Algorithms for Radar Signal Deinterleaving

BITS Pilani
Hyderabad Campus

Presented by
Gagandeep Kaur (2014H140042H)

Contents

- Background
- TOA based deinterleaving
- Algorithms
- Constant PRI Deinterleaving
- Jitter and Stagger PRI Deinterleaving
- Deinterleaving System for constant, stagger, and jitter PRI
- Simulation Results

Background

Introduction

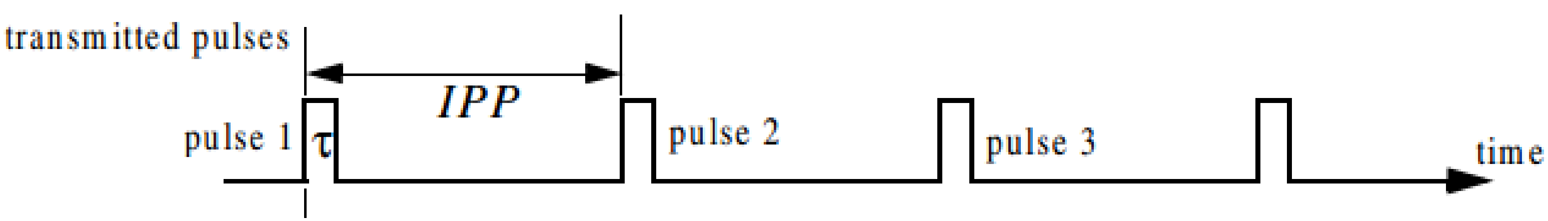
- Radar (Radio Detection and Ranging) is a remote sensing equipment and is considered the greatest advance in sensing remote objects since the telescope was invented.

- It is an object detection and tracking system that uses radio waves to determine the range, angle, or velocity of an object.

The concept of PRI

- Pulsed radars transmit and receive a train of modulated pulses.
- Pulsed radar waveforms can be completely defined by the following:
  - carrier frequency
  - pulse width
  - pulse repetition interval

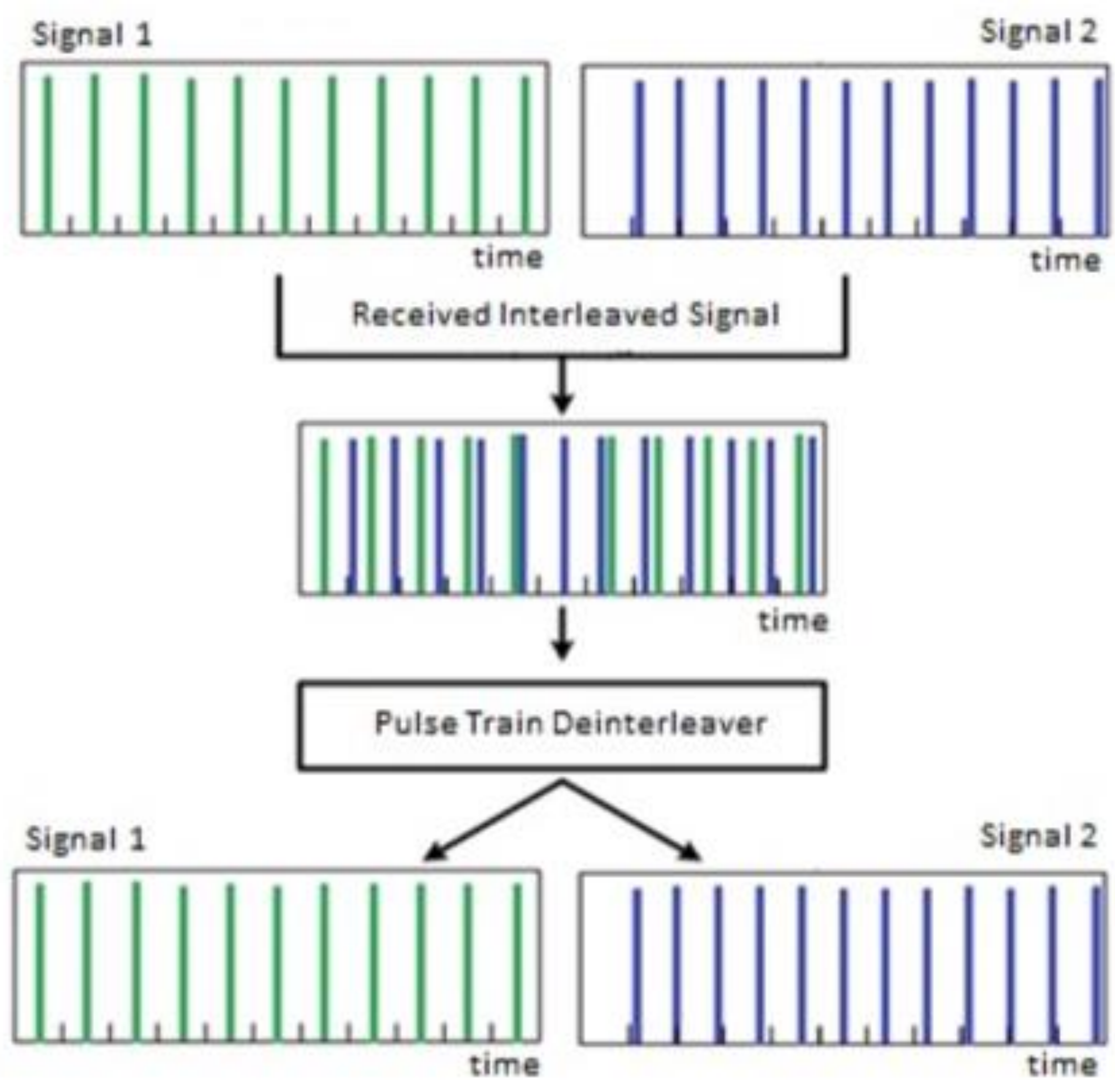
- The Inter Pulse Period (IPP) is referred to as the Pulse Repetition Interval (PRI)



Concept of Interleaving and Deinterleaving (1/2)

- In fields such as ‘radar signal detection’, a passive receiver observes an interleaved pulse train, which is a composite signal comprised of many unknown sources with different pulse repetition intervals (PRIs). This process is known as interleaving.
- It limits the capabilities of reconnaissance systems to identify and locate the threats.
- De-interleaving is the process of isolating the pulses of a single or multiple emitters from a pulse stream containing pulses from two or more signals. This enables users to focus on data from a single emitter at a time, rather than a mixture of data from several emitters.

Concept of Interleaving and Deinterleaving (1/2)



Why Deinterleave

- Goal: classify radar signals by their unique characteristics
- Goal: use these characteristics to:
    - ➢ identify enemy radars operating in the environment
    - ➢ determine their location or direction
    - ➢ inform friendly forces about their threats

Emitter Parameters used in Deinterleaving and Identification

		Pulse Sequence Deinterleaving	Emitter Identification
Measured	Frequency	Very Useful	Very Useful
	Angle of Arrival	Very Useful	Not Useful
	Pulse Width	Very Useful	Some Use
	Pulse Amplitude	Some Use	Not Useful
	Time of Arrival	Very Useful	Not Useful
Derived	Pulse Repetition Interval	Very Useful	Very Useful
	PRI Type	Very Useful	Very Useful
	Antenna Scan Period and Type	Not Useful	Very Useful
	Beam Width / Lobe Duration	Not Useful	Very Useful

Motivation

Modern day electronic battlefield is a very complex scenario

Challenges faced in identifying/ classifying the sources as seen by receiver.

- The signal received in an EW system is a composite signal coming from various sources as mentioned above.

- In addition, noise can make the analysis of the received signals more complex.

- If the emitters are operating on different frequencies, frequency based clustering can be employed in the reconnaissance part of the EW system.

However, radar emitters some sort of frequency agility like frequency hopping. This renders frequency based analysis ineffective in sorting the various signal sources.

Therefore, we look at other parameters available for deinterleaving, like AoA, ToA, Pulse Width, Pulse Amplitude, etc.

- Angle of Arrival (AoA) is a very reliable parameter; however, its measurement is generally a very complex procedure and not available with many of the ESM systems.

- Out of the remaining pulse parameters, Time of Arrival (ToA) is a very reliable parameter.

ToA based deinterleaving methods can be further classified into ‘interval based (PRI based separation)’ and ‘event detection’ based. PRI is a very useful parameter and is derived by performing some transformation on ToA.

References & Bibliography

1.	KEN'ICHI NISHIGUCHI and MASAAKI KOBAYASHI, 'Improved Algorithm for Estimating Pulse Repetition Intervals' in IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS VOL. 36, NO. 2 APRIL 2000
2.	Udaya kumar, Dhananjayulu, Anil kumar, 'Deinterleaving of Radar Signals and its Parameter Estimation in EW Environment' in International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 9, September 2014
3	P. Rakesh Kumar, N.Ramya, V.Dhananjayulu, 'De-Interleaving and Identification of Pulsed Radar Signals Using ESM Receiver System' in in International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 7, July 2015
4	Drake et a1, United States Patent, Patent number: US 8,102,297 B2
5	Navid Daryasafar and Hamid Dehghani, 'Studying an Improved Interval-Only Algorithm for the De-Interleaving of Radar Pulses' in J. World. Elect. Eng. Tech. 4(1): 13-20, April 25, 2015
6	Manish Gupta*, Hareesh G., and Arvind Kumar Mahla, 'Electronic Warfare: Issues and Challenges for Emitter Classification' in Defence Science Journal, Vol. 61, No. 3, May 2011
7	Jingyao Liu, Huadong Meng, and Xiqin Wang, 'A NEW PULSE DEINTERLEAVING ALGORITHM BASED ON MULTIPLE HYPOTHESIS TRACKING' in IEEE Explore, November 2009
8	Shruti Sridharan, Dr.NNSSRK Prasad, Riya George, M Brindha, 'IMPROVED PULSE REPETITION INTERVAL (PRI) DEINTERLEAVING FOR ELECTRONIC SUPPORT MEASURE (ESM) RECEIVER' in ISSN(PRINT):2394-3408,(ONLINE):2394-3416,VOLUME-2,ISSUE-3,2015

Contd..

9	YUJUN KUANG, QINGBO SHI, QIANBIN CHEN, LI YUN, and KEPING LONG, 'A Simple Way to Deinterleave Repetitive Pulse Sequences' in 7th WSEAS Int. Conf. on MATHEMATICAL METHODS and COMPUTATIONAL TECHNIQUES IN ELECTRICAL ENGINEERING, Sofia, 27-29/10/05 (pp218-222)
10	Douglas E. Driscoll & Stephen D. Howard, 'THE DETECTION OF RADAR PULSE SEQUENCES BY MEANS OF A CONTINUOUS WAVELET TRANSFORM' in 0-7803-5041-3/99 $10.00 0 1999 IEEE
11	Ken'ichi Nishiguchi, 'Time-period Analysis for Pulse Train Deinterleaving' in Trans. of the Society of Instrument and Control Engineers Vol.E-4, No.1, 68/78 (2005)
12	MEHMET KADİR ASLAN, 'EMITTER IDENTIFICATION TECHNIQUES IN ELECTRONIC WARFARE' , A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF MIDDLE EAST TECHNICAL UNIVERSITY
13	Richard G. Wiley, 'The Interception and Analysis of Radar Signal'
14	H.S.SHAHHOSEIN, A.NASEU, M.NADERI, 'A New Matrix Method FOP Pulse Train Identification' in IEEE MELECON 2002, May 7-9,2002, Cairo, EGYPT
15	https://en.wikipedia.org
16	Rogers, J. A. V., 1985, "ESM Processor System for High Pulse Density Radar Environments" IEE Proceedings, vol. 132, pp 621-625

Contd..

17	Avionics Department of the Naval Air Warfare Center Weapons Division, 1993, “Electronic Warfare and EW Systems”
18	ASLANBEK M KILINÇARSLAN, A STUDY ON IDENTIFICATION OF RADAR EMITTERS, A MASTER'S THESIS in Electrical and Electronics Engineering, Atılım University
19	Yan Mao, Jun Han, Guohua Guo, Xu Qing, ‘An Improved Algorithm of PRI Transform’ in 978-0-7695-3571-5/09 $25.00 © 2009 IEEE
20	D. Milojevic and B. Popovic, “Improved algorithm for the deinterleaving of radar pulses,” in Radar and Signal Processing, IEE Proceedings F, vol. 139, Feb. 1992, pp. 98–104.
21	AW Ata'a, SN Abdulla, “Deinterleaving of radar signals and PRF identification algorithms”, in Radar, Sonar & Navigation, IET, 2007
22	http://www.pitt.edu/~jdnorton/teaching/HPS_0410/chapters/spacetime/

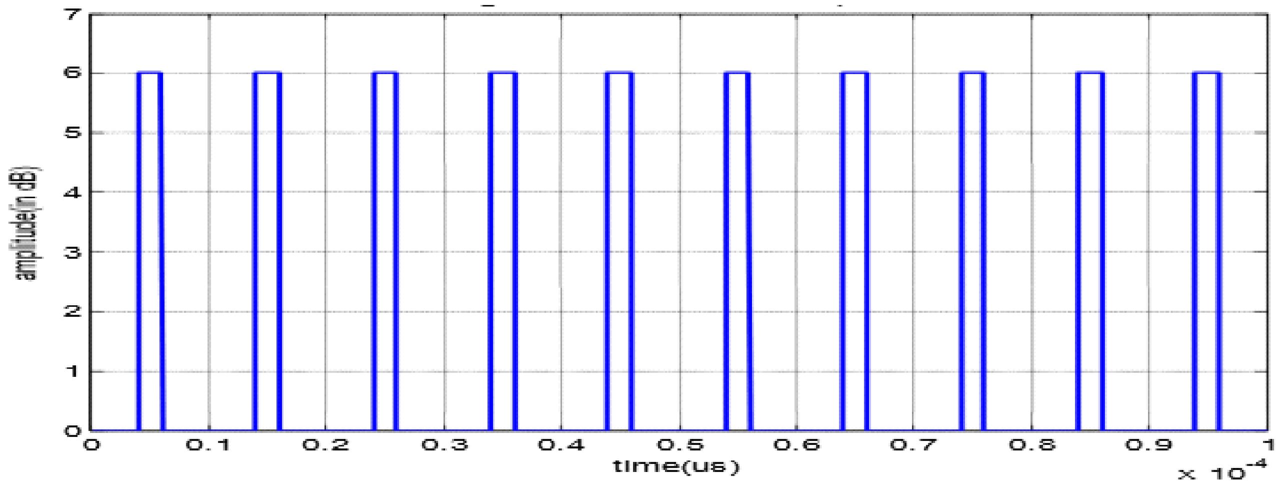
- Reference [1] and [19] have been followed the most

Classification of Pulsed Radar PRI

- CONSTANT
- STAGGERED
- JITTER

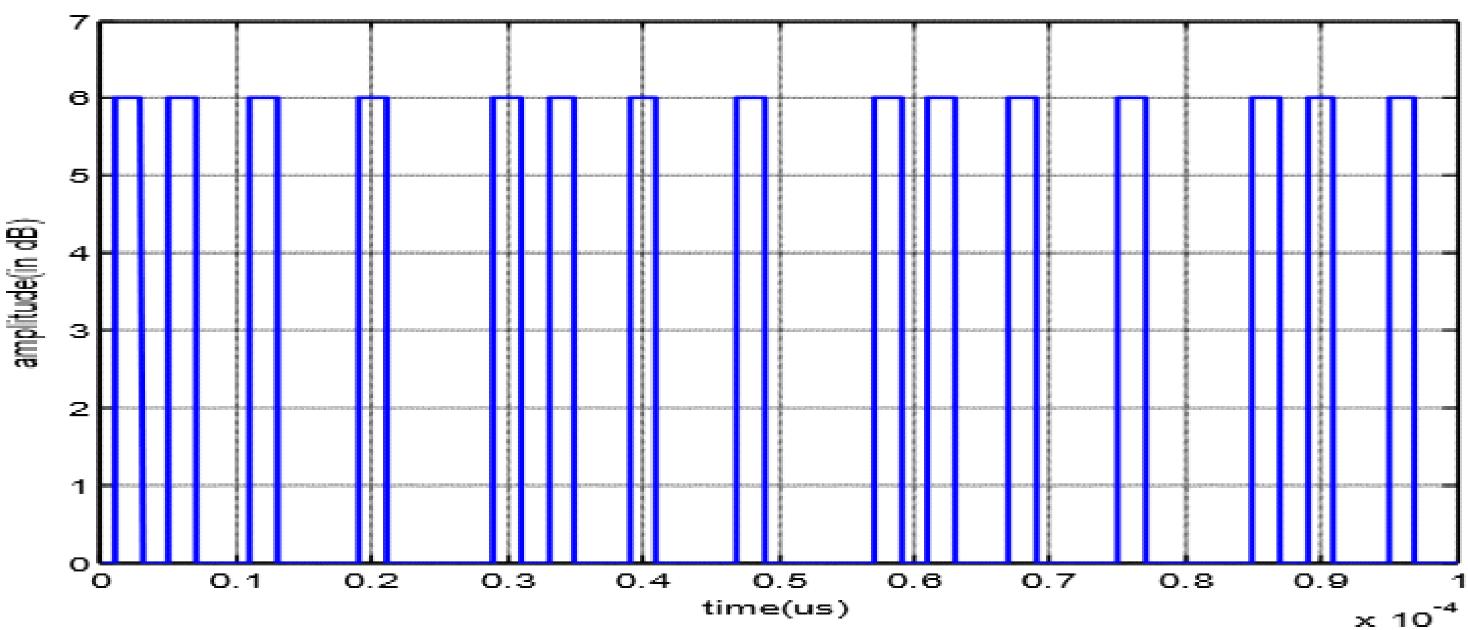
Constant PRI

- Peak variations in PRI are less than 1% of the mean PRI.
- These variances are assumed to be incidental.



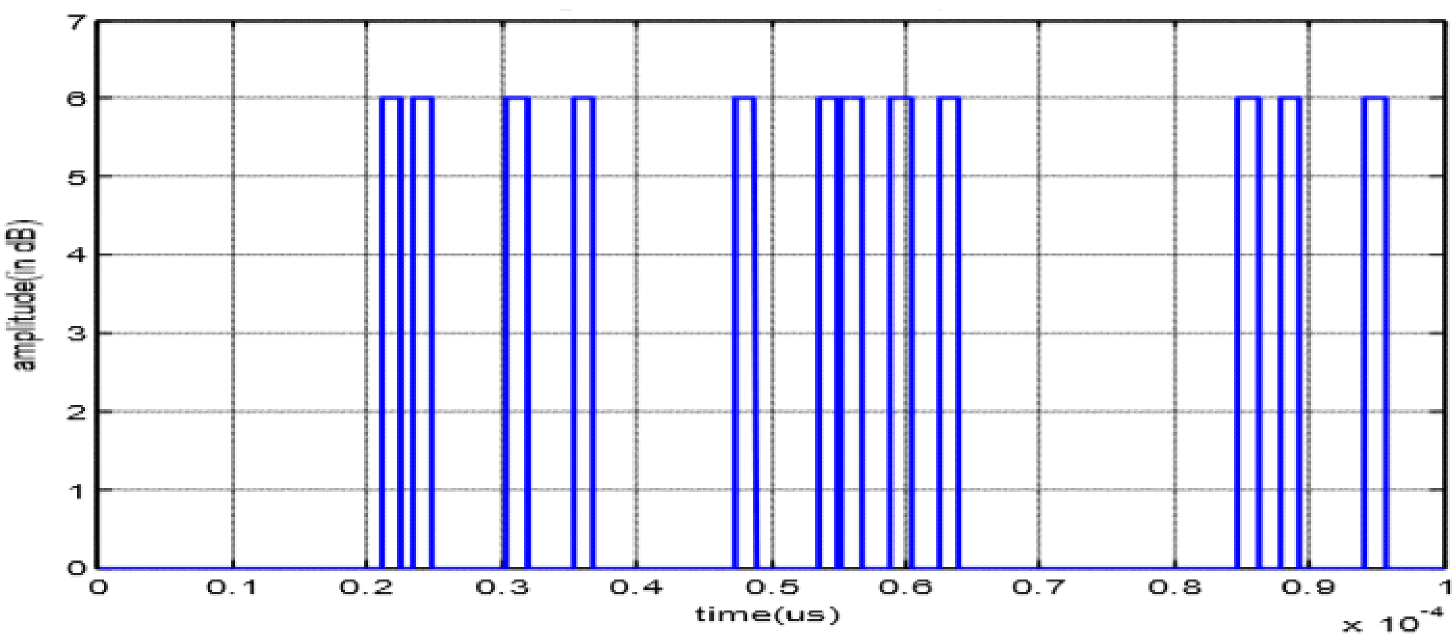
Staggered PRI

- Use of two or more PRIs in a fixed sequence
- PRI is changed from pulse to pulse



Jitter PRI

large variations in PRI up to 30% of the mean PRI



Conditions Affecting Deinterleaving

- Pulse overlap
- Dropped pulses
- Extraneous pulses (multipath)
- False alarms
- Receiver blanking

General procedure for TOA based deinterleaving

- The deinterleaving algorithms generally operate in two stages
  - PRI identification and estimation
  - PRI sorting

- Deinterleaving steps
  1. Take a block of pulses
  2. Eliminate pulses from the known emitters and update the emitter list
  3. Run the deinterleaving algorithm and find the most evident emitter
  4. Eliminate the pulses from this emitter and update the emitter list
  5. Repeat Steps 3 and 4 until the number of remaining pulses is less than a threshold value or no new emitter can be found
  6. Restart the process

TOA interval based deinterleaving Techniques

- Autocorrelation
- Histogramming
- Improvements in Histogramming
  - CDIF
  - SDIF
- Sequence Search
- Wavelet Based Detector
- PRI Transform Related
  - PRI Transform
  - Modified PRI Transform
  - PRI Map

Algorithms

	Constant PRI	Jitter PRI	Stagger PRI
PRI Estimation	SDIF + Sequence Search	Modified PRI Transform	Stagger Analysis
	PRI Transform	SDIF with overlapped bins	
Pulse Sorting	Sliding Window sort	Sequence Search	Sliding Window sort

Algorithms for Constant PRI Estimation

- SDIF + Sequence Search

- Autocorrelation

Base of all PRI estimation techniques

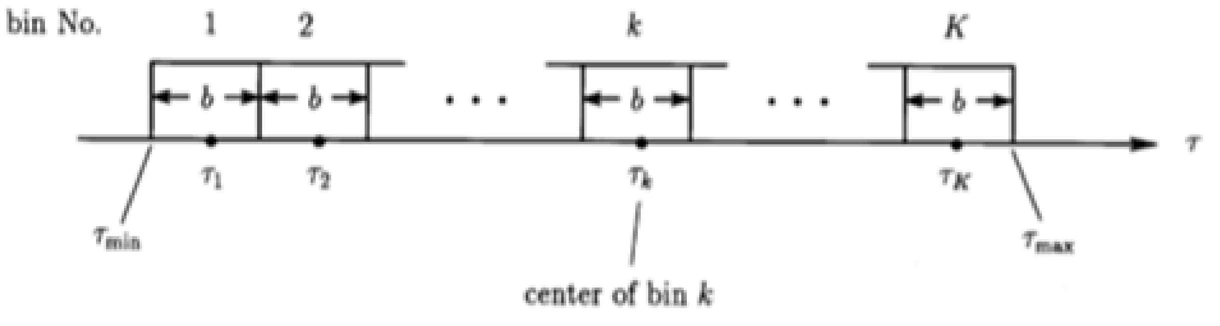
- PRI Transform

SEQUENCE SEARCH

- Assumes an initial PRI estimate
- The algorithm starts from the first pulse in the buffer (at t1), and then looks for another pulse at (t1 + PRI)
- If a pulse is not found, the algorithm restarts with the second pulse
- If a pulse is found, the algorithm continues with this pulse and looks for the next pulse.
- It is generally not used as a PRI estimation algorithm, but a secondary algorithm for sorting out pulses of a particular PRI after PRI estimation.

Autocorrelation

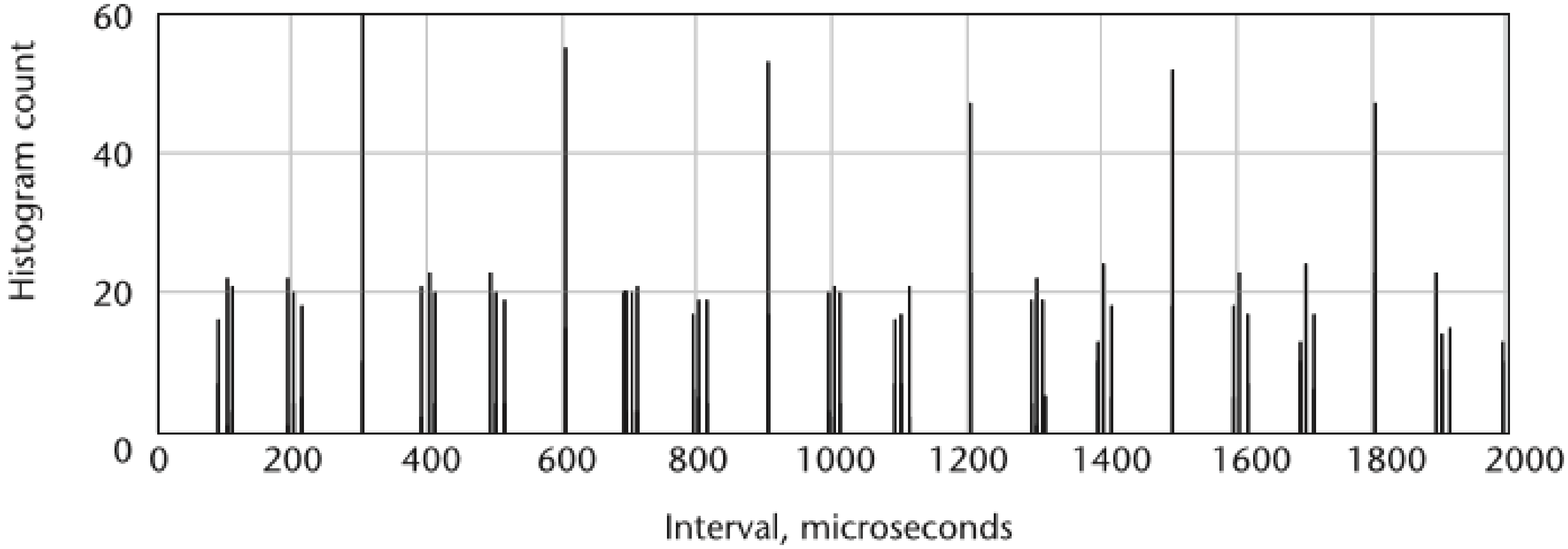
- histogram of pulse intervals made by considering the intervals between all pulse pairs (usually some maximum interval is specified).



$$\sum_{\{(m,n);\, \tau_K - \frac{b}{2} < t_n - t_m \leq \tau_K + \frac{b}{2}\}} 1$$

Delta T Histogram example

- Stagger PRI
- Sub periods : 95 us, 100 us, and 105 us
- Stagger PRI: 300 us



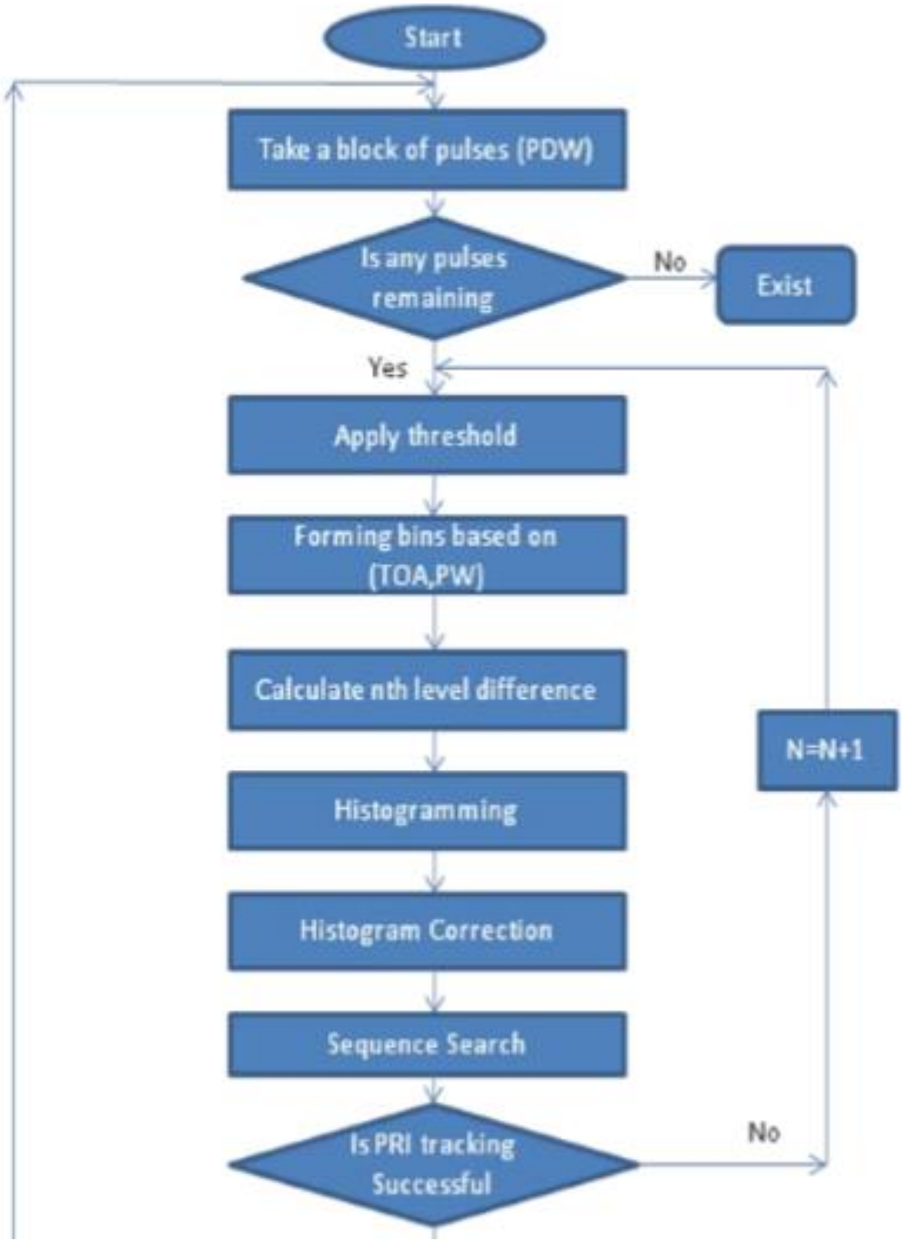
Comments

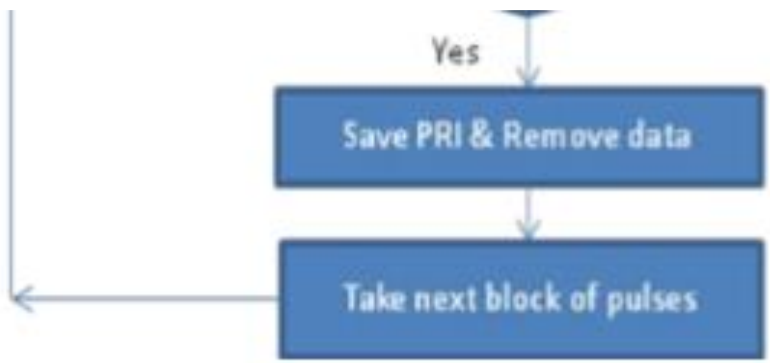
- Affected by subharmonics of PRIs
- Techniques based on autocorrelation to avoid subharmonics:
  - CDIF Histogram
  - SDIF Histogram
  - PRI Transform

S-DIF

- only one level histogram is used at a time
- Assume that N pulses are collected over a particular time period
- The difference between the TOA for adjacent pulses is calculated resulting in N−1 values
- $dj = |TOA(j) - TOA(j+1)|$
- After detection of a PRI, sequence search follows and pulses belonging to a particular PRI are removed.

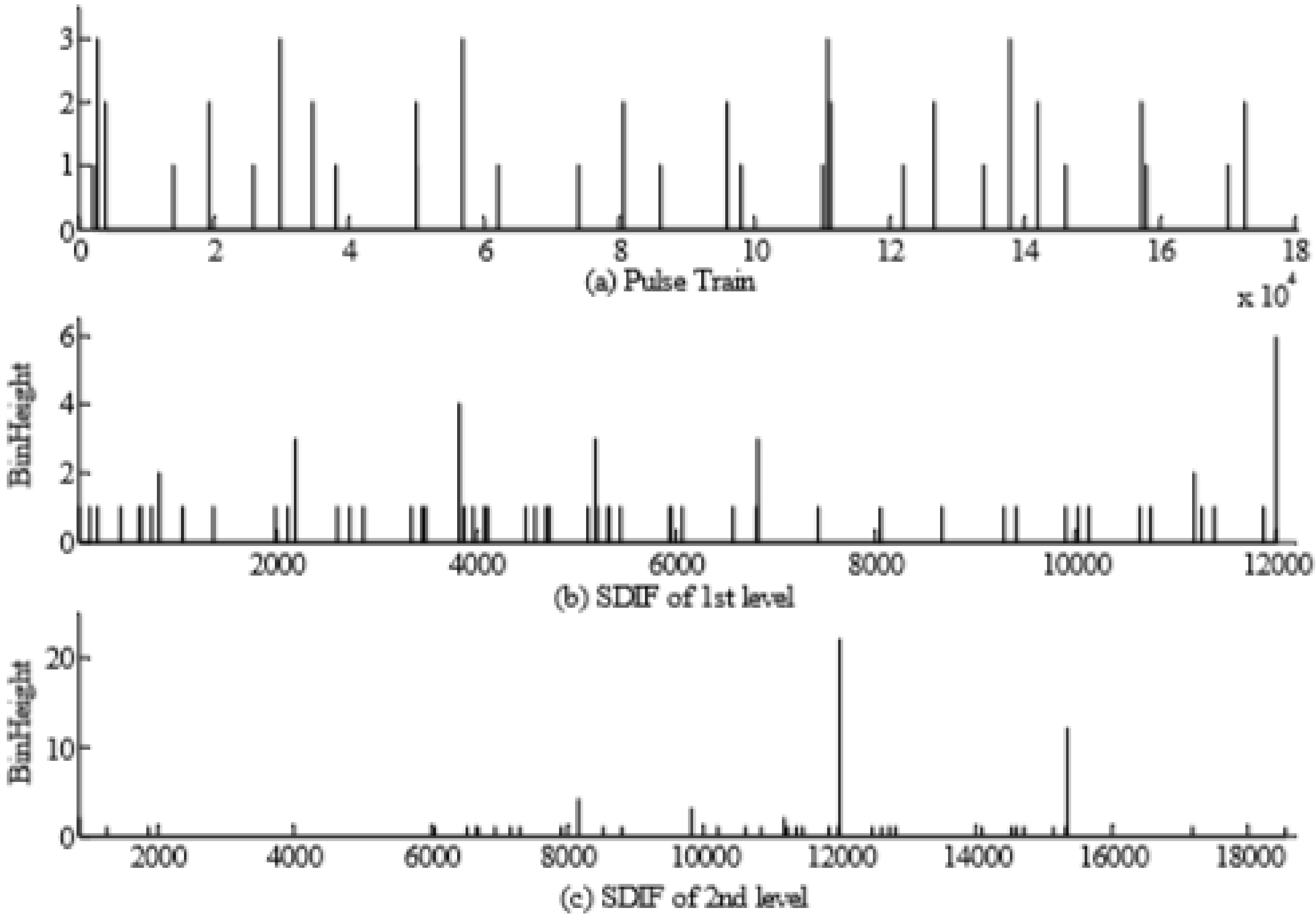
SDIF and Sequence Search





Yes
Save PRI & Remove data
Take next block of pulses

Example SDIF

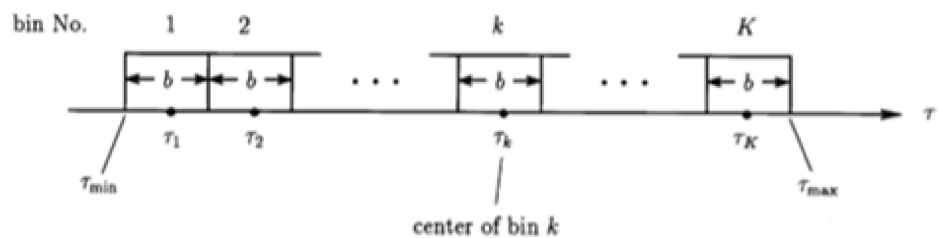


Comments on SDIF

- SDIF histogram tries to suppress subharmonic detection of PRI; however, sometimes it fails.
- SDIF can be improved by including a subharmonic check by thresholding the number of missing pulses.
- SDIF can help in the detection of stagger pulses as well.

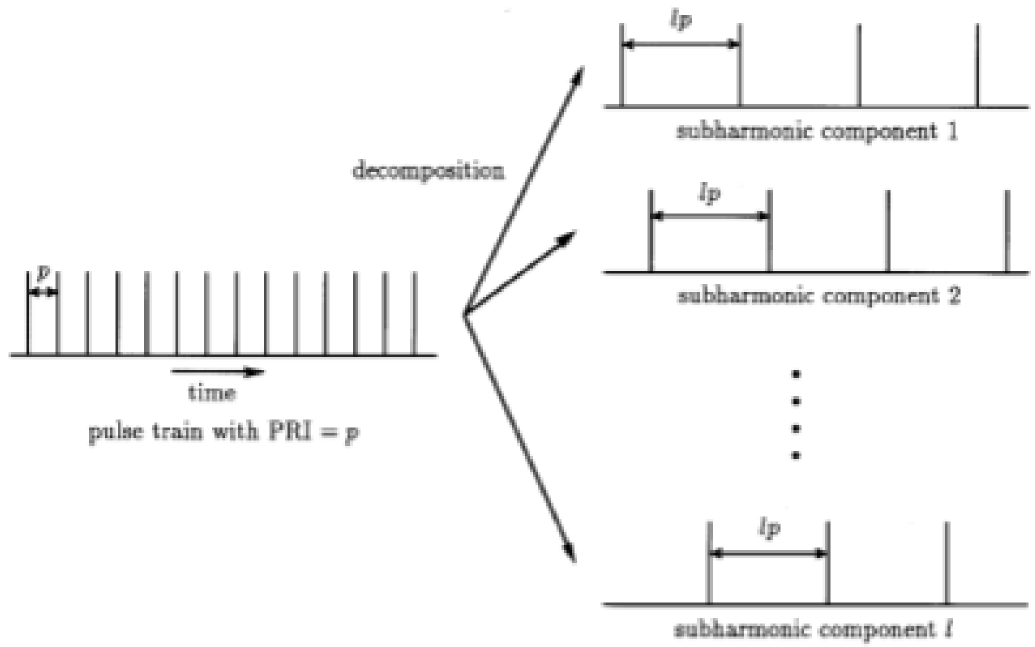
PRI Transform

- $[\tau_{min}, \tau_{max}]$ be the range of the PRI to be investigated. This PRI range is separated into K intervals (bins).
- $b=(\tau_{max}-\tau_{min})/K$
- $\tau_k=(k-1/2)b+\tau_{min}$



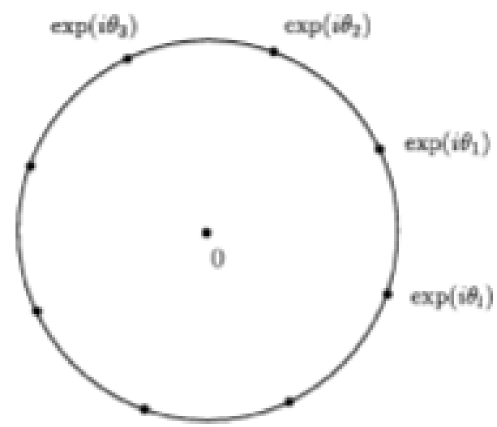
$$\sum_{\{(m,n);\, \tau_k-\frac{b}{2}<t_n-t_m\le\tau_k+\frac{b}{2}\}} \exp\left[\frac{2\pi i t_n}{t_n-t_m}\right]$$

- suppresses the subharmonics in the autocorrelation function almost completely



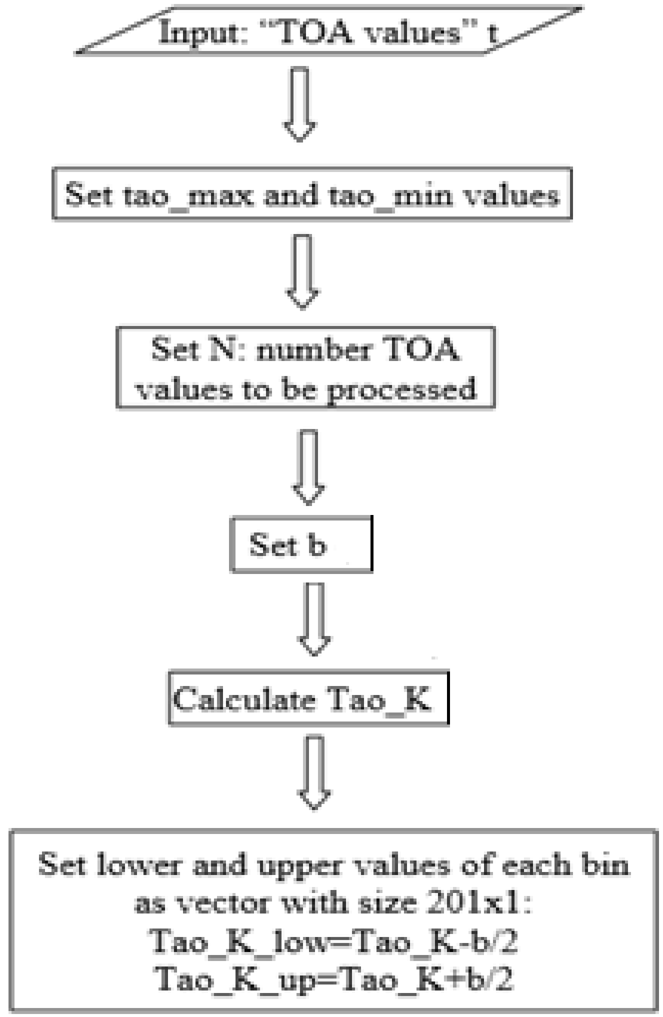
p
time
pulse train with PRI = p
decomposition
lp
subharmonic component 1
lp
subharmonic component 2
lp
subharmonic component l

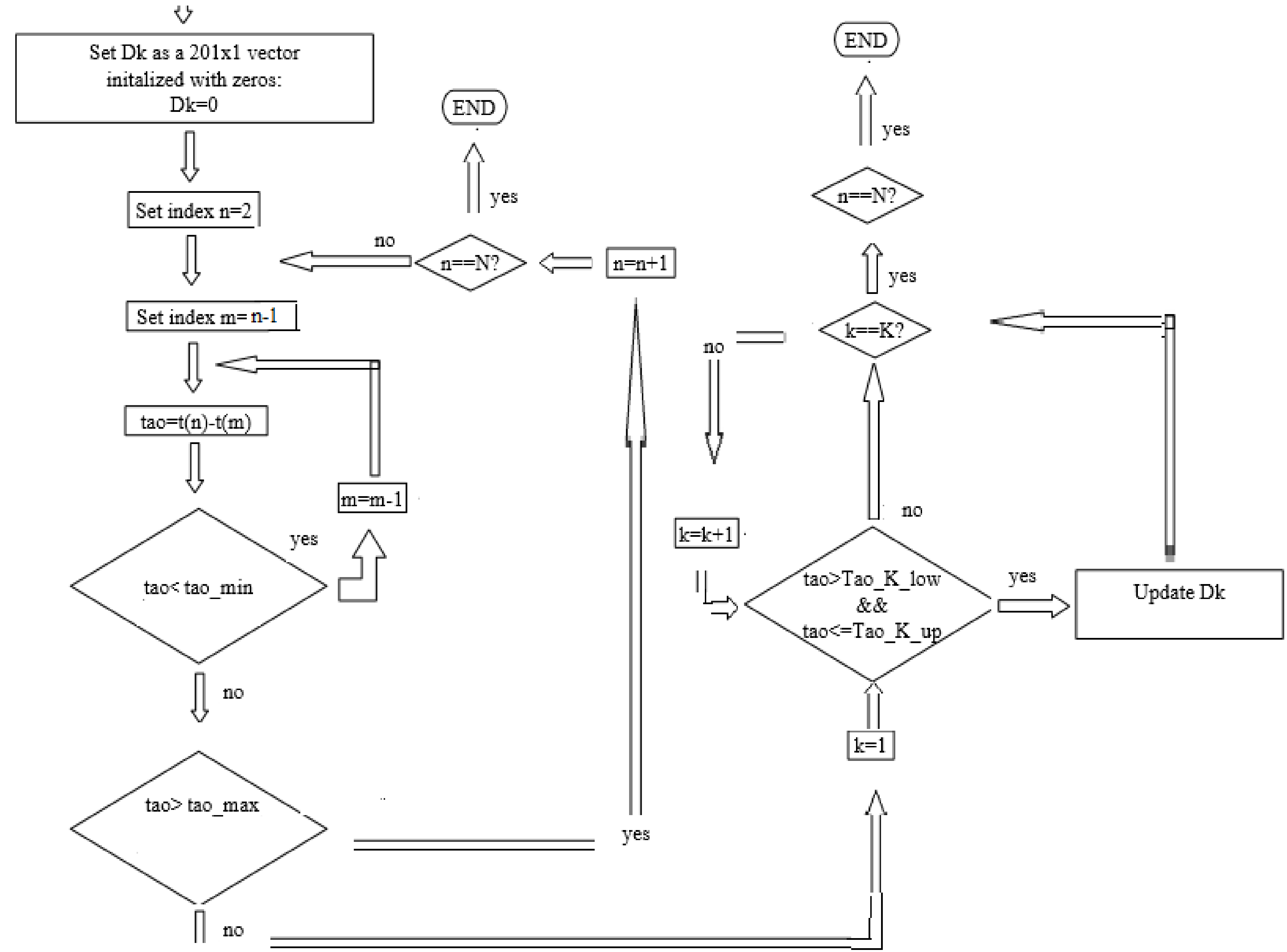
(a)



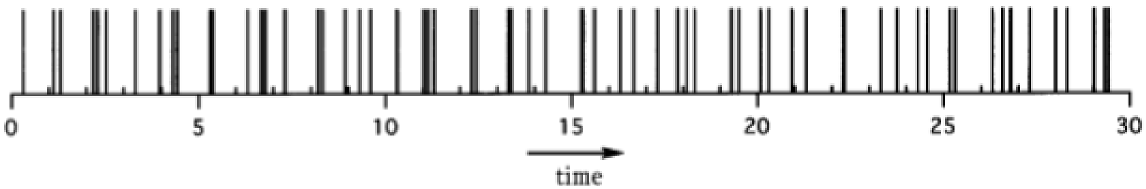
exp(iθ3)
exp(iθ2)
exp(iθ1)
0
exp(iθl)

Procedure to calculate PRI Transform (1/2)

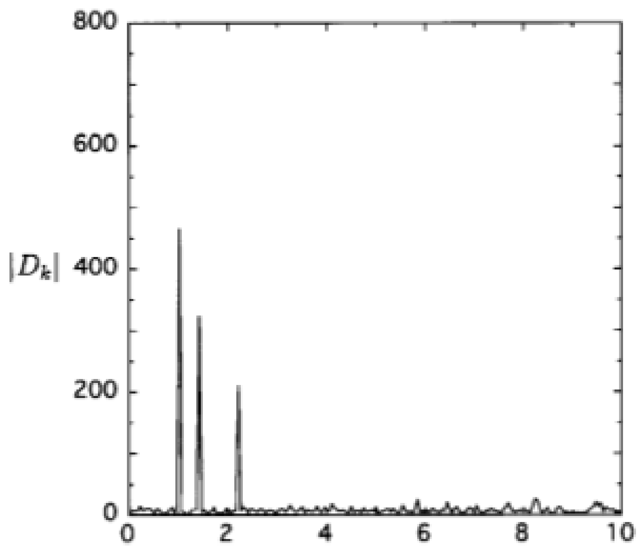



Set Dk as a 201x1 vector
initalized with zeros:
Dk=0
Set index n=2
Set index m= n-1
tao=t(n)-t(m)
tao< tao_min
yes
m=m-1
no
tao> tao_max
yes
no
END
yes
no
n==N?
n=n+1
END
yes
n==N?
yes
k==K?
no
no
k=k+1
tao>Tao_K_low
&&
tao<=Tao_K_up
yes
Update Dk
k=1

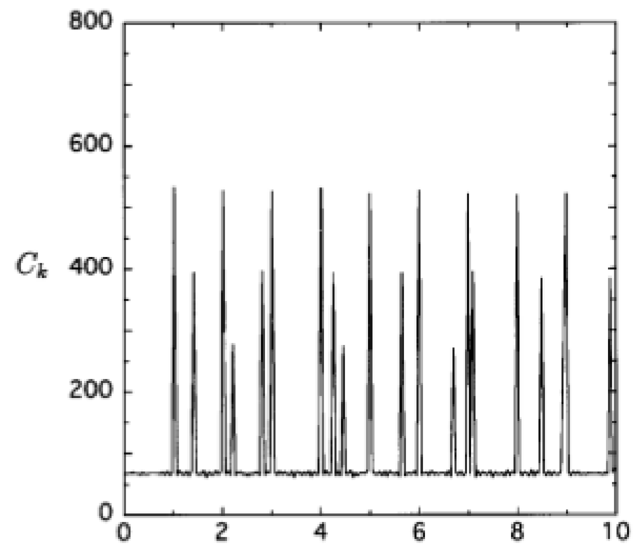
PRI Transform vs Autocorrelation



Input pulse train, which is superimposition of 3 single pulse trains with PRIs 1, √2, and √5



PRI spectrum



Autocorrelation function

Comments on PRI Transform

- It gives all the estimated PRI peaks in just one PRI spectrum
- the range of PRI (tao_min, tao_max) is an input
- In situations where the range cannot be predicted, it is advisable to first go for SDIF and get rough estimate of the upper limit of PRI (tao_max) before performing PRI Transform. **This has been followed in the this project**.

Jitter and Stagger PRI Estimation

PRI Agility and its significance

- Modern multifunction radar systems make use of multiple pulse repetition intervals values during one look at the target
- MTI radars operate by subtracting (in amplitude and phase) the echoes from one PRI from those in the next PRI.
- Changing the PRI changes the blind speeds and helps detect moving targets.
- Moreover, taking the PRIs close to each other can make blind speeds very high for the average PRI.
- Another reason for PRI variation is for protection against electronic attack. PRI jitter is used to prevent jamming of radar receiver.

Deinterleaving of agile PRIs requires specialized algorithms

- Modified PRI Transform
- SDIF with overlapped bins
- Stagger Analysis

Jitter PRI Estimation

Estimating Jitter PRI

Two different interval only TOA based algorithms studied as in the literature

1. SDIF and modifying the sequence search window in steps
2. Modified PRI Transform

Another simplified algorithm utilising a concept of Modified PRI Transform

Transform in SDIF : **Modified SDIF**

Modified PRI Transform

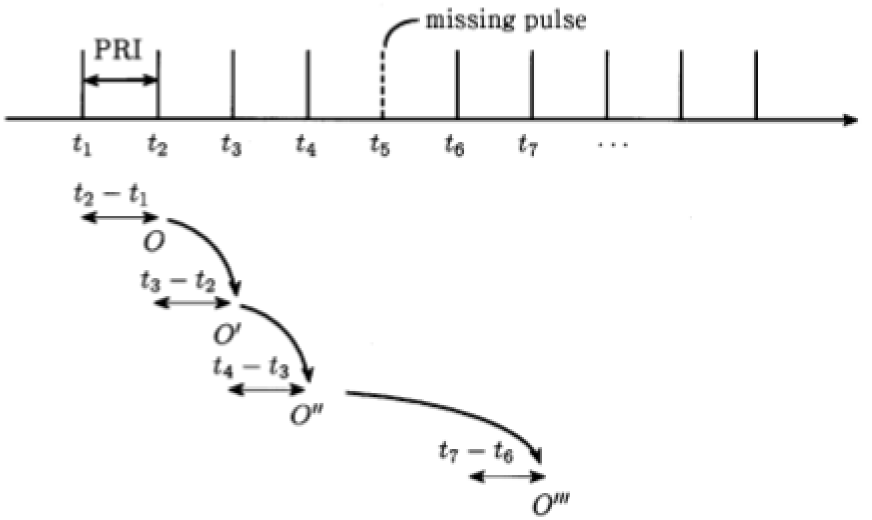
Modified PRI Transform

- PRI Transform does not work for jittered PRI

- Reason: peaks of the PRI spectrum derived from the original PRI transform are reduced
  – Factors that cause this reduction
  – The phase error of the phase factor of the PRI transform is enlarged
  – The pulse pairs, are distributed in several bins around the average PRI.

- Modifications
  – Shifting Time Origins
  – Overlapped PRI Bins

Shifting of Time Origin

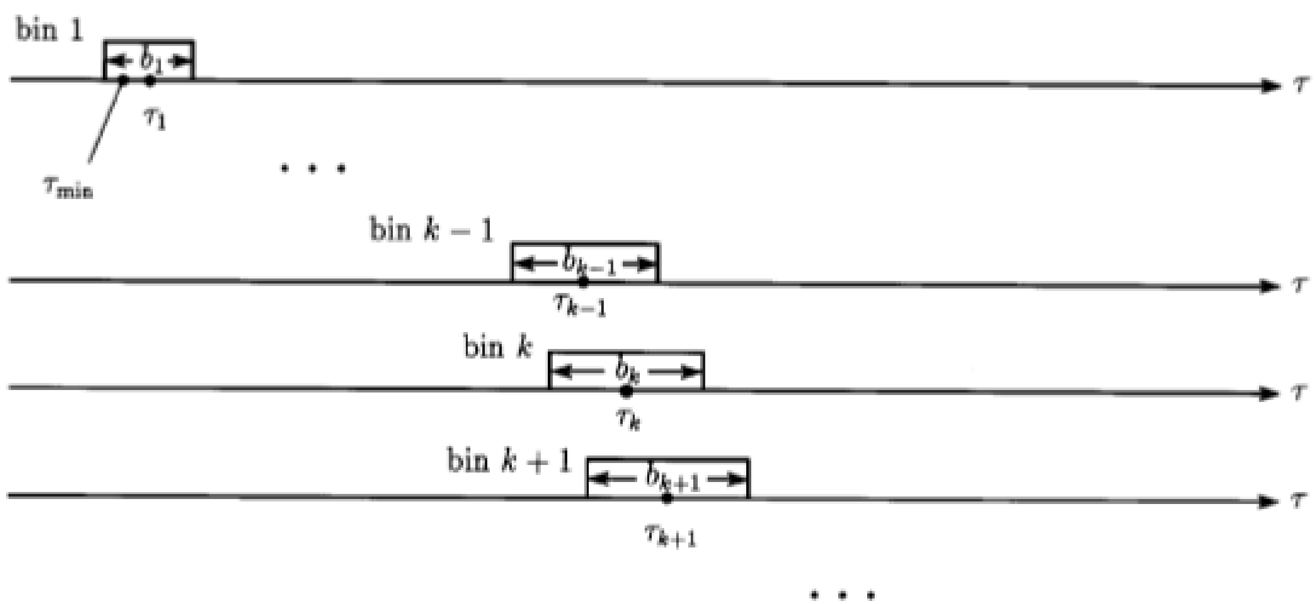


Overlapped Bins

To avoid the reduction of the peaks by the distribution of the pulse pairs, the width of the PRI bins must be greater than the width of the PRI jitter

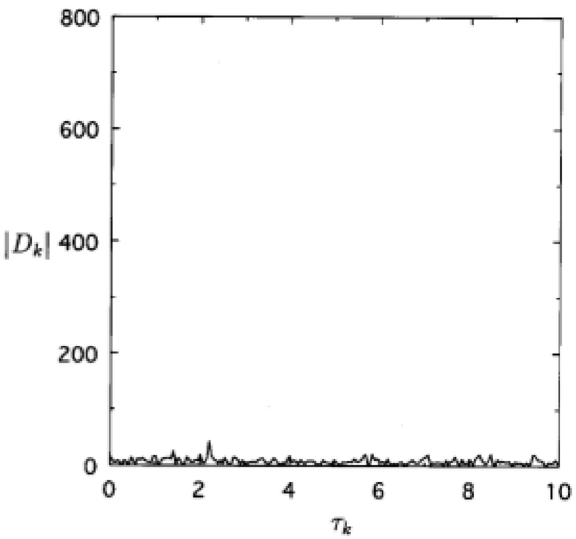
This causes the degradation of the resolution of the estimated PRIs

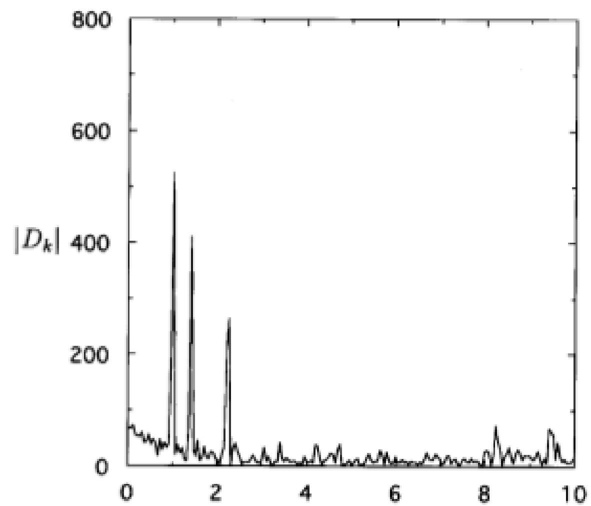
Solution: Overlapped Bins



PRI Transform vs Modified PRI Transform

Jitter =0.01





$$\Delta\tau = (\tau_{\max} - \tau_{\min})/K$$

$$b = (\tau_{\max} - \tau_{\min})/K$$

$$k_1 = \left[\left(\frac{\tau}{1+\epsilon} - \tau_{\min}\right) \Big/ \Delta\tau\right] + 1,$$

$$k_2 = \left[\left(\frac{\tau}{1-\epsilon} - \tau_{\min}\right) \Big/ \Delta\tau\right] + 1$$

Comments on Modified PRI Transform

- Subharmonic suppression in the modified transform is not as commendable as the original transform
- The algorithm needs knowledge of maximum jitter percentage

- **Threshold function** depends on three criteria

  - Criterion by Observation Time

  $$|D_k| \geq \alpha \frac{T}{\tau_k}$$

  - Criterion for Eliminating Subharmonics

  $$|D_k| \geq \beta C_k$$

  - Criterion for Eliminating Noise

  $$|D_k| \geq \gamma \sqrt{T \rho^2 b_k}$$

Modified SDIF

- Superior subharmonic suppression was the main reason to shift to PRI Transform from the less computational algorithm: SDIF
- Subharmonic suppression in the modified transform is not as commendable as the original transform
- Modified PRI Transform algorithm needs knowledge of maximum jitter percentage

Theory

- The main idea behind this modified SDIF is use of ‘**Overlapped Bins**’
- Rather than finding the number of pulses that belong to a particular bin, we find the number of pulses that belong to a range of PRI bins
- We make use of both TDOA histogram and cumulative TDOA histogram

Pseudocode (1/2)

1. Find the maxLevel of SDIF Histogram

2. For hist_level=1:maxLevel do the following processing

   dTOA(i)=TOA_array(i+hist_level)-TOA_array(i);

   [M, C]=hist(dTOA,length(TOA_array)); *%% M denotes the array of number of pulses mapped to corresponding array of PRI bins defined by C.*

3. Find the number of bins for each dTOA based on the jitter by using the equation

   number_of_bins=floor((tao/(bin_size)*jitter)

4. PRI bin limits can be computed as:

   relative_shift=jitter/2;

   count_high=floor((tao*(1+relative_shift))/bin_size)+1;

   count_low=floor((tao*(1-relative_shift))/bin_size)+1;

   number_of_bins=count_high-count_low;

Pseudocode (2/2)

5. form dTOA edges for each dTOA using the above lower limits. Let each such edge be defined as

[lower_limit:upper_limit] for each dTOA.

6. After dTOA edges are obtained for all dTOAs, cumulative histogram between the specified edges is performed wrt the dTOA.

7. Thresholding the cumulative histogram using similar threshold function as used for SDIF, gives the range of PRIs.

8. We can sort the TOA array based on the detected PRIs.

Discussion

- In order to avoid dependence on knowledge of jitter, we can assume different PRI jitter values in small steps and run this algorithm for all those jitter values.

  jitter_array=[0.1:0.01:0.32]

- The value of jitter which gives us smallest number of detected PRIs and/or smallest number of residual pulses after sorting the array, can be considered as the jitter ratio.

Modified SDIF vs Modified PRI Transform

- Thresholding function needs only one tuneable parameter.

- It can be argued that same process of iteratively varying jitter ratio and checking size of residual pulses can be used to determine jitter ratio in case of Modified PRI Transform as well. However, that will be a computationally very intensive algorithm.

- Moreover, SDIF does not need upper limit of PRI.

- determining PRI estimate from Modified SDIF and then using Modified PRI Transform hardly offers an advantage.

Stagger PRI

- Analysis for single stagger PRI identification
- Multiple stagger PRI deinterleaving

Stagger PRI deinterleaving

- The closeness of the individual sub periods of a pulse train makes each stagger pulse train seem like a subharmonic to PRI transform
- PRI Transform cannot deinterleave stagger PRI

Analysis for stagger PRI identification

- First level difference histogram is checked. Stagger PRI will result in nearly equal peaks
- perform difference histogramming at the level equal to the number of peaks
- This method can help deinterleave only one stagger PRI.

MULTIPLE STAGGER ANALYSIS

Method

1. Perform difference histogram at all levels on the interleaved time sequence up to a maximum level without any removal of sequence.

2. Save all the peaks obtained up to maximum level (even if multiple peaks are detected) and find corresponding bins.

3. Sort all possible sequences for each such bin and make a single sequence array by sorting all the sequences for a PRI.

4. Perform stagger PRI identification for sequence array for each PRI as mentioned in the previous technique.

5. If stagger is detected, compare the stagger PRI so obtained with the peak bin. If the values are close enough, we have detected the stagger PRI as well as the individual sub periods.

Discussion

- Performance of this algorithm depends on the performance of the pulse sorting algorithm.

- Sliding window sorting has been used.

PRI Sequence Sorting

- Sequence Search
- Sliding Window sort

Sequence Search with improvement

- Pairs of Time of Arrival are examined whose separation is nearly equal to the PRI.
- After obtaining such a pair, look into the neighbourhood of the pair to obtain better sequent pair.
- Use the pair giving the difference closest to the estimated PRI
- add this pair of ToA to the PRI sequence.
- Window
  - lower limit= 9.99*PRI
  - upper limit=1.01*PRI

Sliding Window sort (1/2)

Pseudocode for Sliding Window Sequence Search

1. for each PRI initialise i=1 and j=2 perform 2 to 10.
2. while j<=number of pulses, perform 3 to 10.
3. Calculate window size as window=1.5*PRI(emitter);
4. For each pair of pulses in the window, calculate the time difference
5. Find the deviation of each of these differences from the estimated PRI.

Sliding Window sort (2/2)

6. If the lesser one of the pair is not present perform steps 7 to 9. Else go to 10.
7. Check distance from the last value present in the sequent.
8. If this value is within a window of the PRI estimated or a near multiple, add it.
9. Else, analyse the last pair in the sequent and the pair which is being examined. If the pair which is being examined has difference closer to PRI, replace the last pair in the sequence with this pair
10. Add the higher value out of the pair.

Discussion

- It helps to form a very accurate sequence of a PRI, provided the value of PRI is estimated very correctly.

- Sliding Window Search is suitable for forming sequences of Constant and Stagger PRIs but not jitter PRI due to the above reason.

- It is computationally more intensive than the conventional sequence search. This is the price paid for the accuracy achieved.

- It helps to sort out the constant and stagger PRIs correctly even when jitter PRIs are also interleaved along.

Methodology

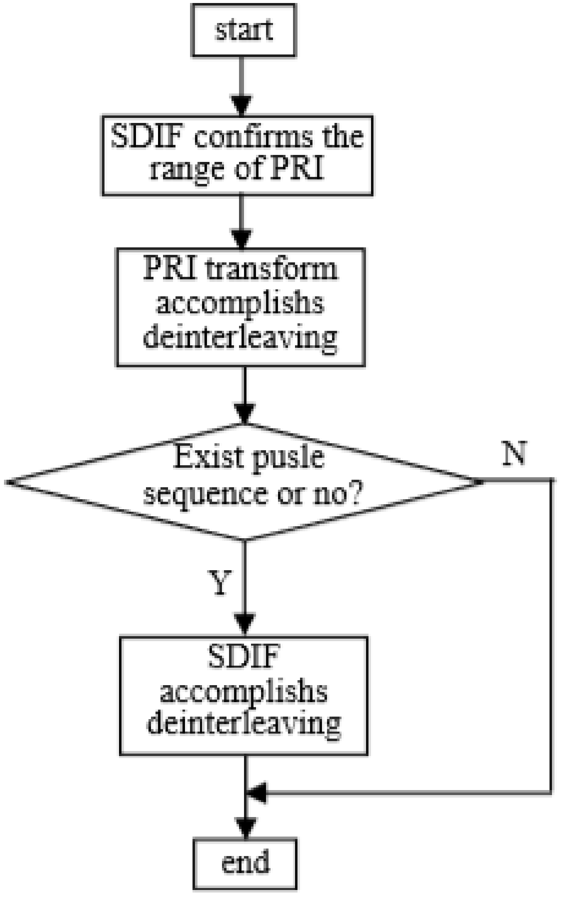
- Threat Generation
- System design as per the reference

Threat Generation

- A simple matlab code has been written to generate interleaved pulse trains.
- Inputs: PRI, jitter percentage, stagger periods (if stagger is to be included)
- Output: Interleaved array of time of arrival.
- To simulate the PPWs as captured at the sensors for a given emitter scenario in a practical scenario, the Lab View based Location Fix simulator provided by Digilogic Systems has been used.
- This system can generate constant and jitter PRI and has been used to reinforce the correctness of the algorithms.

System Model in [19]

- In order to overcome the drawbacks of PRI Transform/ Modified PRI Transform, but maintain its advantages, a system model based is described in [19]



Insights

This paper suggests us that

- mixing more than one deinterleaving technique can perform better than the individual one.

- Length of residual array indicates the completeness of deinterleaving.

This paper mentions about jitter of 8 percent.

In systems where higher jitter is present and multiple staggers are present too, jitter PRI deinterleaving gets impacted.

Therefore, it is advisable to extract stagger PRI pulses before jitter PRI deinterleaving.

Our System Model

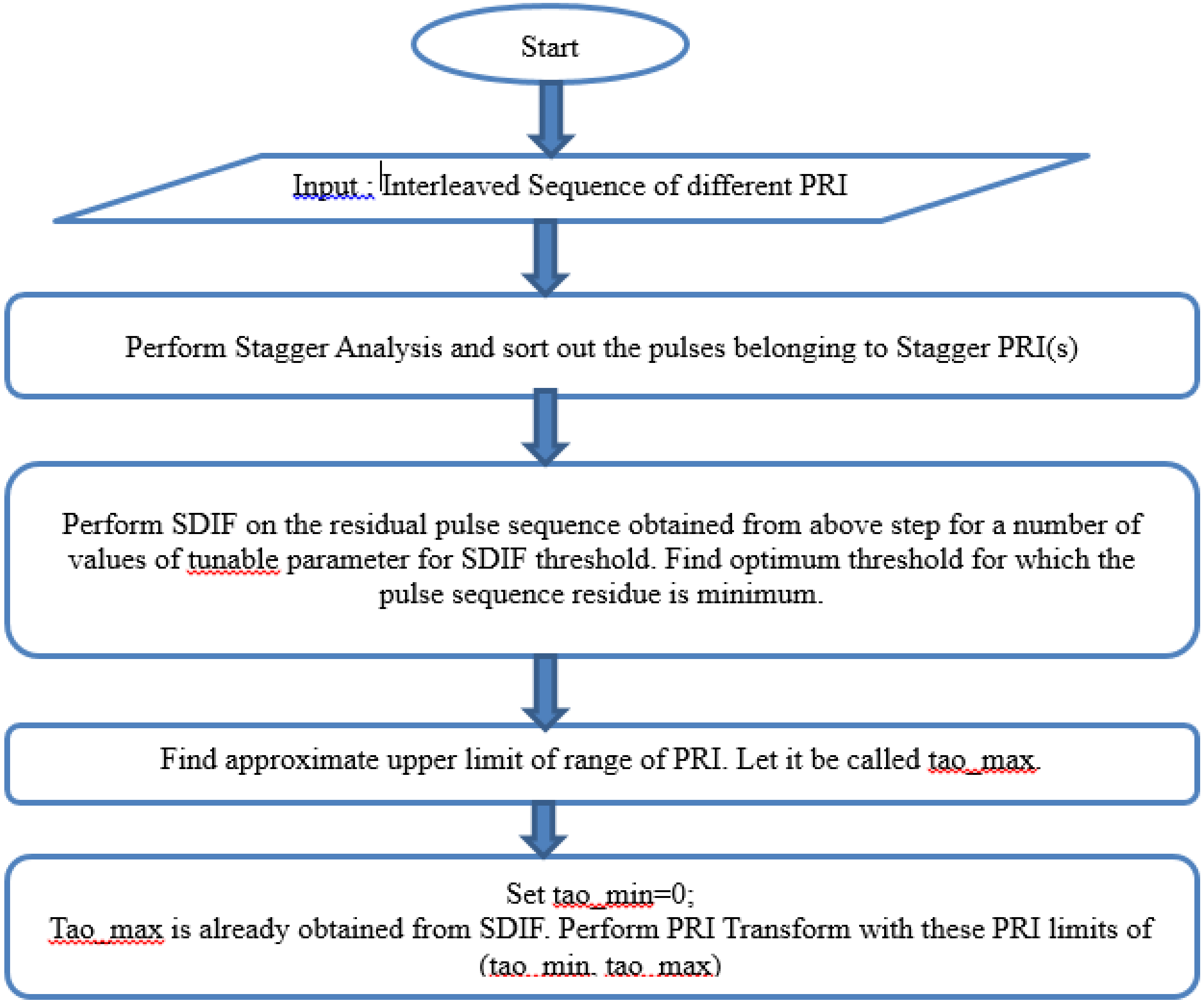
In this thesis, a system using a mix of similar deinterleaving algorithms has been implemented and its differences from the system mentioned in [19] is discussed below

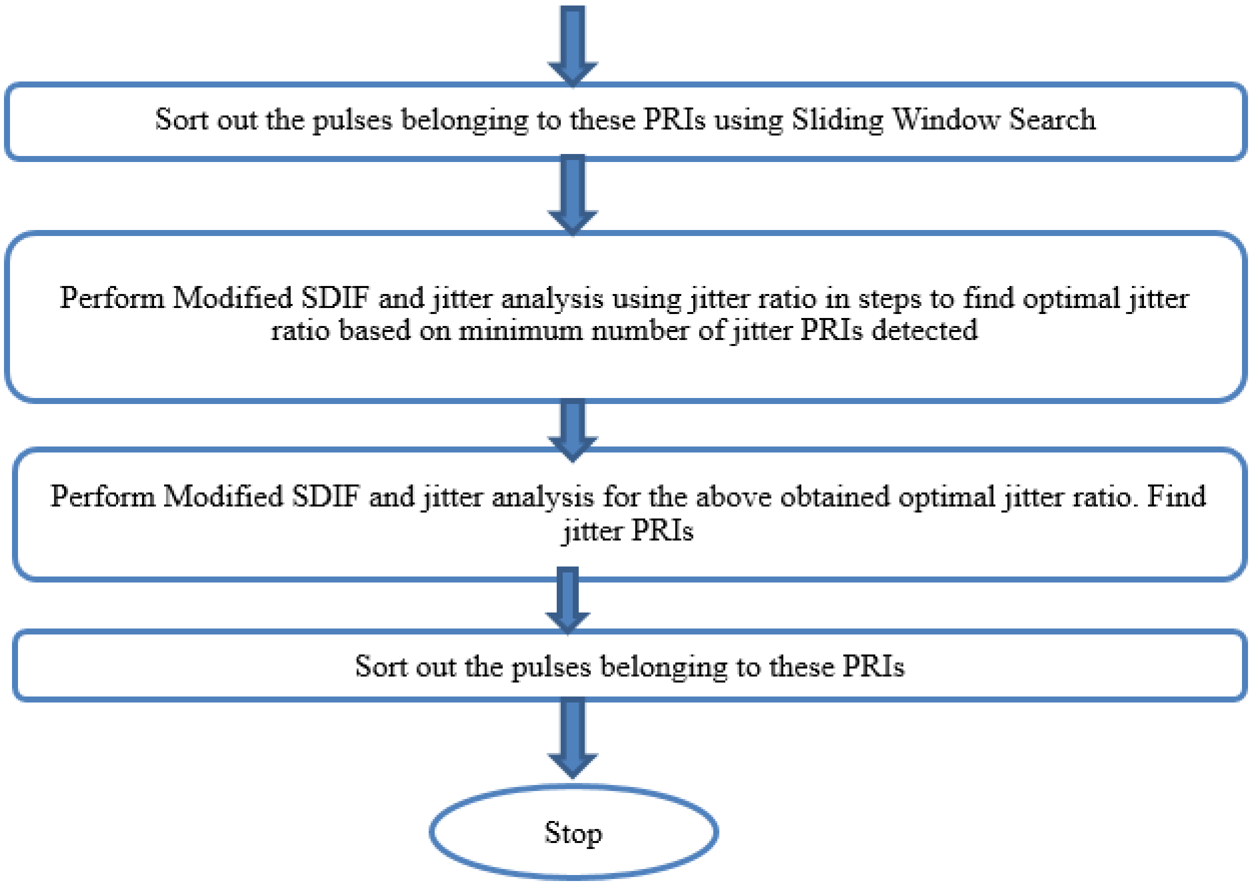
1. We need not implement complete SDIF at all levels for stagger deinterleaving. We need to check at two levels of histogram only.

2. Sometimes PRI transform peaks at 2-3 adjacent bins. In such a case, we will get extra PRIs. To avoid this, the peaks obtained from PRI transformed are clustered

3. We have used SDIF with overlapped bins to deinterleave jitter PRIs

Flowchart


Start
Input : Interleaved Sequence of different PRI
Perform Stagger Analysis and sort out the pulses belonging to Stagger PRI(s)
Perform SDIF on the residual pulse sequence obtained from above step for a number of values of tunable parameter for SDIF threshold. Find optimum threshold for which the pulse sequence residue is minimum.
Find approximate upper limit of range of PRI. Let it be called tao_max.
Set tao_min=0;
Tao_max is already obtained from SDIF. Perform PRI Transform with these PRI limits of (tao_min, tao_max)


Sort out the pulses belonging to these PRIs using Sliding Window Search
Perform Modified SDIF and jitter analysis using jitter ratio in steps to find optimal jitter ratio based on minimum number of jitter PRIs detected
Perform Modified SDIF and jitter analysis for the above obtained optimal jitter ratio. Find jitter PRIs
Sort out the pulses belonging to these PRIs
Stop

Simulation Results

- **PRI Transform**
- **SDIF**
- **SDIF + PRI**
- **Constant and jitter**
- **Constant and stagger**
- **System Model**

PRI Transform

ToA values as given in [1]

Inputs

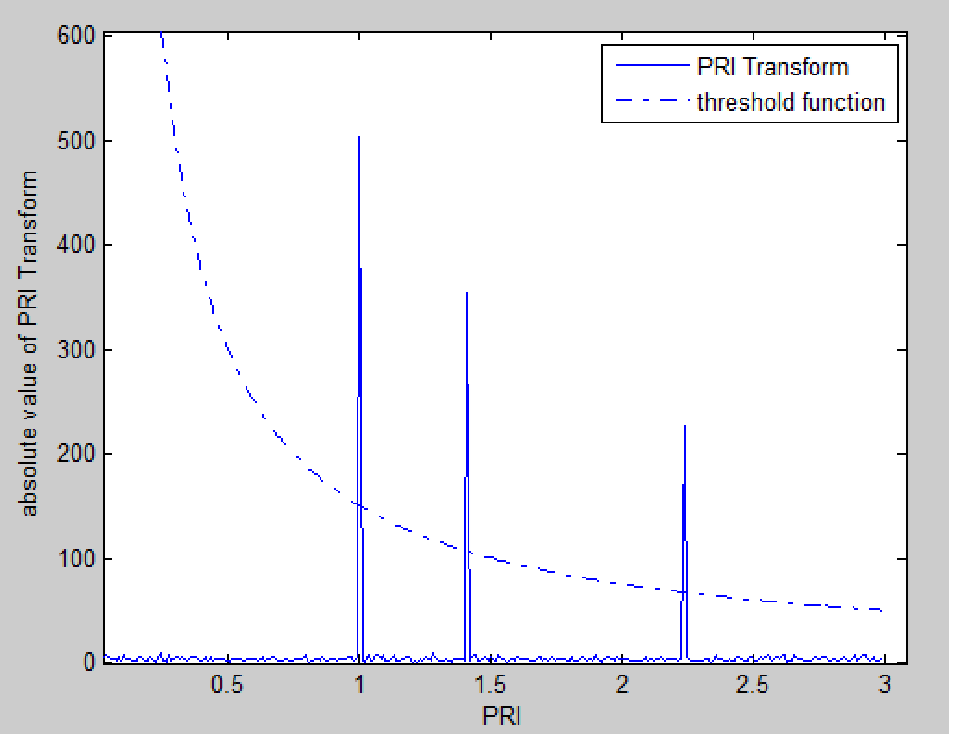
PRI values : 1, sqrt(2), sqrt(3)

Number of pulses:1080

Output

peakBins =

1.0003   1.4147   2.2357



Significance of the value of K

Two different values of K for above scenario

1. K= 2*length(TOA_array)+1 , for this case, K=2161

peakBins =

1.0003   1.4147   2.2357

2. K=201, as per reference [1]

peakBins =

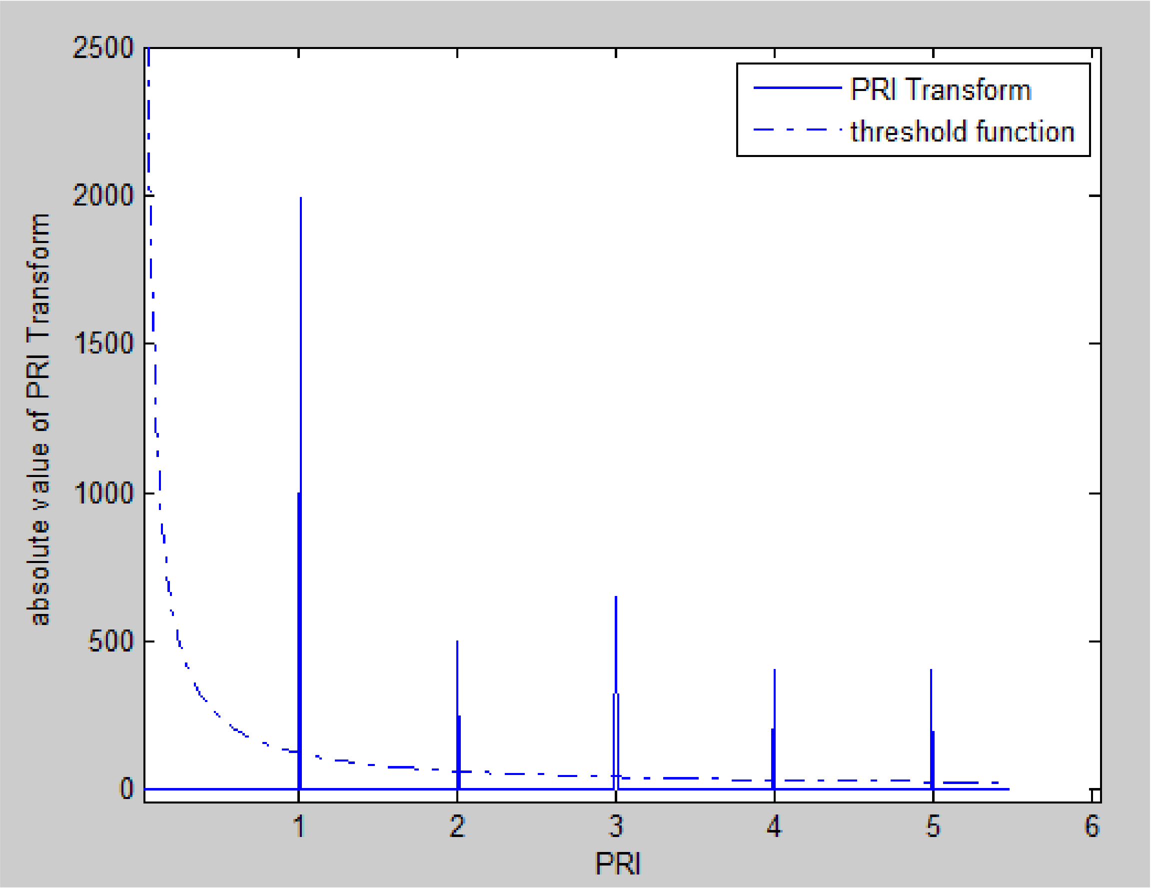
0.9928   1.4100   2.2443

Significance of tao_max

PRIs are 1,2,3,4, and 5

Number of pulses: 1016

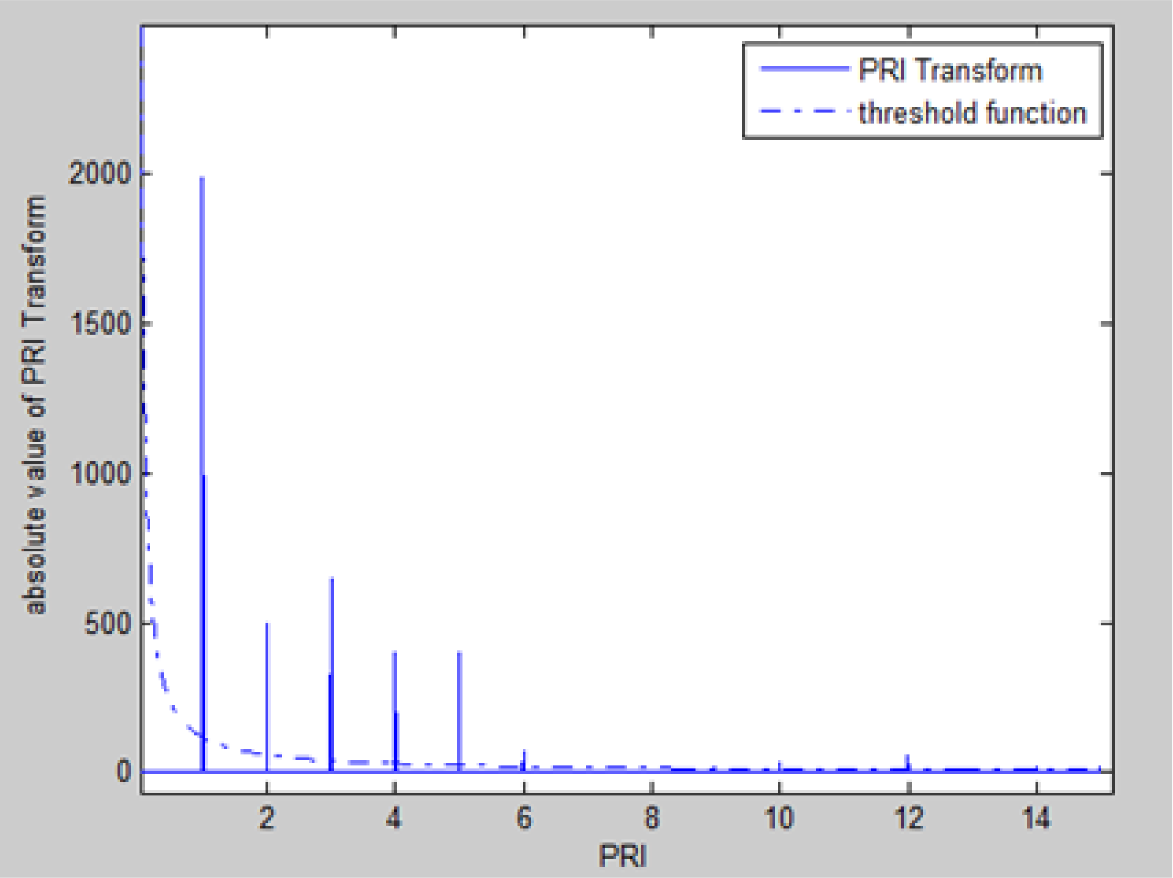
tao_min=0
tao_max=5.5



peakBins =

1.0032 2.0014 2.9995 3.9977 4.9959

tao_min=0
tao_max=15



peakBins =

1.0043 2.0037 3.0030 4.0023 5.0017 6.0010 9.9983 11.9970

Case of missing pulses

Missing pulses in the received TOA array, makes PRI Transform susceptible to subharmonic detection.

Case 1) 10 percent missing pulses

PRI estimation was accurate.

Case 2) 20 percent missing pulses

Actual PRIs: 1, 1.414, 2.236

Detected PRIs: 0.9996 1.4134 2.2349 6.9994

on sorting the TOA according to the detected PRIs using sequence search, 6.9994 formed very few sequents and got rejected

Reason for spurious subharmonic detection

- Phasor addition of different components of subharmonics of a PRI ideally results in cancellation of the effect of all the phasors belonging to a subharmonic of that PRI

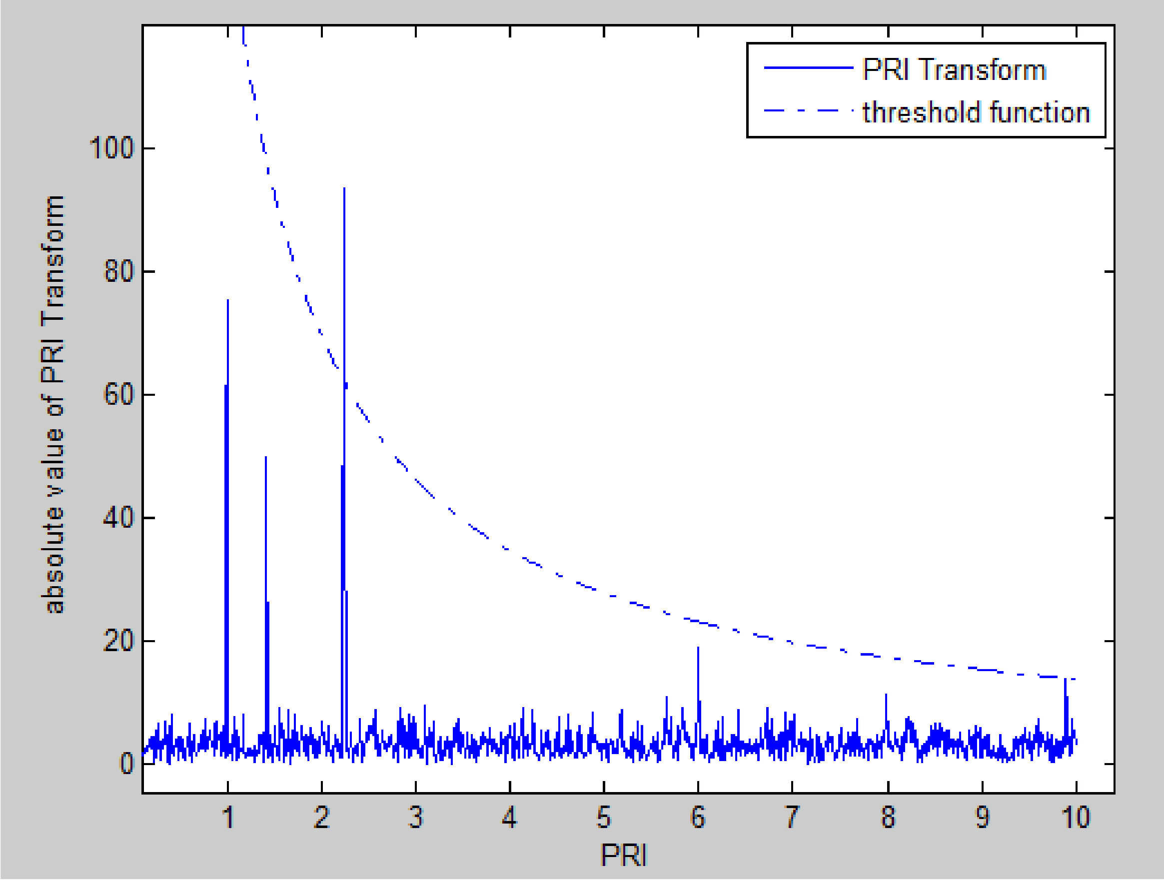
- missing pulses can cause some of the phasors to be absent in the summation expression of the PRI Transform

- As a result, the phasor addition result is cancelled partially.

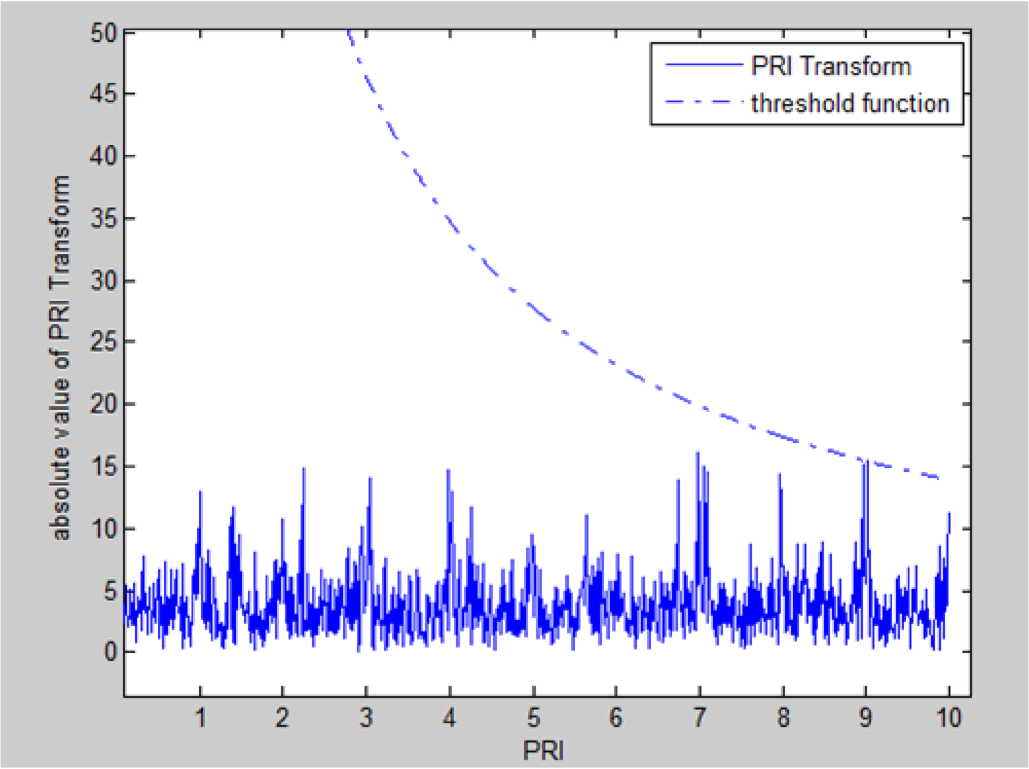
PRI Transform Results for Jitter PRI

Manual values as given in [1]

Jitter = 0.01



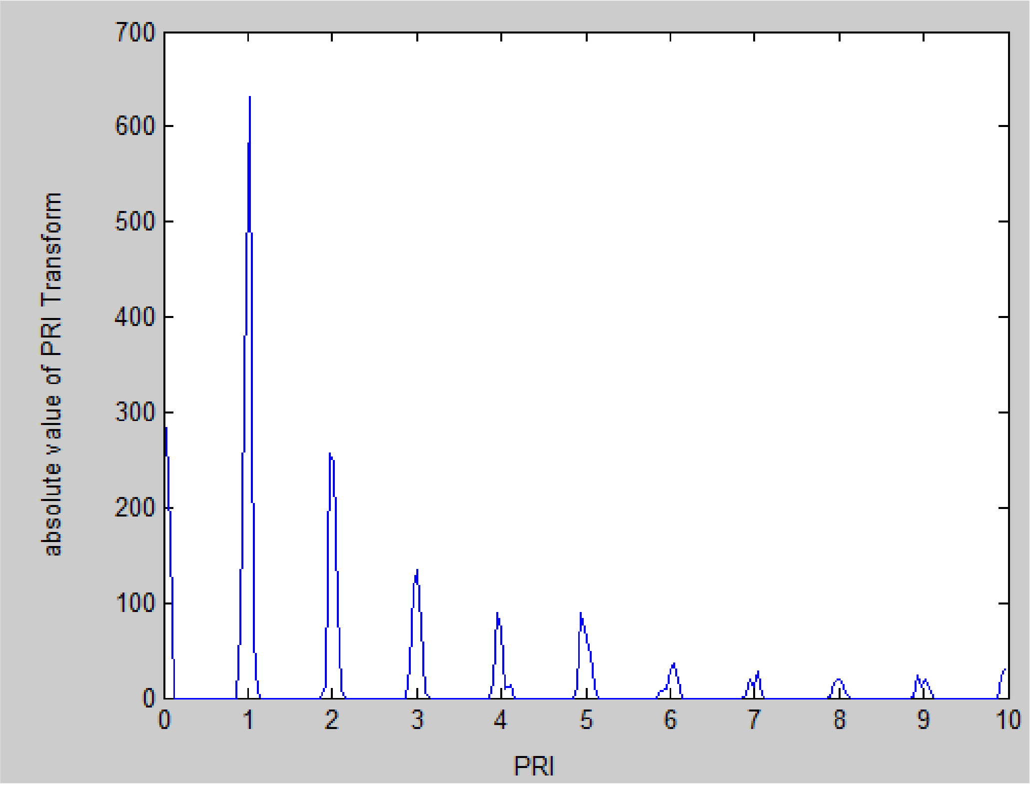
Jitter = 0.1



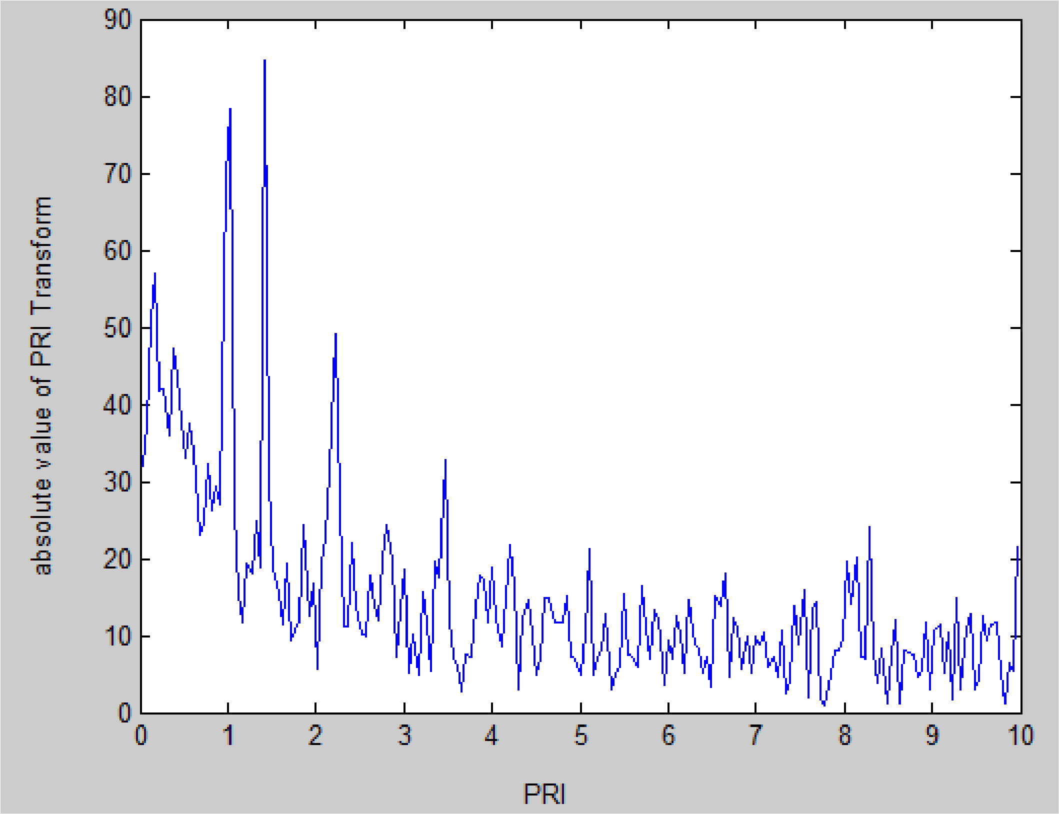
PRI Transform
threshold function
absolute value of PRI Transform
PRI
0
5
10
15
20
25
30
35
40
45
50
1
2
3
4
5
6
7
8
9
10

Modified PRI Transform

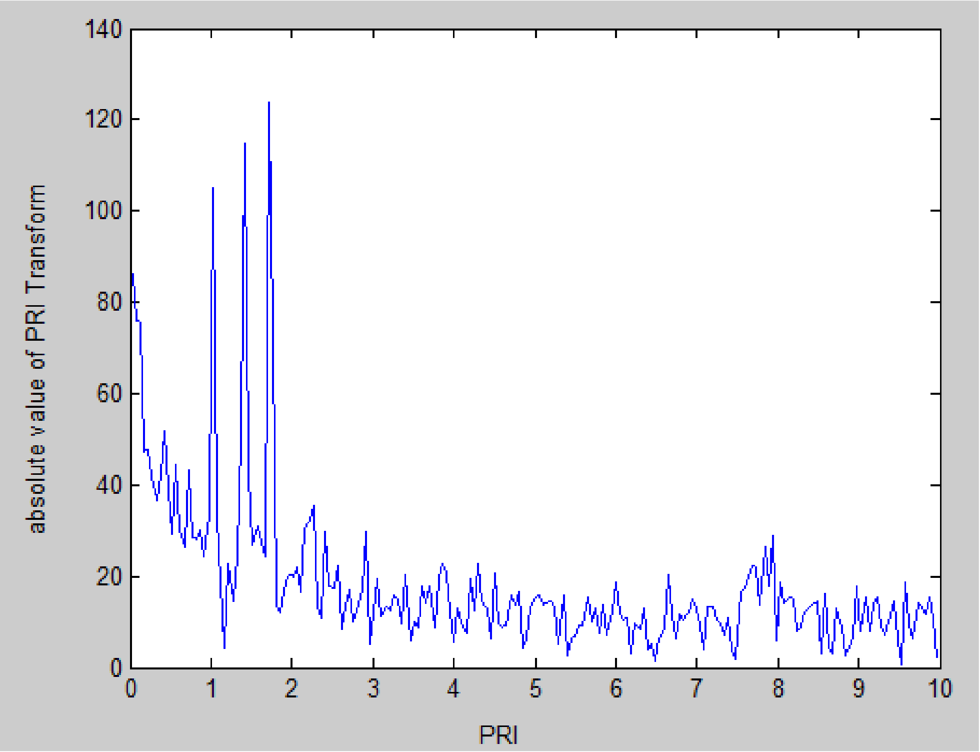
PRI values:1 2 3 4 5 ,
jitter = 0.1



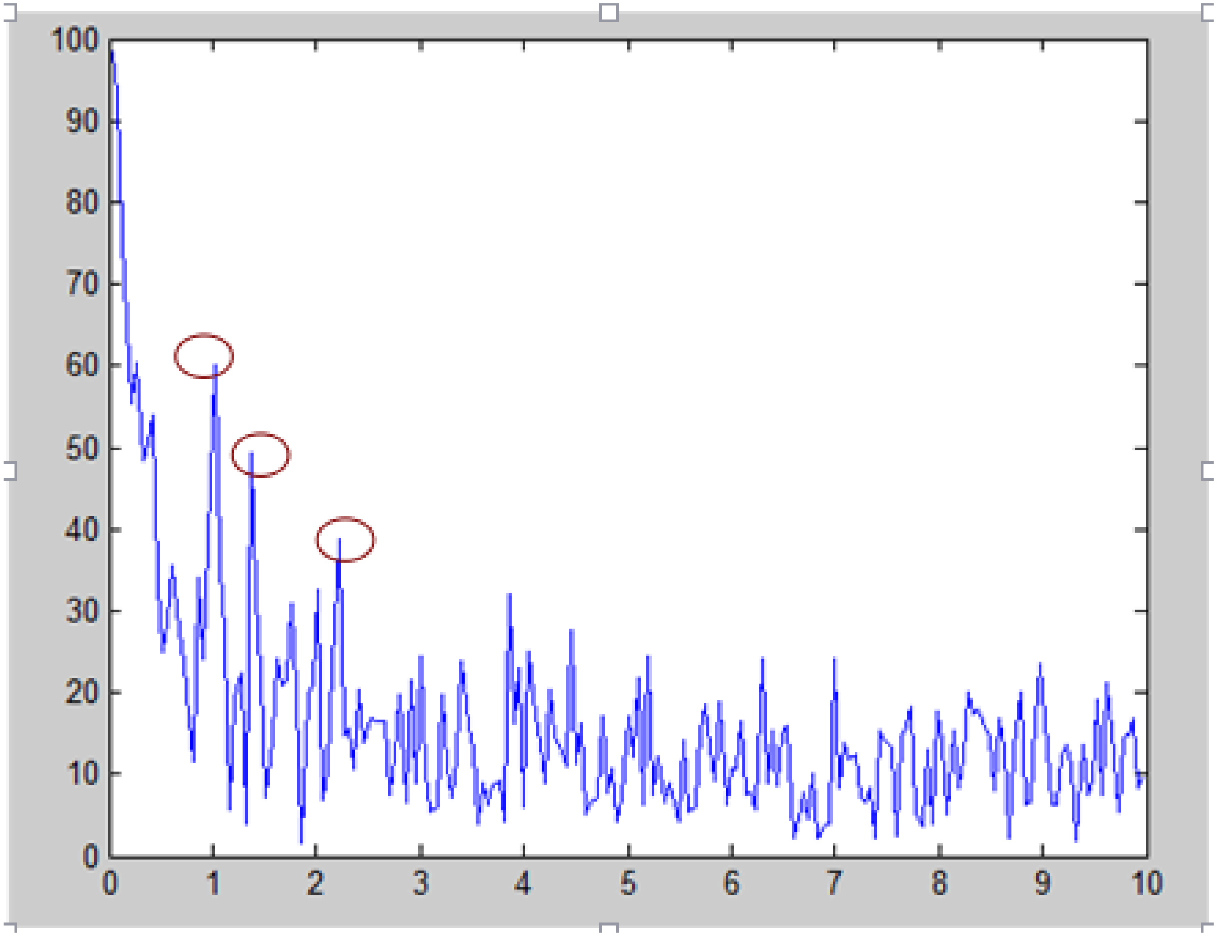
3 emitters
jitter=0.3



5 emitters
jitter=0.1



5 emitters
jitter=0.3



SDIF + PRI Transform

Practical data: 1000 pulses
PRF values: 2k and 7k
PRI values: 500 us, 142.86 us

SDIF Output:
PRI_array =[0.1429, 0.4998, 1.0327]

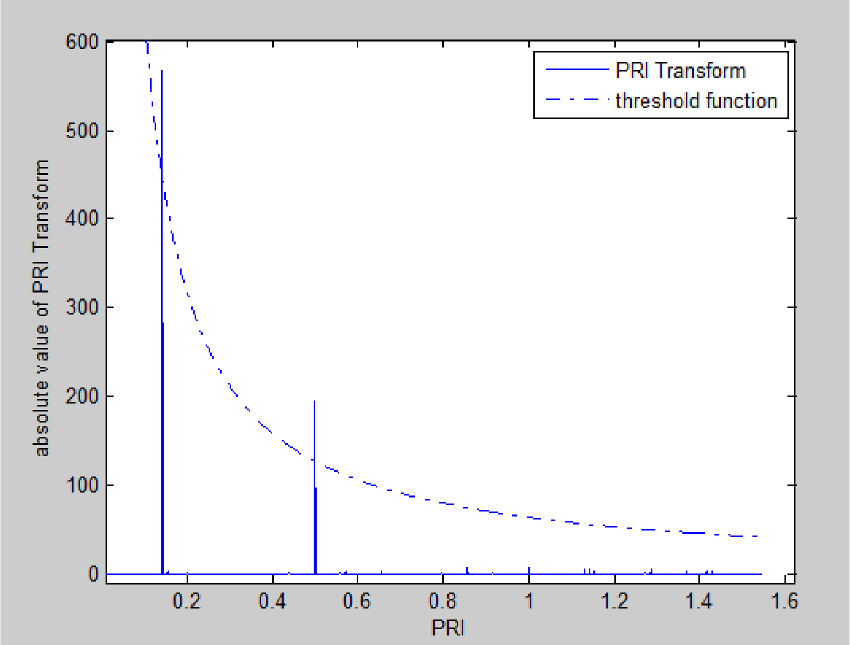
tao_max =

    1.5491

PRI Transform Outcome

peakBins =

    0.1428   0.4997

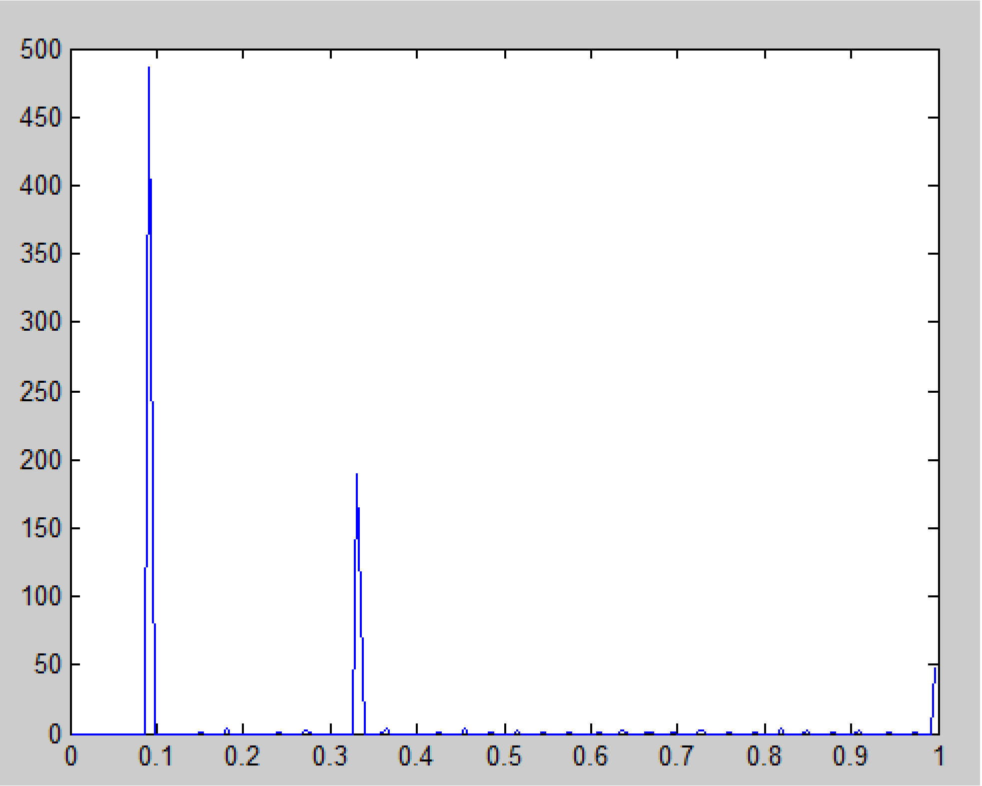


PRACTICAL VALUES
-Effect of POP

PRI Transform

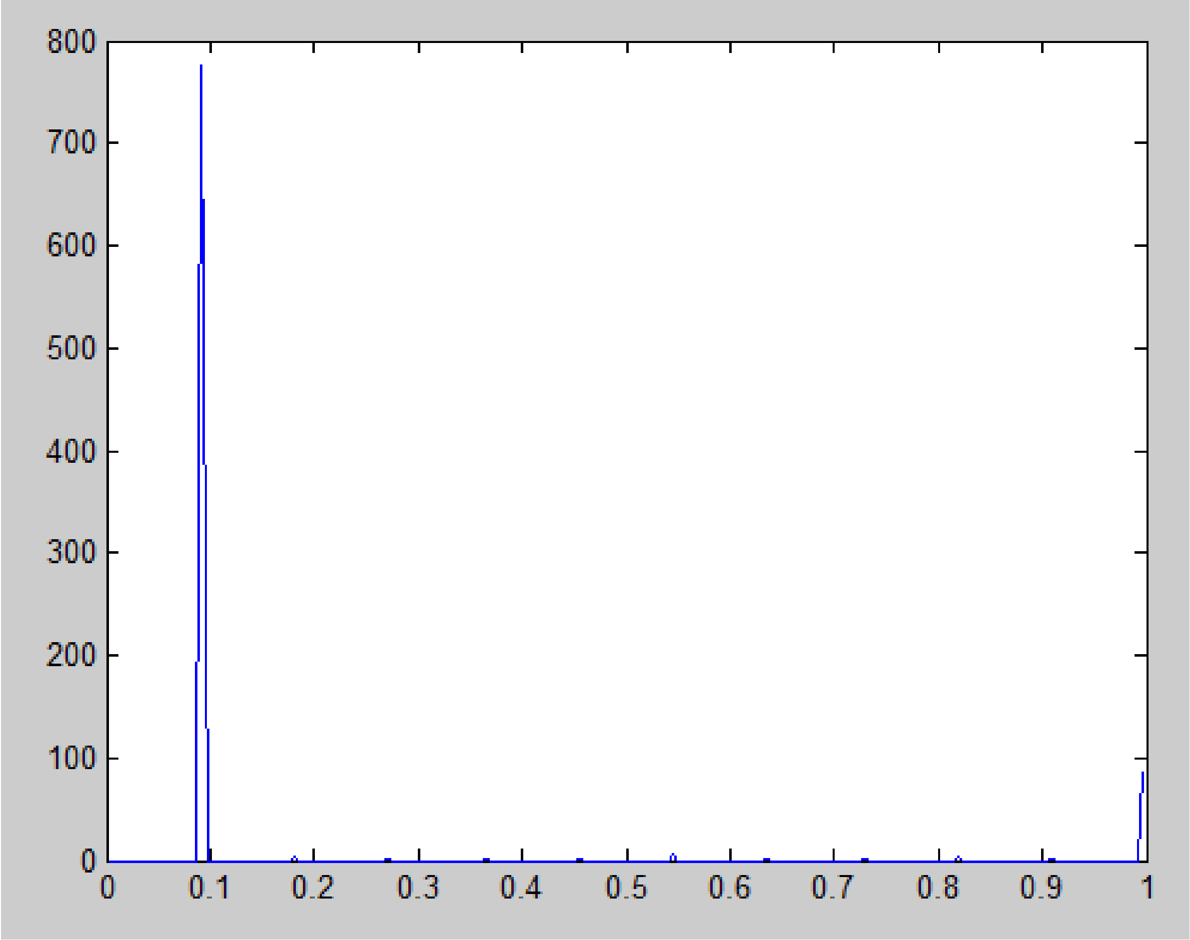
1

- PRI values 90.9 us and 333.33 us
- 1000 pulses
- no POP flags



2

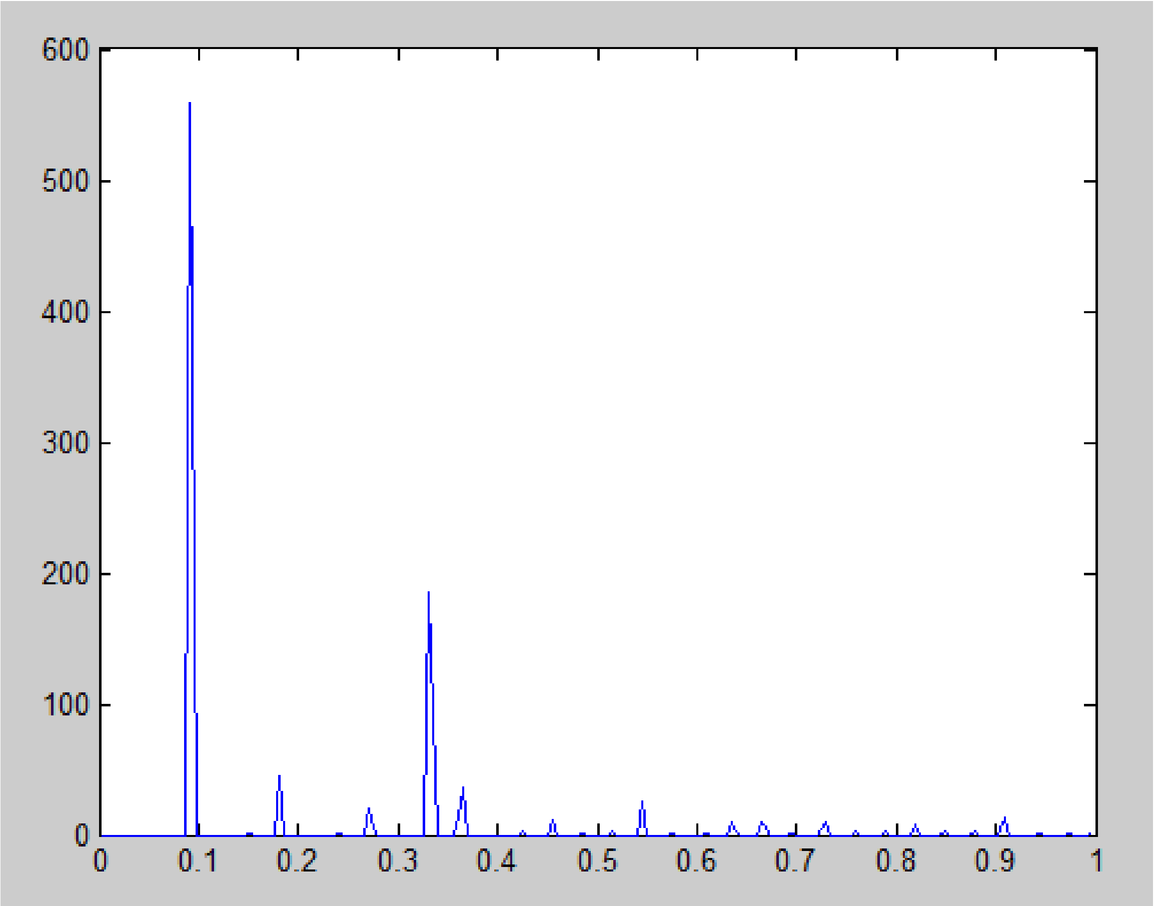
- PRI values 90.9 us and 333.33 us
- 1000 pulses
- many POP flags



Modified PRI Transform

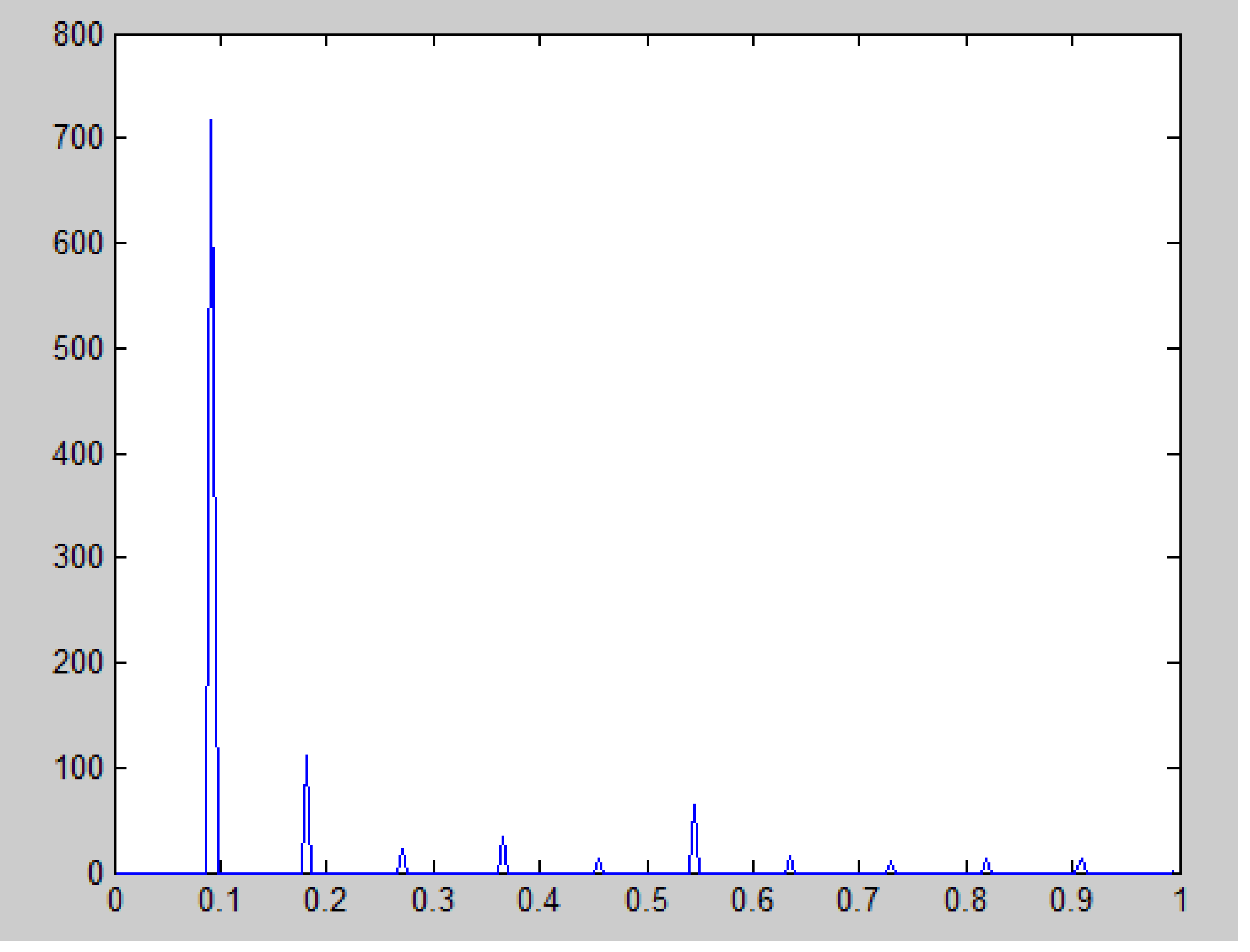
1

- PRI values 90.9 us and 333.33 us
- 1000 pulses
- No Jitter
- No POP



2

- PRI values 90.9 us and 333.33 us
- 1000 pulses
- No Jitter
- many POP flags are present



PRI Transform and Data Segregation

- PRIs from PRI Transform: 142.86 us and 500 us (PRF: 2k and 7k)

4	51514229	142.857
4	51514372	142.858
4	51514514	142.857
4	51514657	142.857
4	51514800	142.857
4	51514943	142.857
4	51515086	142.857
4	51515229	-786358
5	50728871	142.858
5	50729013	142.857
5	50729156	142.857
5	50729299	142.857
5	50729442	142.857
5	50729585	142.857
5	50729728	142.857
5	50729871	142.858
5	50730013	142.857

4	51512014	500
4	51512514	500
4	51513014	500
4	51513514	500
4	51514014	500
4	51514514	-783072
5	50731442	500
5	50731942	500
5	50732442	500
5	50732942	500
5	50733442	500
5	50733942	500
5	50734442	500
5	50734942	500
5	50735442	500
5	50735942	500
5	50736442	500
5	50736942	500

Jitter PRI Deinterleaving Using Modified SDIF

Data Generation Outcome

'mean PRI'	1	1.414	2.236
'jitter ratio'	0.2	0.2	0.2
'individual pulse trains'	<1x202 double>	<1x136 double>	<1x89 double>

Size of Time of Arrival (TOA) array generated: 427

II) Deinterleaving

- Step 1: Find optimum threshold
- >> optimum threshold multiplying factor= 0.3

- Step 2: Find optimum jitter value
- >> optimum jitter=0.24

- Step3: PRI estimation

- PRI Range

0.90269	1.087666157
1.29947	1.565074079
2.0658	2.485266168

Step 4: Pulse sorting

Before interleaving

0	1.062945	2.144103	3.0695	4.152176	5.178648	6.098156	7.053855	8.063232
10	11.37967	12.88179	14.30507	15.67685	17.215	18.73532	20.1635	21.61214
5	7.178942	9.27994	11.51133	13.67555	16.11352	18.53749	20.57345	22.91582

After deinterleaving

0	1.062945	2.144103	3.0695	4.152176	5.178648	6.098156	7.053855	8.06
10	11.37967	12.88179	14.30507	15.67685	17.215	18.53749	20.1635	21.6
5	7.178942	9.27994	11.51133	13.67555	16.11352	25.04857	27.25006	29.5

Size of residue:15

Data Generation

- TOA array length: 584

'mean PRI'	1	1.414	1.732	2.236	4.358
'jitter ratio'	0.3	0.3	0.3	0.3	0.3
'individual pulse trains'	<1x200 double>	<1x135 double>	<1x114 double>	<1x88 double>	<1x47 double>

Deinterleaving

Step 1: Find optimum threshold

>> optimum threshold multiplying factor= 0.3

Step 2: Find optimum jitter value

>> optimum jitter= 0.37

Step3: PRI estimation

1.222798303	1.654297
0.854552464	1.154897
1.908487171	2.577111
17.08836556	22.42248

The range

17.08836556 - 22.42248 is a false alarm.

Residue: 167

Data Generation

- TOA array length: 584

'mean PRI'	1	1.414	1.732	2.236	4.358
'jitter ratio'	0.2	0.2	0.2	0.2	0.2
'individual pulse trains'	<1x201 double>	<1x135 double>	<1x114 double>	<1x87 double>	<1x47 double>

II) Deinterleaving

- Step 1: Find optimum threshold
- >> optimum threshold= 0.3
- Step 2: Find optimum jitter value
- >> optimum jitter= 0.21
- Step3: PRI estimation
- Residue:37

0.893598	1.102199
1.239643	1.527048
1.557483	1.91844
2.06428	2.434001
4.019322	4.767102

Deinterleaving using Stagger Analysis - TDOA histogram (without sequence search)

Step 1: Identifying the PRIs

Level 1 peak bins

[79.9975,89.9975,99.9975]

3 peaks

Step 2: Check at Level 3

269.995370370370

Step 3: Confirm for stagger PRI

2 staggers with large separation in stagger period

I) Data Generation

Stagger periods: 300 and 21

[95;100;105]	[6,7,8]
<1x19 double>	<1x217 double>

II) Deinterleaving result

'Stagger Level'	3	3
'Stagger PRI'	20.96525	300
'Stagger Sub-period'	[6.0076, 6.9974, 7.9872]	[95.0013, 99.9986, 104.9986]
'Stagger Pulse trains'	<3x6 double>	<3x73 double>

3 staggers with close PRI

Stagger PRIs: 300, 270,
225

[95;100;105]	[80;90;100]	[70;75;80]
<1x103 double>	<1x115 double>	<1x136 double>

II) Deinterleaving

'Stagger Level'	3	3	3
'Stagger PRI'	224.9999	269.9999	300
'Stagger Sub-period'	[70.0001, 74.9998, 80.0000]	[80.0002, 89.9995, 99.9999]	[94.7002, 99.9997, 105.2997]
'Stagger Pulse trains'	<3x46 double>	<3x46 double>	<3x46 double>

Deinterleaving mix of different PRI types

I) Data Generation

Constant PRI generation

Constant PRI	11
Phase	0
Number of pulses	46

Jitter PRI generation

'mean PRI'	1.5	5.3
'jitter ratio'	0.2	0.2
'individual pulse trains'	<1x335 double>	<1x94 double>

Stagger PRI generation

Stagger PRI	22.5
Level	3
Sub Period	[7,7.50000000000000,8]
Pulses	<1x67 double>

Size of interleaved
array: 542

Scan Time: 0 to 500

II) Deinterleaving

- Stagger PRI detection and sorting

'Stagger Level'	3
'Stagger PRI'	22.49959
'Stagger Sub-period'	[6.9996 ,7.500, 7.9996]
'Stagger Pulse trains'	<3x21 double>

- Constant PRI deinterleaving

Constant PRI	10.99645
Number of pulses	43

Jitter PRI deinterleaving

1. PRI Range

Lower PRI Limit	Upper PRI Limit	Jitter ratio	Number of pulses
1.285409	1.650277	0.23	320
4.699379	6.028617	0.23	87

Size of Residue after removal of Constant, stagger and jitter PRI Pulses: 28

Comparing first few elements of generated and detected jitter PRI time of arrivals

Generated

0	1.649284855	3.152587	4.764641	6.135707395	7.781969	9.408783	10.92806	12.4075
10	15.59386637	20.90559	25.70501	31.5248178	36.82581	41.94773	46.90205	52.33522

Detected

1.649285	3.152587301	4.764640954	6.135707	7.781969	9.408783045	10.92806	12.4075	13.85885
15.59387	20.90559431	25.8159531	31.52482	36.82581	41.94773146	46.90205	52.33522	57.71487

I) Data Generation

Constant PRI generation

Constant PRI	11
Phase	0
Number of pulses	46

Jitter PRI generation

'mean PRI'	1.5	5.3
'jitter ratio'	0.3	0.3
'individual pulse trains'	<1x333 double>	<1x94 double>

Stagger PRI generation

Stagger PRI	22.5	28.5
Level	3	3
Sub Period	7,7.5,8	9,9.5,10
Phase	7.2	3.7

Size of interleaved array: 542

Scan Time: 0 to 500

II) Deinterleaving

- Stagger PRI detection and sorting

'Stagger Level'	3
'Stagger PRI'	22.49959
'Stagger Sub-period'	[6.9996 ,7.500, 7.9996]
'Stagger Pulse trains'	<3x21 double>

- Constant PRI deinterleaving

Constant PRI	10.99645
Number of pulses	43

Jitter PRI deinterleaving

1. PRI Range

Lower PRI Limit	Upper PRI Limit	Jitter ratio	Number of pulses
1.285409	1.650277	0.23	320
4.699379	6.028617	0.23	87

Size of Residue after removal of Constant, stagger and jitter PRI Pulses: 28

Comparing first few elements of generated and detected jitter PRI time of arrivals

Generated

0	1.649284855	3.152587	4.764641	6.135707395	7.781969	9.408783	10.92806	12.4075
10	15.59386637	20.90559	25.70501	31.5248178	36.82581	41.94773	46.90205	52.33522

Detected

1.649285	3.152587301	4.764640954	6.135707	7.781969	9.408783045	10.92806	12.4075	13.85885
15.59387	20.90559431	25.8159531	31.52482	36.82581	41.94773146	46.90205	52.33522	57.71487

I) Data Generation

Constant PRI generation

Constant PRI	11
Phase	5
Number of pulses	<1x46 double>

Jitter PRI generation

'mean PRI'	1.5	5.3
'jitter ratio'	0.3	0.3
'individual pulse trains'	<1x333 double>	<1x94 double>

Stagger PRI generation

Size of interleaved array: 590

Stagger PRI	22.5	28.5
Level	3	3
Sub Period	7,7.5,8	9,9.5,10
Phase	7.2	3.7
Number of pulses	<1x67 double>	

II) Deinterleaving

- Stagger detection and sorting

- Constant PRI deinterleaving

'Stagger Level'	3	3
'Stagger PRI'	22.5	28.5005
'Stagger period'	[7.00005133470226,7.49994866529774,7.99994866529774]	[9.00052186878728,9.50047216699801,10.0004224652087]
'Stagger Pulse trains'	<3x22 double>	<3x17 double>

Constant PRI	11.0035
Number of pulses	43

- Jitter PRI deinterleaving

Lower PRI Limit	Upper PRI Limit	Jitter ratio	Number of pulses
1.2948	1.6975	0.28	312
4.4988	5.9588	0.28	82

Comparing first few elements of generated and detected jitter PRI time of arrivals

Generated

0	1.446918412	3.078111	4.730742	6.311848	7.774463	9.338764	10.7101	12.26287	13.84171
10	**15.96707211**	**21.49285**	**26.16531**	**31.09669**	**36.8162**	**42.60196**	**47.27272**	**52.52464**	**57.37795**

Detected

0	1.446918412	3.078111	6.311848	7.774463	9.338764	10.7101	12.26287	13.84171	15.38717
4.730742	10	15.96707	21.49285	26.16531	31.09669	36.8162	47.27272	52.52464	57.37795

Size of Residue: 38

- When one of the PRIs is a multiple of another, only the smaller PRI is detected by PRI transform. This is due to subharmonic suppression.

- The inability of PRI Transform to detect stagger pulse trains is also due to subharmonic suppression.

- Length of the residual sequence after deinterleaving is a measure of performance of the deinterleaving algorithm. This has been exploited in tuning the threshold as well as finding the jitter present in an interleaved pulse sequence.

- The Sliding Window Sequence search gives a very accurate PRI sequence; however, it is computationally extensive. Moreover, it works well only if PRI estimate is quite exact. This makes it unsuitable for sorting jitter PRI.

- Deinterleaving of Jitter PRI using Modified PRI Transform or the Modified SDIF described in this report assume that all the jitter PRI have equal jitter ratios. This limits their capabilities to perform for systems where different PRIs have different dithering ratios.

- PRI at HCF of individual PRI is also detected by PRI Transform which is a false alarm.

- In cases of 30 percent jitter, 2-3 emitters can be detected using Modified PRI Transform. Similar is the case with Modified SDIF.

- Too many POPs affect the accuracy of both PRI Transform and Modified PRI Transform.

- Impact of missing pulses on PRI Transform:

PRI Transform performed well for 10 percent missing pulses; however, could not perform well beyond that.

When tested for 20 percent missing pulses, subharmonics were detected.

- In modern electronic warfare scenario, PRIs are becoming more and more complex. Deinterleaving such PRIs is a very challenging task.

- ToA interval based deinterleaving is sensitive to thresholding function. Practically, there is no thresholding function that works for every case. This paves the way for other deinterleaving methods like multi parameter deinterleaving, pattern matching based techniques, and event based techniques (based on ToA).