Implementation of Embedded Multi target Tracking System Algorithm for Active Phased Array Radar

M.C.Narasimhamurthy*(1)
LRDE, DRDO, Ministry of Defence
Government of India, Bengaluru, India

Dr.G.V.Jayaramaiah(2)
Dept. of Electronics & Communication Engg
Dr AIT, Mallathahalli, Bengaluru, India

***Abstract* - Multi Target Tracking Systems not only recognize the presence of the target, but it determines the targets location in range and in one or two angle coordinates. As it continues to observe the target the radar can provide target track and predict where target will be in the future. Prediction of the future position of the target will be useful for verity of purposes like guidance, navigation etc. An accurate estimation of target parameters and prediction of the target future location can be achieved by suitable processing of the received signal. Data processing is done by implementation of sophisticated algorithms. Kalman Filter algorithm and Joint probabilistic data association algorithm are implemented to track the target in multi-cluttered environment. Algorithms of this kind are designed and developed to track multi-targets with all possibilities thus yielding efficient and reliable system.**

Index Terms— Kalman Filter, Data Filtering, Data Association

I. INTRODUCTION

RADAR (Radio Detection And Ranging) is an electromagnetic system for location and detection of reflecting objects like aircraft, ships spacecraft, people, vehicles and the natural environment. It functions by emitting radiation into the open space and detecting the echo signals reflected from a target. The reflected echo energy that is returned to the Radar receiver not only indicates the presence of a target, but by comparing the received echo signal with the signal that was transmitted, its location can be determined along with other target-related information.

In Radar Data Processing (RDP) is a subsystem. It is a real time multi-tasking program consisting of a set of computer algorithms to utilize and process the radar measurements and automatically track the dynamics of one or more targets. Radar data processor (RDP) does Track While Scan (TWS) processing for multiple target tracking. Based on the data received via receiver/processor the objects next progression/position of the target is calculated.

A tracking system problem can be divided into two problems, namely Data Filtering and association. It is assumed that the Filter is receiving the signal distance in constant and discrete time periods. Approximation is performed by the Filter to obtain the target's actual position from these noisy measurements. In process to obtain the approximation, some prefixation of system is frequently presumed. Hence, the first problem relates to the selection of a specific Filter that can relate with most of the consequence in model where it is planned to be implemented.

The ethos of Data association is the mathematical calculation of association for all instance with entire computational measurements in the current scan bound, and the concurrent use of those probabilities as weighting coefficients in the formation of a weighted average computational measurements for updating each instance.

The computation is done by developing suitable algorithms for data association and filtering for multiple and diverse target tracking. It involves Software design. For software design Mat lab is used.

I. TRACKING SYSTEM

Tracking Radars

The tracking radar system estimates the coordinate distance of the object/target and provides reception data to the processor which may be used to estimate target path/location and to predict its future traversal path. Major feature which distinguishes it from other radar is method by which angle tracking is accomplished. Tracking radars may be classified into two types:-

a) Continuous tracking radar: This proposed system supplies continuous and concurrent tracking data on a targeted object.
b) Tracks while radar scan: This system provides computational data on multiple diverse targets.
In general, both systems deploy unique type of equipment's.
The antenna beam in continuous radar is positioned by an angle by servo-mechanism and it is actuated by an error signal. The various methods for generating error angle maybe differentiated as conical scan, sequential lobing and monopulse or simultaneous lobing.

When a tracking mode is enabled in the radar, it does not have knowledge of other potential targets. Each time a radar scans past a target, its coordinate estimates can be obtained for further computation. If the change in target coordinates between each scan is not too large, so it makes possible to reconstruct the trail of target from the received computed data. When the traffic is so dense the operator cannot keep track of information from radar, trajectory of the target may be automatically processed by using computational system. Such processing is called ADT (automatic detection and track).
An angular sector radar the provides target track is sometimes called track while scan radar. This terminology also applied to radars that scan a limited angular sector to provide tracking information at a high data rate on a particular target within its field of view. Missile control radars are of this type.

Introduction To Tracking and Data Association
Tracking is a process of computing measurements obtained from a target object in succession to sustain an evaluate from its current state, which typically consists system components such as radiated signal strength, spectral characteristics etc.

Constantly and significantly varying parameters such as coupling coefficients, propagation velocity.
Measurements are corrupted-noise records related to a target, such as
1. Straight estimation of target's position.
2. Field range and azimuth angle from a sensor.
3. Difference in time of arrival i.e., obtained by cross-correlating data between two sensors.
4. Distinct frequency from narrow-band signal reflected by the target.
5. Observed frequency difference which may be caused due to Doppler shift effect occuring two sensors in the system environmnet.
6. Also depends on the Signal strength.

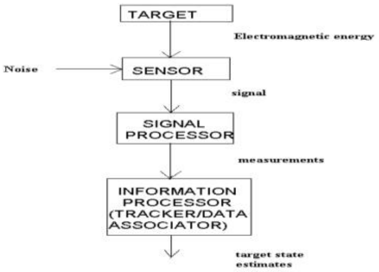

Figure: 1.1 components of typical tracking system

A trail path is a trajectory state computed from a set of measurements that will be related to a particular target. The crux of the multitarget-tracking problem states to carry out this data correlation process for coordinates whose origin is uncertain which may be caused due to:-

II.DESIGN OF MULTITARGET TRACKING SYSTEM

A simplified schematic of modern Tracking Radar is shown below. Antenna receive energy from a particular direction. Pedestal moves the antenna through electro-mechanical means to scan the beam across the search space in azimuth or elevation or both for tracking function. Radar Controller forms the nodal control element of any radar system.

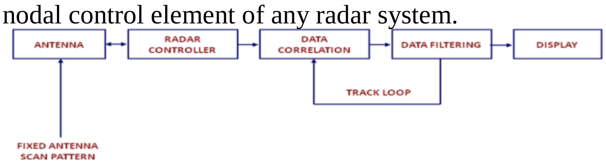

Figure: 1.2 Functions that comprise TWS System

A tracking system ken can be divided into two main propositions such as:-

1. Data Association.
2. Data Filtering.

It is presumed that the coordinate measurements are passed into the filter in a discrete, distinct and constant interval. Target's actual location/position can be estimated/approximated using a precise filter by utilizing the noisy measurements. Assorted system model is often assumed to obtain the coordinate approximates. As the target starts its maneuver action, the filter starts to diverge to the target's point of action. Due to the characteristics of the chosen model that describes the target trajectory, it provides impossible probability to expect that the corresponding filter will yield correct estimates when the target model maneuver's. Hence the first problem related to the selection of a precise Filter that relates with most of the issues in the application wherein it is intended to be implemented.

Data processed from JDPA(Data Association) and Kalman Filter are effectively used to track multi-targets efficiently. Radar Data Processor (RDP) uses the measurement sequence over successive scans to recursively estimate the target position and velocity.

a) General information of kalman filter

The Kalman Filter consist of two significant steps:
1. Prediction.
2. Updation or Correction.

In the initial phase, state is predicted with dynamic model and later in second phase, it is updated/corrected with observation model, this results in minimization of error covariance, which is an optimal estimator. This process is repeated each time in each phase with the state of previous time step considered as initial value for computation in next instance hence, Kalman filter is called recursive filter.

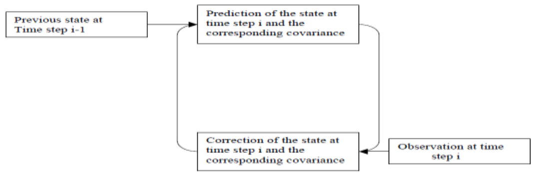

Figure: 1.3 Recursive process in Kalman Filter

The Kalman filter consists of the components such as State Vector, Dynamic Model and the Observation Model.

State Vector:

The state vector contains the variable of interest. This component describes the state of dynamic system and the degree of freedom is also represented via the State vector. The elements in the state vector cannot be measured directly, but they can be inferred from values that are measurable.
Elements of state vector can be orientation angles, position, velocity and so on. The state vector contains two values which is the a priori value- predicted value before update and a posteriori value- corrected value after update.

Dynamic Model:

This model enables to describe the transformation of the state vector over time. It is fairly being represented by system with distinct equations:

$$X(t1) = F_x(t1) + n(t1)$$

Wherein F relates to dynamic matrix and is constant, x (t1) belonging to state vector and n(t1) characterized by dynamic noise that is usually assumed as white noise thus containing co-variance matrix Q(t1).

OBSERVATION MODEL:

This observation model represents the relationship between the state and the coordinate measurements. The vector form of this system is represented as:-

$$L(t1) = H_x(t1) + w(t1)$$

Where L(t1) is an observation vector during discrete time steps t1, H belonging to observation matrix which being constant and w(t1) being the noise of the measurement with co-variance matrix R(t1).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 01, January-2018

PREDICTION:

As the name suggests, prediction becomes the initial operational phase of Kalman Filter. By shirking the dynamic noise and solving the differential equations describing the dynamic model thus enabling to calculate the predicted state, better than a priori state. The respective matrices equation considered for Kalman Filter is mentioned below:

X_pred = A* X_filt

P_pred = A * P_filt * A'+Q

CORRELATION:

The two main functions of RDP are correlation and TSE. Correlation is as important to RDP as filter process. The need for correlation process arises in a multitarget-tracking environment, i.e. when the radar sends multiple reports corresponding to closely spaced targets. In such a situation, it will be very difficult to unambiguously pair each of the reports to the track maintained by RDP, and one has to look for optimal assignment solutions with a matrix of m row and n columns. Whatever may be the method employed, the correlation, process should tell the filter process which report is paired with the track. Obviously, a wrong correlation may also contribute significantly to the build of large tracking errors.

UPDATION/CORRELATION:

During the Updation Phase, the predicted state vector matrix (X) is significantly enhanced upon the inspections made at each iterations of time step 't'.

K = P_pred* H' (H * P_pred * H'+R)-1

P_filt = (1-K * H) * P_pred

X_filt = X_pred + K (Z – H * X_pred)

In this phase, corrected value is predicted using the weight combination of estimated state and coordinate measurements, which means measurements weight will be high if the co-variance measurement value is much smaller than that of the predicted state and the predicted states will be low. Hence this helps in reduction of uncertainty in the model thus making it more reliable.

Kalman Filter Algorithm

In general, Kalman filter is used to remove noise interference from a signal, thus the measuring process must be delineated by linear system. By considering the discrete linear time system, the state space is represented as below thus providing a clear picture of the algorithm:

State Space equation:

$$x_{k+1} = Ax_k + Bu_k + w_k$$

Output equation:

$$y_k = Cx_k + z_k$$

From the above equations *A*, *B*, and *C* are coordinate measurement matrices,

k being the time index;

x provides the state of the system;

u delivers the input to the system;

y helps in measuring the output of the system;

w being the process noise and

z is acquired as the measurement noise.

Each of these above components are the vectors quantities thus contains more than one element in the nature. The component *x* contains all the data related to the current system state, to note *x* cannot be measured directly hence we measure component *y*, which is a function of *x* that is corrupted by the noise *z*. As a part of cognition, we can use *y* to help us obtain an estimate of *x*, but we cannot necessarily take the information from *y* at face value because it is corrupted by noise. In the current scenario, elementary laws of physics mention's that the velocity *v* will be delegated using the below equation:

$$v_{k+1} = v_k + Tu_k$$

Which means that the current velocity will be equal to the present velocity adding-up the commanded acceleration multiplied by *T*. But the previous equation does not give a precise value for *vk*+1. Instead, the velocity will be perturbed by noise. Noise velocity being a random variable may get changed along the time. So, the possible equation for *v* can be delegated as follows:

$$v_{k+1} = v_k + Tu_k + \tilde{v_k}$$

Below equation shows the derivation for position *p:*

$$p_{k+1} = p_k + Tv_k + \frac{1}{2}T^2u_k + \tilde{p_k}$$

Below equation *x* shows the state vector consisting of position and velocity:

$$x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix}$$

Finally, knowing that the measured output is equal to the position, we can write our linear system equations as follows:

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u_k + w_k$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + z_k$$

zk is noise measurement caused due to some particular cases such as instrumentation errors.

It needs to accurately estimate the state *x* thus accurately estimating the position of *p* and velocity *v*. suppose we have a linear system model as described previously. We want to use the available measurements *y* to estimate the state of the system *x*. We know how the system behaves according to the state equation, and we have measurements of the position, so how can we determine the best estimate of the state *x*. We want an estimator that gives an accurate estimate of the true state even though we cannot directly measure it. What criteria should our estimator satisfy would be a question, but two obvious requirements come to mind that being considered first, we want the average value of our state estimate to be equal to the average value of the true state. That is, we don't want our estimate to be biased one way or another. Mathematically, we would say that the expected value of the estimate should be equal to the expected value of the state. Second, we want a state estimate that varies from the true state as little as possible. That is, not only do we want the average of the state estimate to be equal to the average of the true state, but we also want an estimator that results in the smallest possible variation of the state estimate. Mathematically, we can state that we need to find the estimator with the smallest possible error variance. It so happens that the Kalman filter is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 01, January-2018

the estimator that satisfies these two criteria. But the Kalman filter solution does not apply unless we can satisfy certain assumptions about the noise that affects our system. Remember from our system model that *w* is the process noise and *z* is the measurement noise. We must assume that the average value of *w* is zero and the average value of *z* is zero. We must further assume that no correlation exists between *w* and *z*. That is, at any time *k*, *wk*, and *zk* are independent random variables. Then the noise covariance matrices *Sw* and *Sz* are defined as:

Noise Process Covariance:

$$S_w = E(w_k w_k^T)$$

Noise Measurement Covariance:

$$S_z = E(z_k z_k^T)$$

where *wT* and *zT* indicate the transpose of the *w* and *z* random noise vectors, and E(·) means the expected value.

There are many alternative but equivalent where *wT* and *zT* indicate the transpose of the *w* and *z* random noise vectors, and E(·) means the expected value. There are many alternative but equivalent ways to express the equations. One of the formulations is given as follows:

$$K_k = AP_kC^T(CP_kC^T + S_z)^{-1}$$
$$\hat{x}_{k+1} = (A\hat{x}_k + Bu_k) + K_k(y_{k+1} - C\hat{x}_k)$$
$$P_{k+1} = AP_kA^T + S_w - AP_kC^TS_z^{-1}CP_kA^T$$

That's the Kalman filter. It consists of three equations, each involving matrix manipulation. The *K* matrix is called the Kalman gain, and the *P* matrix is called the estimation error covariance. The state estimate ($\hat{x}$) equation is fairly intuitive. The first term used to derive the state estimate at time *k* + 1 is just *A* times the state estimate at time *k*, plus *B* times the known input at time *k*. This would be the state estimate if we didn't have a measurement. In other words, the state estimate would propagate in time just like the state vector in the system model. The second term in the equation is called the correction term and it represents the amount by which to correct the propagated state estimate due to our measurement. Inspection of the *K* equation shows that if the measurement noise is large, *Sz* will be large, so *K* will be small and we won't give much credibility to the measurement *y* when computing the next. On the other hand, if the measurement noise is small, *Sz* will be small, so *K* will be large, and we will give a lot of credibility to the measurement when computing the next x.

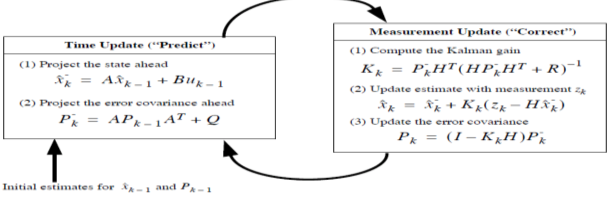

Figure: 1.4 Overview of Kalman filter

Desktop Environment:- The MATLAB desktop is a graphical user interface that transforms MATLAB into an integrated environment for exploration and development [14].

Matrices:- MATLAB functions with scalars, vectors, and matrices quantities. A scalar is just a one-by-one matrix, and a vector component is nothing, but a thin, long matrix represented as either a row or a column. To summarize, MATLAB basically works on matrix's

Numeric's:- This section contains some demos of numeric calculation in MATLAB. This numeric involves demos illustrating, expounding differential equations, curve fitting and Fast Fourier transform (FFT).

Graphics:-These demos illustrate graphics functions that create 2-D and 3-D plots of data, and introduce visualization techniques.

MATLAB provides a high-level language and development tools that let you quickly develop and analyze your algorithms and applications.

Development Tools:- Development tools help users implement their algorithm efficiently. This tool kit includes the following components:

MATLAB Editor:- Provides standard editing and debugging features, such as setting break points and single stepping.

M-Lint Code Checker:- Enables the users to Analyzes the code and suggests modifications to enhance its performance and reliability.

MATLAB Profiler:- Provides the overall picture of the total time spent while the system executing each line of code from the block.

Directory Reports:- Generates reports regarding the efficiency of code, file disintegration, code debug, file dependency chain and so on.

Toolboxes

These are specialized and dedicated MATLAB program file container built particularly for solving specific block of problems. Toolboxes are associated with fields such as controls, signal processing system identification and so on[14].

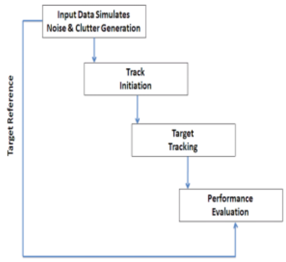

Figure 1.5: Matlab flow chart radar data processing tool

MATLAB, version-7.7 is used as a tool to simulate the Multi target tracking algorithms as it has all the above features which are user friendly.

III.GUI Model for multitarget tracking :

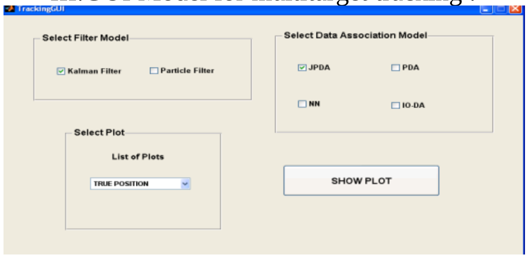

Figure: 1.6 Graphical User Interface diagram of Multi target tracking system

GUI is visually presented as a front end of the software application. MATLAB toolbox contains functions for state estimation for linear systems. The toolbox will run under MATLAB version 7.7 and higher, and it is independent of other MATLAB toolboxes. All functions exist as m-files the purpose of this toolbox is to make implementations of the Kalman filter and JPDA algorithms for multi target tracking. Two frames are used to group logically related objects. One to group filter algorithms and the other is to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 01, January-2018

group data association algorithms. Pop up menu contains the list of tracking plot to be viewed. The desired plot is viewed clicking the show plot button (push button) after selecting the plot title in Popup menu.

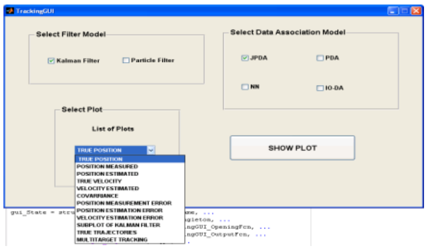

Figure: 1.7 Screen snapshot showing the list of plots

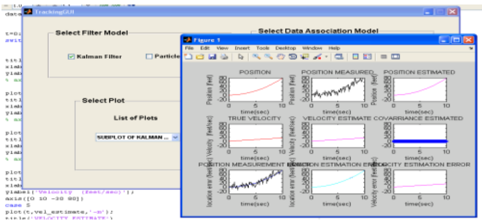

Figure: 1.8 Screen snapshot showing the subplot

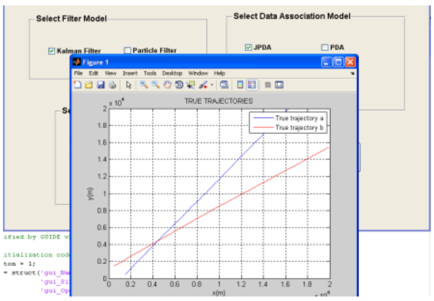

Figure: 1.9 Screen snapshot showing the true trajectories plot

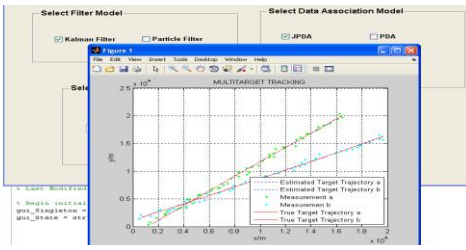

Figure: 2.0 Screen snapshot showing the estimated trajectories plot.

IV. SIMULATION AND RESULTS

The Tracking system was simulated using the software Matlab (7.7version) and the following results were obtained. Various scenarios were generated to depict the performance of the Tracking system. Pop up menu in GUI contains the list of tracking plot to be viewed. The desired plot is viewed clicking the show plot button (push button) after selecting the plot title in Popup menu. Below are the visual appearance of unique plots from the tracking system:

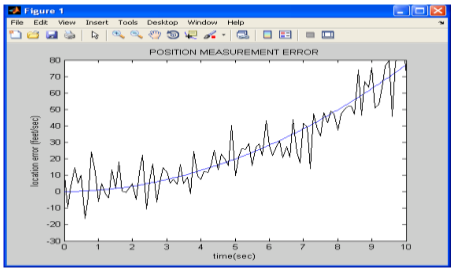

Figure: 2.0 Position Measured Error

The above figure clearly shows different plots like True position of the target, Position measured, Position estimated, True velocity of the target, Velocity estimated, Covariance estimated, Position measured error, Position estimation error and Velocity Estimation error. Different Plot are combined into single subplot for easy analysis of the plots

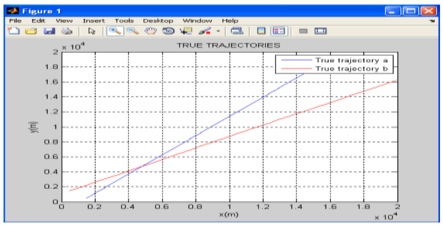

Figure: 2.1 Target Trajectories.

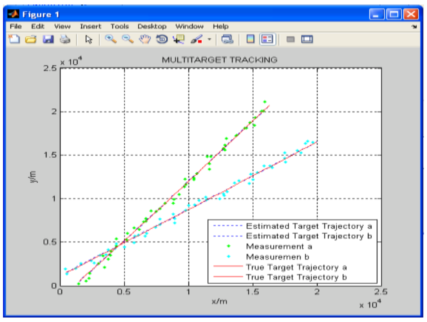

Figure: 2.2 Target and its trail trajectories path

Using the perception of Joint probabilistic data association algorithm and Kalman filter algorithm, enabled the design and development of Multi-Target tracking system therefore simulated with the use of MATLAB version 7.7. The plot shows literal Trajectories, Measurement of the target objects and Estimated Target Trajectories.

V.CONCLUSION

Implementation Embedded Multi target Tracking System Algorithm for Active Phased Array Radar involves design and development of Kalman filter Algorithms. The design is for Radar data processing. The subsystem Radar data processing forms the nodal element in radar system, does Track While Scan (TWS) processing for multiple targets and target tracking. Multi target tracking system design GUI based using Kalman filter algorithm. Algorithms are combined in tool box using Matlab 7.7, GUI based to design custom solution for a specific problem. The toolbox contains several blocks which facilitates the end users to simulate these algorithms by just dragging and dropping them into their required files. These reduce the job of end users in coding these blocks by using various software's. Direct comparison of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 01, January-2018

algorithms and filters are always easy. The Design and development Multi target tracking system is successfully carried out and simulated.

REFERENCES

Journals and conference papers

[1] Yaakov Bar Shalom, "The probabilistic data association filter", IEEE Control system magazine, Dec-2009 pg. 82-100
[2] Ristic, B.; Hernandez, M.L, "Tracking systems", IEEE Radar conference 2008 Page(s): 1 – 2
[3] Gorji, A.A.; Tharmarasa, R.; Kirubarajan, T., "Tracking multiple unresolved targets using MIMO Radars",Aerospace Conference 2010 IEEE, Publication Year: 2010, pg 1-14.
[4] Jian Wang; Qianhao Liang; Kun Liang; Wei Shangguan, "A new extended kalman filter based carrier tracking loop" Microwave antenna propagation and EMC Technologies foe wireless communications 2009 3rd IEEE International symposium on Publication Year: 2009, Page(s): 1181 - 1184
[5] Zhixu Zhao; Shiqi Yu; Xinyu Wu; Congling Wang; Yangsheng Xu, A Multi-target tracking algorithm using texture for real time surveillance, IEEE International Conference on Publication Year: 2009, Page(s): 2150 – 2155
[6] M. de. Fro, A.Graziano, R. Miglioli, A. Farina, "IMMJPDA Versus MHT and kalman filter near NN correlation performance comparison", IEE Proc. Radar, Sonar Navig, Vol 144, No. 2, April 1997, pg.49-55
[7] Soonho Jeong, Jitendra K. Tugnait, "Tracking of Two Targets in Clutter with Possibly Unresolved Measurements", IEEE Transactions on Aerospace and Electronic systems, Vol 44 No.2, April 2008, pg. 748-763

Text Books and Manuals

[8] P. Ramesh Babu: "Digital Signal Processing" 2nd Edition, Scitech Publications 2003, pp. 1-5
[9] A. Farina, F.A. Studer, "Radar data processing. Introduction and tracking" (vol. I). Researches Studies Press. England, John Wiley, 1985.
[10] Samuel Blackman and Robert Popoli, "Design and Analysis of Modern Tracking Systems", Artech House, 1999.
[11] Yaakov Bar-shalom "Multitarget Multisensor Tracking: Advanced applications" , Artech House, Inc. 1990
[12] Brain D Haln, 'Essential MATLAB for Scientists and Engineers", John Wiley & sons, Inc. 1997
[13] Eva Part-Enander and Anders Sjoberg. "The Matlab Handbook 5", Addison Wesley, 1999.
[14] Marchand(Patrick) Holland (O Thomas) : "Graphics and GUI with Matlab" Edition-3 Publishing, 2003