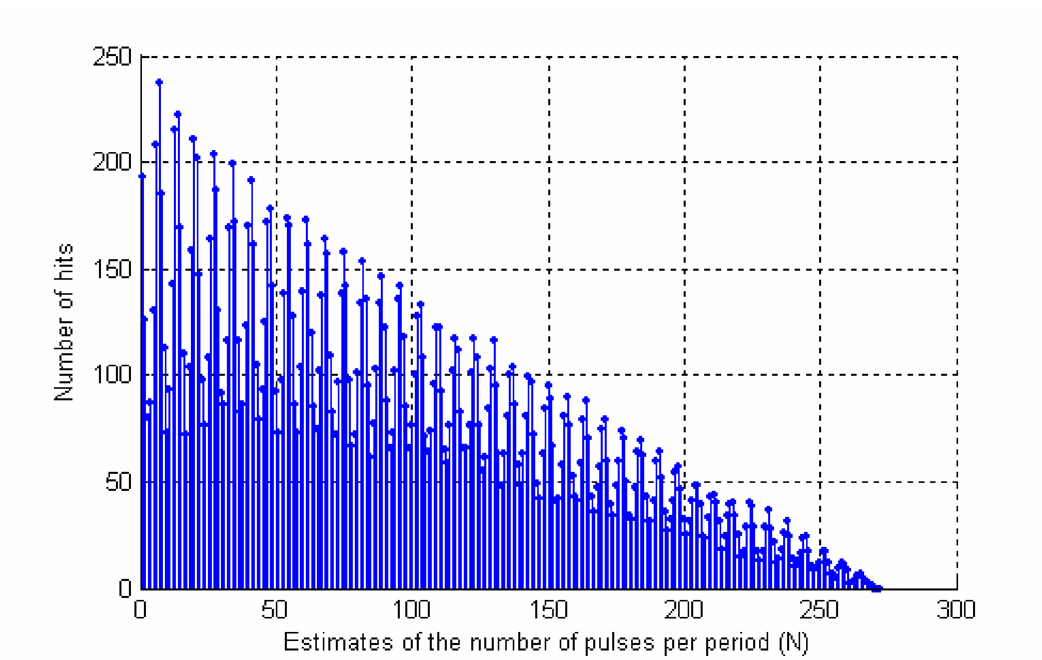

Figure 40. Finding Period of the Sequence

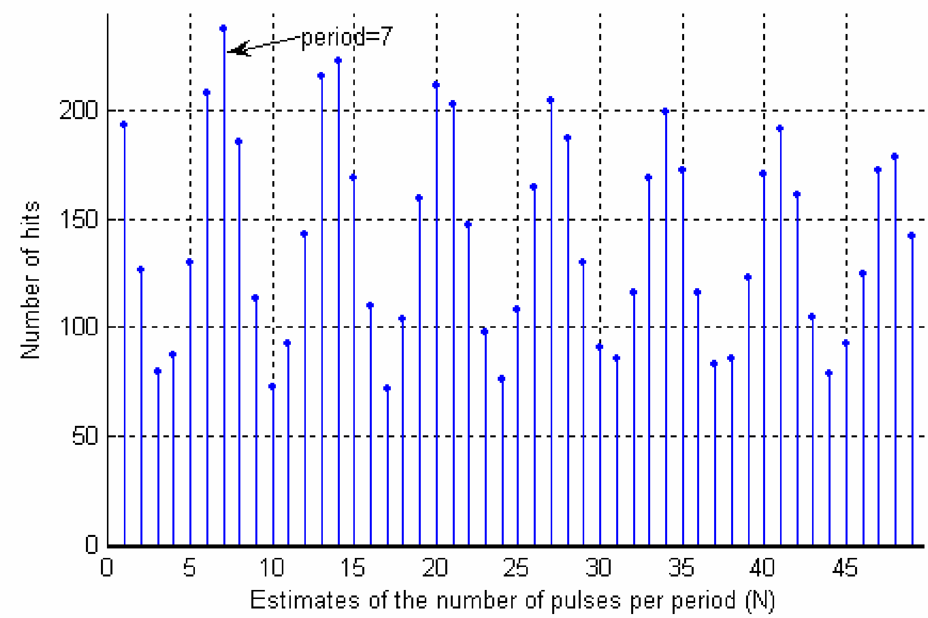

Figure 41. Enlargement of "Figure 40"

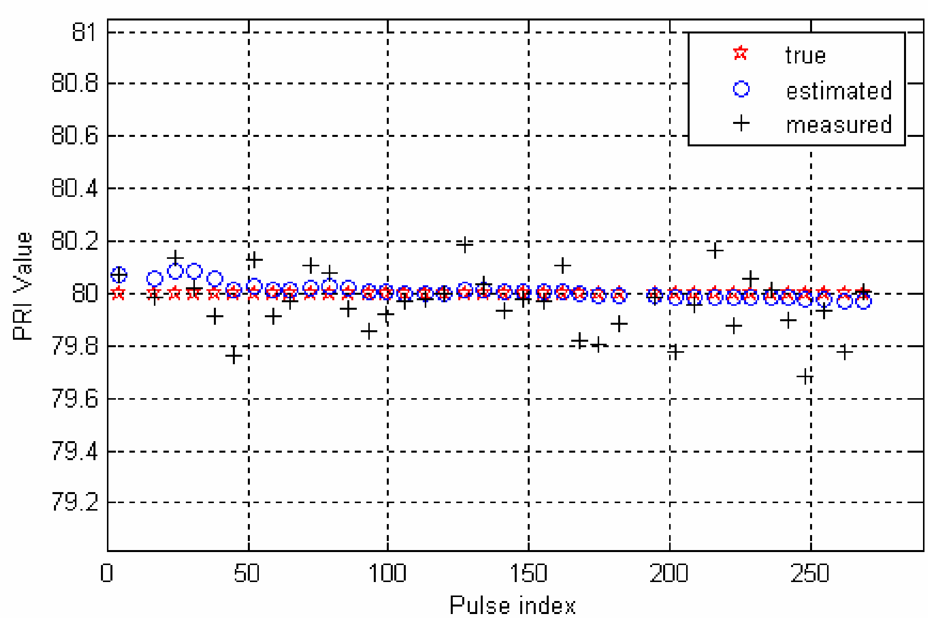

Figure 42. Measured and Estimated PRI Values for the PRI value of 80

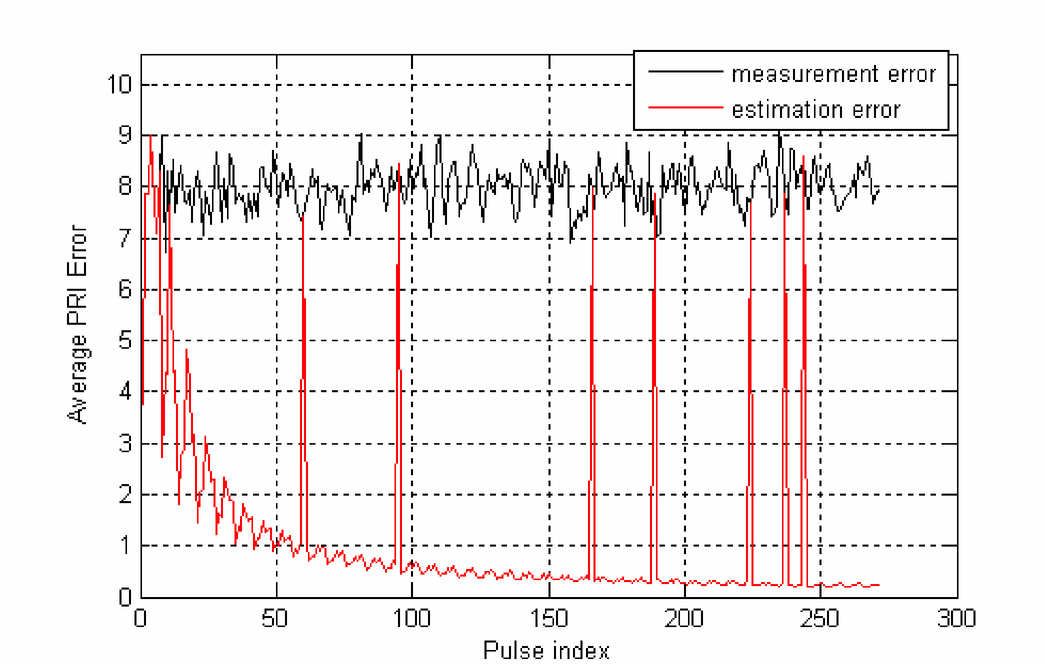

Figure 43. Average PRI Error

<u>Comment on simulations</u>:

It can be observed from the results of the simulations 1, 2, 3 and 4 (Figures 28, 29, 32, 33, 36, 37, 40 and 41) that the detection of period of the staggered PRI sequence performed well even if there are missing pulses in the received data and repetitions (existence of the same PRI values) in the staggered PRI sequence. The convergence property of the algorithm is shown by plotting the true, measured and estimated PRI values on the same graph (Figures 30, 34, 38 and 42). It is obvious from the Figures 30, 34, 38 and 42 that the estimated PRI values obtained from noisy measured PRI values converges to the true PRI values after some number of PRI estimations. The average measurement and estimation error distributions are also plotted on the same graph to increase the comprehension. According to the Figures 31, 35, 39 and 43, the average estimation error is much smaller than the average measurement error. From Figures 39 and 43 which show the average and measurement estimation errors in the case of missing pulses, it can be observed that the effect of the missing pulses is minimized, because the algorithm is also robust to the effect of the missing pulses. When Figures 39 and 43 are analyzed, we see some fluctuations on the graph. The fluctuations occur only at the missing pulse point which means that "the next received PRI value is greater than the current prediction plus four standard deviations, so it is assumed to be a missing PRI. It is not used to update the filter and the next prediction is adjusted to compensate for the missing PRI. At these points, prediction is the sum of two PRIs. As a result, the estimation error increases at the point where there is a missing pulse but this does not affect the next PRI estimations. These fluctuations at the missing pulse points are due to the methodology used in the algorithm to make the algorithm robust to the effect of the missing pulses in the received data.

4.3 Comparison of Algorithm I and Algorithm II

Five different simulations were implemented to compare the performances of Algorithm I and Algorithm II. In the simulations, staggered PRI sequence with periods 7 and 10 were used. Simulations 1 and 3 show the performances of both algorithms in the case of no missing pulse. Simulations 2 and 4 show the performances of both algorithms in the case of missing pulses. Simulation 5 shows the implementation runtimes of both algorithms with respect to the total number of pulses and the period of the staggered PRI sequence. The implemented simulations are as follows:

<u>Simulation 1:</u>

In this simulation, PRI estimation performances of both algorithms in the case of no missing pulses were compared. So, the average PRI estimation errors for both algorithms were shown on the same graph (Figure 44) to increase comprehension. The same staggered PRI sequence with same noise levels is used. One period of the staggered PRI sequence used in the simulation is [60 70 75 80 85 85 90]. This sequence is called as a 6-level, 7-position staggered PRI sequence. It is assumed that there is no missing pulse in the received data. The number of pulses in the received data is 280. Monte Carlo simulations with 500 runs were implemented, standard deviation of the system noise (sigma_w, with zero mean) is 0.01 and standard deviation of the measurement noise (sigma_v, with zero mean) is 0.6. As it can be seen easily from Figure 44, PRI estimation performance of Algorithm I is slightly better than the PRI estimation performance of Algorithm II.

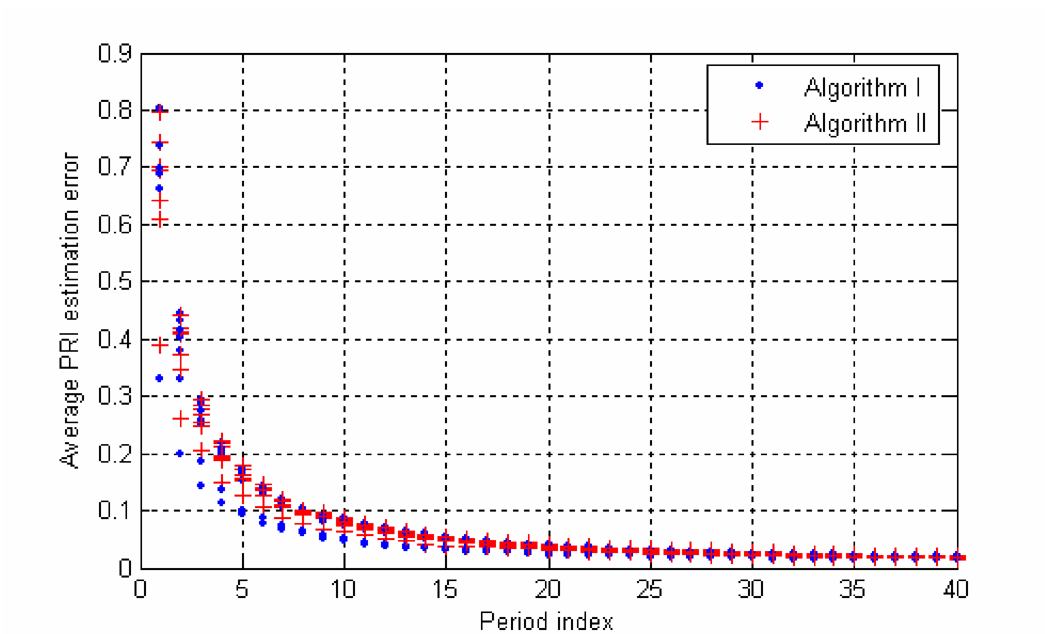

Figure 44. Average PRI Estimation Error for Algorithm I and Algorithm II

<u>Simulation 2:</u>

In this simulation, PRI estimation performances of both algorithms in the presence of missing pulses were compared. So, the average PRI estimation errors for both algorithms were shown on the same graph (Figure 45) to increase comprehension. The same staggered PRI sequences with the same noise levels are used. One period of the staggered PRI sequence used in the simulation is [60 70 75 80 85 85 90]. This sequence is called as a 6-level, 7-position staggered PRI sequence. It is assumed that there are 8 missing pulses in the received data with indices 11, 61, 97, 169, 193, 229, 243 and 251. The number of periods in the received data is 40. Monte Carlo simulations with 500 runs were implemented, standard deviation of the system noise (sigma_w, with zero mean) is 0.01 and standard deviation of the measurement noise (sigma_v, with zero mean) is 0.6.

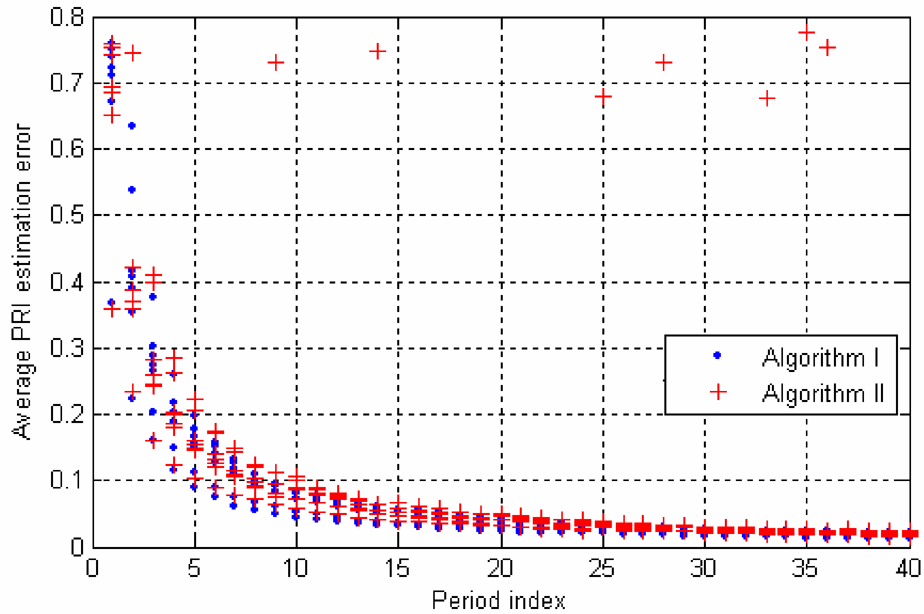

Figure 45. Average PRI Estimation Error for Algorithm I and Algorithm II

Simulation 3:

As in Simulation 1, PRI estimation performances of both algorithms in the case of no missing pulses were compared. So, the average PRI estimation errors for both algorithms were shown on the same graph (Figure 46) to increase comprehension. The same staggered PRI sequence with same noise levels is used. One period of the staggered PRI sequence used in the simulation is [60 70 75 80 85 85 90 90 95 100]. This sequence is called as an 8-level, 10-position staggered PRI sequence. It is assumed that there is no missing pulse in the received data. The number of pulses in the received data is 280. Monte Carlo simulations with 500 runs were implemented, standard deviation of the system noise (sigma_w, with zero mean) is 0.01 and standard deviation of the measurement noise (sigma_v, with zero mean) is 2. As it can be seen easily from Figure 46, PRI estimation performance of Algorithm I is slightly better than the PRI estimation performance of Algorithm II.

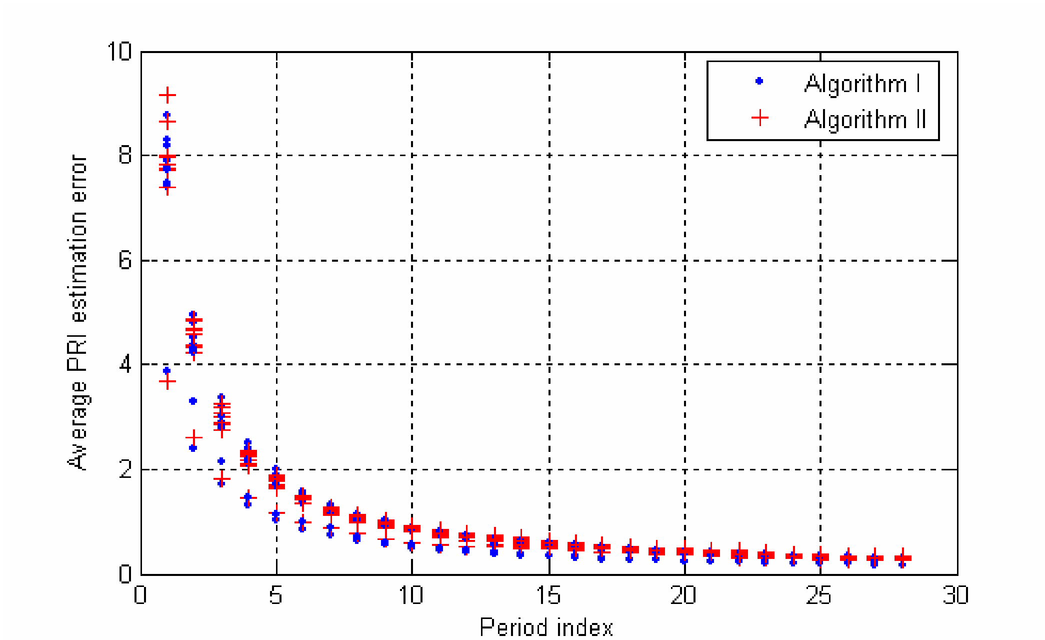

Figure 46. Average PRI Estimation Error for Algorithm I and Algorithm II

<u>Simulation 4:</u>

In this simulation, PRI estimation performances of both algorithms in the presence of missing pulses were compared for the staggered sequence used in Simulation 3. So, the average PRI estimation errors for both algorithms were shown on the same graph (Figure 47) to increase comprehension. The same staggered PRI sequences with the same noise levels are used. One period of the staggered PRI sequence used in the simulation is [60 70 75 80 85 85 90 90 95 100]. This sequence is called as a 8-level, 8-position staggered PRI sequence. It is assumed that there are 8 missing pulses in the received data with indices 11, 61, 97, 169, 193, 229, 243 and 251. The number of periods in the received data is 28. Monte Carlo simulations with 500 runs were implemented, standard deviation of the system noise (sigma_w, with zero mean) is 0.01 and standard deviation of the measurement noise (sigma_v, with zero mean) is 2.

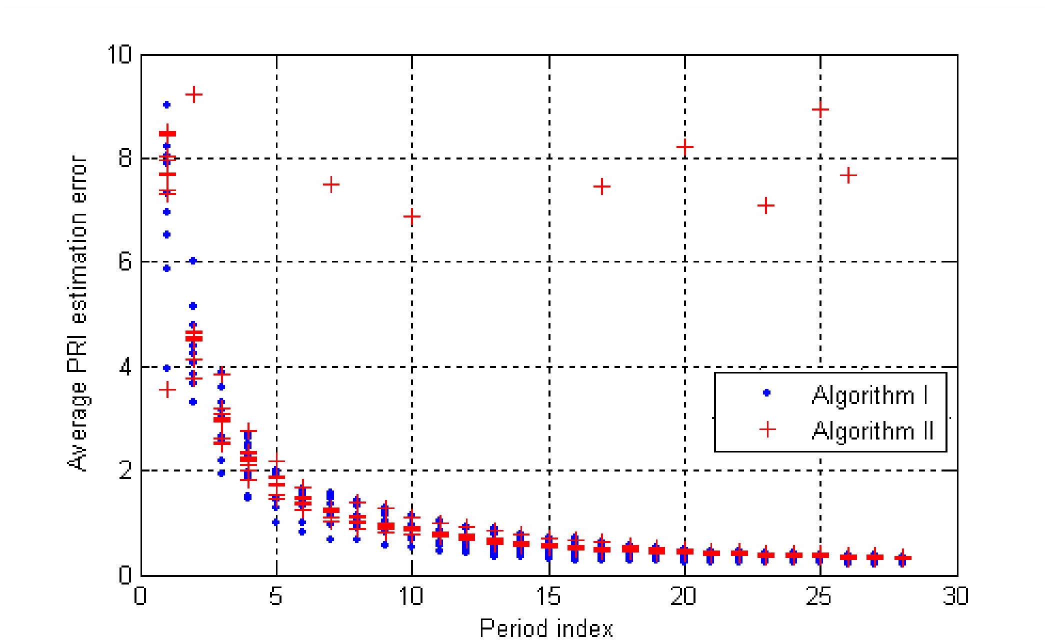

Figure 47. Average PRI Estimation Error for Algorithm I and Algorithm II

Simulation 5:

In this simulation, implementation times of both algorithms were compared. Figure 48 shows the algorithm runtimes in second with respect to total number of pulses in the received data. The same staggered PRI sequence is used with a period of 7 in the simulation.

Figure 49 shows the algorithm runtimes in second with respect to the period of the staggered PRI sequence. The same staggered PRI sequence is used with different periods.

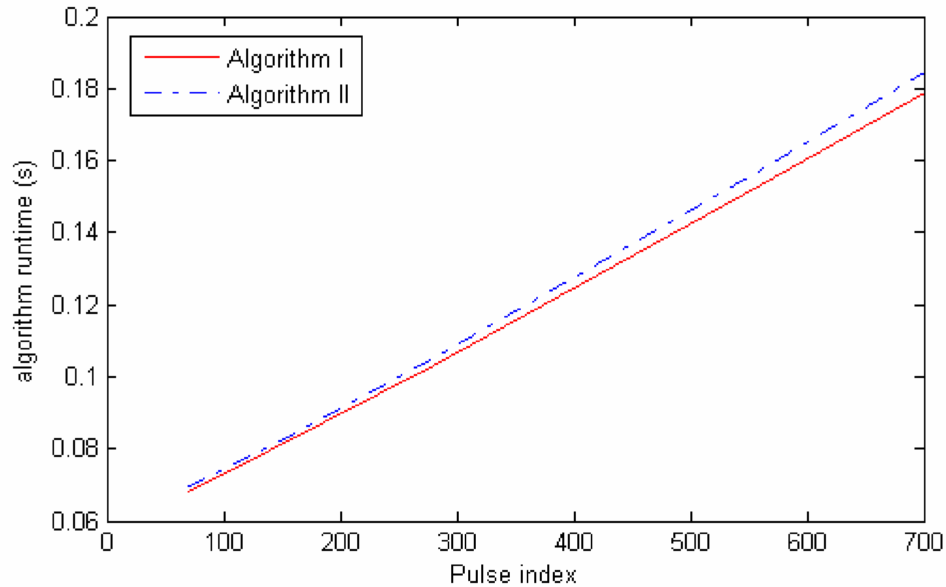

Figure 48. Implementation Runtime versus Pulse Number

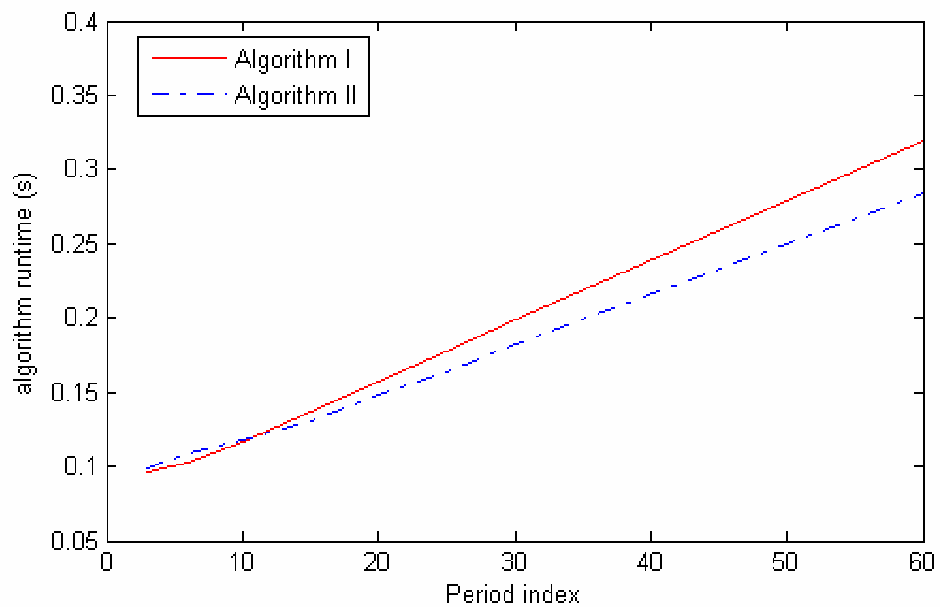

Figure 49. Implementation Runtime versus Period

Comment on simulations:

When Figures 44, 45, 46 and 47 are analyzed, the average estimation error of Algorithm I is slightly smaller than the average estimation error of the Algorithm II for the same staggered sequence with the same measurement and system noise levels.

Figure 48 shows the algorithm's runtimes in second with respect to total number of pulses with the same staggered sequence with a period of 7. According to the results shown on Figure 48, Algorithm I is slightly faster than the Algorithm II. This is due to the computational complexity of the Algorithm II which takes the DFT of the staggered sequence and then calculates the state vector and measurement matrix to initiate the filter. Figure 49 shows the algorithm's runtimes in second with respect to the period of the staggered PRI sequence. The same staggered PRI sequence is used with different periods. According to the results shown on Figure 49, if the period of the staggered PRI sequence increases, the runtime of the Algorithm I increase with respect to the runtime of the Algorithm II. This is due to the different methodologies used in the algorithms. For example, for a staggered PRI sequence with a period of *N,* measurement matrix *H* is *NxN* matrix in Algorithm I but in Algorithm II, it is *1xN* vector.

CHAPTER 5

CONCLUSION

In this study, different algorithms for tracking step (constant when the jitter on PRI is eliminated) and periodic staggered PRI sequences have been developed using Kalman Filter model. Step PRI sequence is analyzed according to the Kalman filter model [2]. Some useful simplifications are obtained in the model described for step PRI case. Two algorithms with different methodologies have been developed for periodic staggered PRI sequences.

At the beginning, Kalman filter model is described and the general Kalman filter time update and measurement update equations are analyzed. Since a staggered PRI sequence case can be considered as a periodic discrete event process, detailed information is given about discrete Kalman filter model.

The algorithms developed for periodic staggered sequences consist of two parts: detection of period of the staggered PRI sequence and Kalman filter model with discrete Fourier representation (DFT) of the staggered PRI sequence [4] and without DFT of the staggered PRI sequence in other words in time domain. The detection of period part determines the period of the sequence and obtains a period of data for Kalman filter model which is used to predict the pulse repetition interval. Moreover, another algorithm is developed for tracking step pulse repetition interval sequence and some useful Kalman filter time update and measurement update equations are obtained for this type of PRI agility. Methodologies of these estimation algorithms are explained and some simulations are performed displaying the performances of the algorithms developed for periodic staggered PRI sequences.

Robustness of the algorithms to missing pulses, which can occur due to the failure of the measuring apparatus to detect or receive a pulse in the measured data (measured TOAs) is also considered while developing the algorithms. Robustness of the algorithms to missing pulses is obtained by applying some gating techniques [18, 19] to measured TOAs. The algorithms have been developed according to the cumulative jitter model [10, 11].

In order to check the performance of the algorithms (Algorithm I and Algorithm II) developed for staggered PRI sequences, the received data is needed. In other words, the measured time of arrival values must be generated. So the algorithms were tested using simulated data. The simulated data consists of measurement noise, system noise and missing pulses. The aim of the simulations is to investigate PRI estimation performances of the algorithms in the case no missing pulse and in the presence of missing pulses. Results acquired from different parameters (true, measured and estimated PRI values) are plotted on the same graph to increase the comprehension. When the simulations performed for the algorithms are considered, the results can be summarized as below:

- The implementation time (CPU usage) of Algorithm II (Algorithm II predicts the PRI values with taking Fourier representation of the staggered PRI sequence) is slightly more than Algorithm I (Algorithm I does not take the discrete Fourier transform of the periodic staggered PRI sequence to calculate the state vector, initial prediction error covariance matrix and measurement matrix).
- The complexity of the algorithms increases with period (level) of the staggered PRI sequence. For high periods, the Algorithm II is a bit faster than the Algorithm I.
- The PRI estimation performance of the Algorithm I is slightly better than the second one in the presence of the same staggered sequence and noise levels. This is due to the different methodologies used in the algorithms, In Algorithm II, periodic staggered PRI sequence is represented in discrete Fourier transform to initiate prediction part of the

algorithm. On the other hand, Algorithm I use received PRI values directly to initiate prediction part of the algorithm.

- In each algorithm, estimated PRI values converge to the actual PRI values but the convergence in Algorithm I is slightly better than the one in Algorithm II. This is also due to the methodologies used in the algorithms.
- Both algorithms are robust to the effects of missing pulses
- Detection of the period part in both algorithms performed well. Correct period was found even in the presence of missing pulses.

The methodologies used in the algorithms, the level of the staggered PRI sequence and the number of pulses in the received data are important factors determining the computational time of the algorithms. When Algorithm I and Algorithm II are compared, the Algorithm II is slightly faster than the Algorithm I for high periods of staggered PRI sequences. However, the estimation performance of the Algorithm I is slightly better than the performance of the Algorithm II as it can be observed from the simulation results.

REFERENCES

[1] L. J. Levy, “The Kalman Filter: Navigation’s Integration Workhorse”, Transaction of Applied Physics Laboratory of the Johns Hopkins University, 1997

[2] G. Welch and G. Bishop, “An Introduction to Kalman Filter”, Transaction of Department of Computer Science University of North Carolina, Chapel Hill, April 2004

[3] D. Simon, “Kalman Filtering”, Transaction of Innovative Software, June 2001

[4] M. Hock, “Kalman Filter Predictor and Initialization Algorithm for PRI Tracking”, Transaction of United States Naval Research Laboratory, June 1998

[5] P.S. Maybeck, “Stochastic Models, Estimation and Control”, Academic Press Inc., London, 1979

[6] R. G. Brown and Patrick Y. C. Hwang, “Introduction to Random Signals and Applied Kalman Filtering”, John Wiley & Sons, Inc., New York, 1997

[7] M. S. Grewal, “Kalman Filtering: Theory and Practice Using MATLAB”, John Wiley & Sons, Inc., New York, 2001

[8] K. V. Ramanchandra, “Kalman Filtering Techniques for Radar Tracking”, Marcel Dekker, Inc., New York, 2000

[9] H. W. Sorenson, “Kalman Filtering: Theory and Application”, IEEE Press, New York, 1985

[10] S.D. Howard and S. Sirianunpiboon, “A Robust Recursive Filter for Radar Pulse Train Parameter Estimation and Tracking “, accepted for publication, IEEE TAES, 1996

[11] D. A. Gray, B. J. Slocumb and Stephan D. Elton, “Parameter Estimation for Periodic Discrete Event Processes”, in Proc. IEEE ICASSP’94, Vol IV, pp.93-96, 1994

[12] G. Noone, “Radar Pulse Train Parameter Estimation and Tracking using Neural Networks”, Proc. ANNES, 1995

[13] S. D. Elton and B. J. Slocumb, “A robust Kalman Filter for estimation and tracking of a class of periodic discrete event processes”,

[14] H. L Van Trees, “Detection, Estimation and Modulation Theory, Part I”, John Wiley & Sons, Inc., New York, 2001

[15] A. V. Oppenheim, R. W. Schafer and J. R. Buck, “Discrete-Time Signal Processing”, Prentice Hall, 1999

[16] B. Abraham and J. Ledolter, “Statistical Methods for Forecasting”, John Wiley and Sons, New York, 1983

[17] B. J. Slocomb and E. W. Kamen, “The pulse train PDA analysis and deinterleaving filter”, Proc. SPIE Vol. 3068, p. 296-307, July 1997

[18] Y Bar-Shalom and Xiao-Rong Li, “Multitarget-Multisensor Tracking: Principles and Techniques”, YBS Publishing, 1995

[19] S. Blackman, “Multiple-Target Tracking with Radar Applications”, Artech House, 1986

[20] “Electronic Warfare Tutorial”,http://ourworld.compuserve.com/homepages-/edperry/ewtutor1.htm, September 1996

[21] E. Arnaud and B. Cernushi, “Conditional Filters for Image Sequence Based Tracking”, IEEE Transactions on Image Processing, 2004

APPENDIX

COMPUTER PROGRAMS WRITTEN IN MATLAB

Algorithm I

This function accepts all parameters required for the algorithm and returns the estimated PRIs, average PRI estimation error and period of the staggered PRI sequence.

```
global dftsizerrorarray;
dftsizerrorarray=[];
runtime=0;
total_estimated_pri_set=zeros(1,350);
total_measured_pri_set=zeros(1,350);
run=10;

for i=1:run
   tic
   close all;
   one_period_pri_set=[60 70 90 70 100 150 170];
   period=length(one_period_pri_set);
   how_many_period=50;
   pulse_number=how_many_period*period;
   sigma_w=0.01;  % variance of noise w, mean of w=0
   sigma_v=0.6;  % variance of noise v, mean of v=0

   %In this part the toa values are simulated
   toa=[];
   toa=one_period_pri_set(1)+sigma_w*randn;
   k=2;
   for i=2:pulse_number
      if k==(period+1)
         k=1;
         toa(i)=toa(i-1)+one_period_pri_set(1)+sigma_w*randn;
      else
         toa(i)=toa(i-1)+one_period_pri_set(k)+sigma_w*randn;
      end
      k=k+1;
   end
```

```
%In this part the missing pulse indices are found according to the given
%ratio
missing_pri_percent=0;

total_missing_pri_indices_number=round(pulse_number*missing_pri_percent/100)
;
r=0;
k=0;
missing_pri_indices=[];
if total_missing_pri_indices_number>0
    missing_pri_indices=randint(1,1,[period+1,pulse_number-2]);
    for i=2:total_missing_pri_indices_number
    k=0;
    while k==0
        t=randint(1,1,[period+1,pulse_number-2]);
        if length(intersect(missing_pri_indices,t))==0
            missing_pri_indices=cat(2,missing_pri_indices,t);
            k=1;
        end
    end
    end
else
    missing_pri_indices=[];
end
missing_pri_indices=sort(missing_pri_indices);
%missing_pri_indices=[15 25 141 155 160 218 252 268];
%missing_pri_indices=[15 25 30 36 47 53 65 69 78 155 160 218 252 268]
%In this part the measured toa values are calculated
measured_toa=[];
t=1;
k=1;
if  length(missing_pri_indices)==0
    for i=1:pulse_number
        measured_toa(i)=toa(i)+sigma_v*randn;
    end
else
    for i=1:(pulse_number-length(missing_pri_indices))
        if k==missing_pri_indices(t)
            k=k+1;
            t=t+1;
            if t>length(missing_pri_indices)
                t=1;
            end
            if k==missing_pri_indices(t) % If there are consecutive missing pulses
                k=k+1;
            end
            measured_toa(i)=toa(k)+sigma_v*randn;
        else
```

```
        measured_toa(i)=toa(k)+sigma_v*randn;
      end
      k=k+1;
    end
  end
  true_pri=[];
  for i=1:how_many_period
    true_pri=[true_pri one_period_pri_set];
  end
  measured_pri_set=[];
  measured_pri_set(1)=measured_toa(1);
  for i=2:(pulse_number-length(missing_pri_indices))
    measured_pri_set(i)=measured_toa(i)-measured_toa(i-1);
  end

  %In this part period of the staggered sequence is calculated
  obtained_data_buffer_size=length(measured_pri_set);
  magic_matrix=zeros(2,obtained_data_buffer_size);
  for i=1:obtained_data_buffer_size
    magic_matrix(1,i)=i;
  end
  r=1;
  while r<obtained_data_buffer_size
    for i=r+1:obtained_data_buffer_size
      if ((measured_pri_set(i)<=(measured_pri_set(r)+4*(sigma_w+2*sigma_v)))
&& (measured_pri_set(i)>=(measured_pri_set(r)-4*(sigma_w+2*sigma_v))))
        magic_matrix(2,i-r)=magic_matrix(2,i-r)+1;
      end
    end
    r=r+1;
  end
  period=find(max(magic_matrix(2,:))==magic_matrix(2,:));
  period=period(1);
  stem(magic_matrix(1,:),magic_matrix(2,:),'.');
  xlabel('Estimates of the number of pulses per period (N)');
  ylabel('Number of hits');
  hold on;
  grid on;

  %In this part, the initial values and the state covariance matrix are calculated
  x=[]; %state vector
  x(1)=measured_toa(1);
  for i=2:period
    x(i)=measured_toa(i)-measured_toa(i-1);
  end
  x=x';
  lamda=x;
  value_noted=sigma_w^2+2*(sigma_v^2)
```

```
   C_lamda=zeros(period,period); %covariance matrxi of state vector
   C_lamda(1,1)=value_noted;
   C_lamda(1,2)=-sigma_v^2;
   for k=2:(period-1)
      C_lamda(k,k-1)=-sigma_v^2;
      C_lamda(k,k)=value_noted;
      C_lamda(k,k+1)=-sigma_v^2;
   end
   C_lamda(period,period-1)=-sigma_v^2;
   C_lamda(period,period)=value_noted;
   P=C_lamda;
   Q=0;
   estimated_pri_set=[];
   take_measurement_toa=[];
   l=1;
   Inn=[];
   n=1;
   k=1;
   z=0;
   for i=1:(how_many_period)
      R=zeros(period,period); %R measurement covariance matrix
      period_index=i;
      deneme=period*period_index-(period-1);
      for t=1:period
         R(t,t)=deneme*(sigma_w^2)+sigma_v^2;
         if t<period
            R((t+1):period,t)=deneme*(sigma_w^2);
            R(t,(t+1):period)=deneme*(sigma_w^2);
         end
         deneme=deneme+1;
      end
      H=ones(period,period);
      H=l*H;
      for c=1:period-1
         H(c,c+1:period)=H(c,c+1:period)-1;
      end
      predicted_measured_toa_array=[];
      predicted_measured_toa_array=H*x;
      a=0;
      d=1;
      while ((k>=n && k<(n+period)) &&(n+period-
1)<=obtained_data_buffer_size)
         for u=1:period
            measured_toa(k);
            predicted_measured_toa_array;
            if ((measured_toa(k)>=(predicted_measured_toa_array(u)-
4*(k*sigma_w+sigma_v)))&&(measured_toa(k)<=(predicted_measured_toa_array(
u)+4*(k*sigma_w+sigma_v))))
```

```
                a=a+1;
                take_measurement_toa(a)=measured_toa(k);
            elseif d==u
                d=d+1;
                H(u,:)=[];
                R(u,:)=[];
                R(:,u)=[];
            end
        end
        k=k+1;
        d=d+1;
    end
    n=n+a;
    k=n;
    S=H*P*H'+R;%Compute the covariance of the Innovation
    K=P*H'*inv(S); %Form the Kalman Gain Matrix
    Inn=take_measurement_toa'-H*x;
    x=x+K*Inn; %Update the state estimate
    take_measurement_toa=[];
    P=P-P*H'*inv(S)*H*P+Q; %Compute the covariance of the estimation error
    estimated_pri_set=[estimated_pri_set  x'];
    l=l+1;
  end
  total_estimated_pri_set= total_estimated_pri_set+estimated_pri_set;
  total_measured_pri_set=total_measured_pri_set+measured_pri_set;
  runtime=runtime+toc;
end
estimated_pri_set=total_estimated_pri_set/run;
measured_pri_set=total_measured_pri_set/run;

figure;
k=1:1:pulse_number;
plot(k,true_pri(1,:),'p',k,estimated_pri_set(1,:),'ro',k,measured_pri_set(1,:),'ko');
hold on;
grid on;
xlabel('Pulse Number');
ylabel('PRI Value');

figure;
plot(abs((true_pri(1,:)-measured_pri_set(1,:))),'-p');
hold on;
grid on;
plot(abs((true_pri(1,:)-estimated_pri_set(1,:))),'-r');
xlabel('Pulse Number');
ylabel('Average (absolute) PRI Error');

figure;
plot(abs((true_pri(1,:)-estimated_pri_set(1,:))),'ro');
```

```
xlabel('Pulse Number');
ylabel('Average (absolute) PRI Estimation Error');
dftsizerrorarray=abs((true_pri(1,:)-estimated_pri_set(1,:)));
runtime=runtime/run
```

Algorithm II

This function is used to implement the Kalman Filter Algorithm I, it accepts all parameters required for the algorithm and returns the the estimated PRIs, average PRI estimation error and period of the staggered PRI sequence.

```
global ffterrorarray;
runtime=0;
total_estimated_pri_set=zeros(1,92);
total_measured_pri_set=zeros(1,92);
total_true_pri_set=zeros(1,92);
run=1;
for i=1:run
   tic
   close all;
   one_period_pri_set=[240 290 350 410 240];
   period=length(one_period_pri_set);
   how_many_period=20;
   pulse_number=how_many_period*period;
   sigma_w=0.1;  %standard deviation of system noise w, mean of w=0
   sigma_v=0.9; %standard deviation of measurement noise mean of u=0
   %The missing pulse indices are found according to the given missing PRI ratio
   missing_pri_percent=8;

total_missing_pri_indices_number=round(pulse_number*missing_pri_percent/100;
   r=0;
   k=0;
   missing_pri_indices=[];
   if total_missing_pri_indices_number>0
      missing_pri_indices=randint(1,1,[period+1,pulse_number-2]);
      for i=2:total_missing_pri_indices_number
         k=0;
         while k==0
            t=randint(1,1,[period+1,pulse_number-2]);
            if length(intersect(missing_pri_indices,t))==0
               missing_pri_indices=cat(2,missing_pri_indices,t);
```

```
            k=1;
         end
      end
   end
else
   missing_pri_indices=[];
end
missing_pri_indices=sort(missing_pri_indices);
%In this part the toa values are simulated
toa=[];
toa=one_period_pri_set(1)+sigma_w*randn;
k=2;
for i=2:pulse_number
   if k==(period+1)
      k=1;
      toa(i)=toa(i-1)+one_period_pri_set(1)+sigma_w*randn;
   else
      toa(i)=toa(i-1)+one_period_pri_set(k)+sigma_w*randn;
   end
   k=k+1;
end
%In this part the measured toa values are calculated
measured_toa=[];
t=1;
k=1;
if  length(missing_pri_indices)==0
   for i=1:pulse_number
      measured_toa(i)=toa(i)+sigma_v*randn;
   end
else
   for i=1:(pulse_number-length(missing_pri_indices))
      if k==missing_pri_indices(t)
         k=k+1;
         t=t+1;
         if t>length(missing_pri_indices)
            t=1;
         end
         if k==missing_pri_indices(t) % If there are consecutive missing pulses
            k=k+1;
         end
         measured_toa(i)=toa(k)+sigma_v*randn;
      else
         measured_toa(i)=toa(k)+sigma_v*randn;
      end
      k=k+1;
   end
end
true_pri_set=[];
```

```
    for i=1:how_many_period
        true_pri_set=[true_pri_set one_period_pri_set];
    end
    measured_pri_set=[];
    measured_pri_set(1)=measured_toa(1);
    for i=2:(pulse_number-length(missing_pri_indices))
        measured_pri_set(i)=measured_toa(i)-measured_toa(i-1);
    end
    R=sigma_w^2+2*(sigma_v^2);

    for s=1:length(missing_pri_indices)

true_pri_set(missing_pri_indices(s))=true_pri_set(missing_pri_indices(s))+true_pri
_set(missing_pri_indices(s)+1);
        true_pri_set(missing_pri_indices(s)+1)=true_pri_set(missing_pri_indices(s));
    end
    for i=1:length(missing_pri_indices)
     true_pri_set(missing_pri_indices(i)+1)=0;
    end
    d=find(true_pri_set);
    true_pri_set=true_pri_set(1,d);
    %In this part period is calculated
    obtained_data_buffer_size=length(measured_pri_set);
    magic_matrix=zeros(2,obtained_data_buffer_size);
    for i=1:obtained_data_buffer_size
        magic_matrix(1,i)=i;
    end
    r=1;
    while r<obtained_data_buffer_size
        for i=r+1:obtained_data_buffer_size
            if ((measured_pri_set(i)<=(measured_pri_set(r)+4*(
sigma_w+2*sigma_v))) && (measured_pri_set(i)>=(measured_pri_set(r)-6*(
sigma_w+2*sigma_v))))
                magic_matrix(2,i-r)=magic_matrix(2,i-r)+1;
            end
        end
        r=r+1;
    end
    period=find(max(magic_matrix(2,:))==magic_matrix(2,:));
    period=period(1);
    stem(magic_matrix(1,:),magic_matrix(2,:),'.');
    xlabel('Estimates of the number of pulses per period (N)');
    ylabel('Number of hits');
    hold on;
    grid on;

    first_period_pri_set=[];
    for i=1:period
```

```
    first_period_pri_set=[first_period_pri_set measured_pri_set(i)];
end
estimated_pri_set=[];
p=period;
omega=ones(1,p)/p;
for k=1:((p-1)/2)
    row_array_cos=[2/p]
    row_array_sin=[0]
    for i=1:(p-1)
        value_cos=2*cos(2*pi*i*k/p)/p;
        row_array_cos=cat(2,row_array_cos,value_cos);
        value_sin=2*sin(2*pi*i*k/p)/p;
        row_array_sin=cat(2,row_array_sin,value_sin);
    end
    omega=cat(1,omega,row_array_cos);
    omega=cat(1,omega,row_array_sin);
end
if rem(p,2)==0
    row_array_cos=[1/p];
    for i=1:(p-1)
        value_cos=cos(pi*i)/p;
        row_array_cos=cat(2,row_array_cos,value_cos);
    end
    omega=cat(1,omega,row_array_cos);
end
lamda=first_period_pri_set';
value_noted=sigma_w^2+2*(sigma_v^2);
C_lamda=zeros(p,p);
C_lamda(1,1)=value_noted;
C_lamda(1,2)=-sigma_v^2;
for k=2:(p-1)
    C_lamda(k,k-1)=-sigma_v^2;
    C_lamda(k,k)=value_noted;
    C_lamda(k,k+1)=-sigma_v^2;
end
C_lamda(p,p-1)=-sigma_v^2;
C_lamda(p,p)=value_noted;
C=omega*C_lamda*(omega');
P=C; %initial_estimate_error_covariance
Q=0;
xhat=omega*lamda %initial state estimate
i=1;
k=1;
control=0;
deneme=0;
while i<obtained_data_buffer_size+1
    H=calculate_H(k,period);
    S=H*P*H'+R; %Compute the covariance of the Innovation
```

```
        K=P*H'*inv(S); %Form the Kalman Gain Matrix
        Inn=measured_pri_set(i)-H*xhat;
        if (H*xhat+4*sqrt(R)<measured_pri_set(1,i))
            xhat=xhat;
            measurement_prediction=measured_pri_set(1,i);
            k=k+1;
            control=1;
            deneme=deneme+1;
        end
        if control==1
            estimated_pri_set=[estimated_pri_set  measurement_prediction];
            P=P+Q;
            control=0;
        else
            xhat=xhat+K*Inn; %Update the state estimate
            estimated_pri_set=[estimated_pri_set  H*xhat];
            P=P-P*H'*inv(S)*H*P+Q; %Compute the covariance of the estimation
error
        end
        i=i+1;
        k=k+1;
    end
    total_estimated_pri_set= total_estimated_pri_set+estimated_pri_set;
    total_measured_pri_set=total_measured_pri_set+measured_pri_set;
    total_true_pri_set=total_true_pri_set+true_pri_set;
    runtime=runtime+toc;
end
estimated_pri_set=total_estimated_pri_set/run;
measured_pri_set=total_measured_pri_set/run;
true_pri_set=total_true_pri_set/run;
P_array{1,:};
figure;
k=1:1:obtained_data_buffer_size;
plot(k,true_pri_set(1,:),'p',k,estimated_pri_set(1,:),'ro',k,measured_pri_set(1,:),'ko');
hold on;
grid on;
xlabel('Pulse Number');
ylabel('PRI Value');
figure;
plot(abs((true_pri_set(1,:)-measured_pri_set(1,:))),'-p');
hold on;
grid on;
plot(abs((true_pri_set(1,:)-estimated_pri_set(1,:))),'-ro');
xlabel('Pulse Number');
ylabel('Average (absolute) PRI Error');
figure;
plot(abs((true_pri_set(1,:)-estimated_pri_set(1,:))),'r-');
xlabel('Pulse Number');
```

```
ylabel('Average (absolute) PRI Estimation Error');
ffterrorarray=abs((true_pri_set(1,:)-estimated_pri_set(1,:)));
runtime=runtime/run
```

<u>calculate_H</u>

This function is used to calculated the measurement matrix according to the found period of the staggered PRI sequence

```
function [H_calculated]= calculate_H(l,periodu)
kkk=l-1;
period=periodu;
if period==1
   disp('This is not a staggered PRI sequence');
else if rem(period,2)==0
       array=[1];
       for i=1:period/2
           value=[cos(2*pi*kkk*i/period) sin(2*pi*kkk*i/period)];
           array=cat(2,array,value);
       end


   else
       array=[1];
       for i=1:(period-1)/2
       value=[cos(2*pi*kkk*i/period) sin(2*pi*kkk*i/period)];
       array=cat(2,array,value);
       end
   end
end
if rem(period,2)==0
last_index_of_array=max(size(array));
array=array(1,1:last_index_of_array-1);
end
H_calculated=array;
```