Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp344-350)

Simulated Performance Evaluation of a Selective Tracker Through Random Scenario Generation

Dil Muhammad Akbar Hussain, Zaki Ahmed*
Department of Computer Science & Engineering
Aalborg University
Niels Bohs Vej 8, 6700, Esbjerg
DENMARK
http://www.cs.aue.auc.dk/~akbar/
* School of Electronic, Communication and Electrical Engineering
University of Hertfordshire
College Lane, Hatfield, Herts AL10 9AB UNITED KINGDOM
Office Tel. +44 1707286279

Abstract: The paper presents a simulation study on the performance of a target tracker using selective track splitting filter algorithm through a random scenario implemented on a digital signal processor. In a typical track splitting filter all the observation which fall inside a likelihood ellipse are used for update, however, in our proposed selective track splitting filter less number of observations are used for track update. Much of the previous performance work [1] has been done on specific (deterministic) scenarios. One of the reasons for considering the specific scenarios, which were normally crossing targets, was to test the efficiency of the track splitting algorithm for different situations. However this approach only gives a measure of performance for a specific, possibly unrealistic, scenario and it was felt appropriate to develop procedures that would enable a more general performance assessment. Therefore, a random target motion scenario is adopted. Its implementation in particular for testing the proposed selective track splitting algorithm using Kalman filters is investigated through a number of performance parameters which gives the activity profile of the tracking scenario. This kind of performance evaluation can provide a guide line for developing appropriate tracking systems of various categories.

Keywords: Simulation, Multiple Target Tracking, State estimation, Track splitting algorithm.

1. Background

Tracking of a single target, in the ideal situation where one noisy measurement is obtained at each radar scan, can be done using standard Kalman filter techniques. In the multi-target case, an unknown number of measurements are received at each radar scan and, assuming no false measurements, each measurement has to be associated with an existing or new target tracking filter. When the targets are well apart from each other then forming a measurement prediction ellipse around a track to associate the correct measurement with that track is a standard technique [2]. When targets are near to each other, then more than one measurement may fall within the prediction ellipse of a filter and prediction ellipses of different filters may interact. The number of measurements accepted by a filter will therefore be quite sensitive in this situation to the accuracy of the prediction ellipse. Several approaches may be used for this situation [3][4][5], one of that is called the track splitting algorithm. In this algorithm, if n measurements occur inside a prediction ellipse, then the filter branches or splits into n tracking filters. This situation, which results in an increased number of filters, makes the algorithm computationally expensive. Some mechanism for restricting the excess tracks that originated from track splitting is required, since eventually this process may result in more than one filter tracking the same target. The first criterion is the support function which uses the likelihood function of a track as the pruning criterion and the second the similarity criterion which uses a distance threshold to prune similar filters tracking the same target [1].

2. Selective Track Splitting Filter

As it is stated earlier that in standard track splitting filter all those observations falling inside the likelihood ellipse are equally probable for update. In the selective track splitting filter we select at the most three of these observations nearest to the predicted position of the track. The argument for such a selection is that in fact if n number of observations are inside the most likelihood ellipse then (n – 1) wrong track observation pairing (updates) will take place. Therefore, most of the observations are false update so making update with false observations not only affects accuracy but also memory requirement and computational load increases exponentially if this track update persists for a while. However, in our selective track splitting filter algorithm we select at the most three observations closer to the predicted position inside the likelihood ellipse. The simulation results obtained with this strategy are encouraging for a moderate realistic number of targets in the same vicinity. The system performance parameters obtained provide more statistics of this selective strategy.

3. Motion Model Consideration

The motion of a target being tracked is assumed to be approximately linear and modeled by the equations

$$\underline{x}_{n+1} = \Phi \underline{x}_n + \Gamma \underline{w}_n \quad (1)$$

$$\underline{z}_{n+1} = H \underline{x}_{n+1} + \underline{v}_{n+1} \quad (2)$$

Where the state vector

$$\underline{x}_{n+1}^T = (x \quad \dot{x} \quad y \quad \dot{y})_{n+1} \quad (3)$$

is a four-dimensional vector, $\underline{w}_n$ the two-dimensional disturbance vector, $\underline{z}_{n+1}$ the two dimensional measurement vector and $\underline{v}_{n+1}$ is the two-dimensional measurement error vector. Also Φ is the assumed (4x4) state transition matrix, Γ (4x2) is the excitation matrix and H (2x4) is the measurement matrix and they are defined respectively,

$$\Phi = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\Gamma = \begin{bmatrix} \Delta t^2/2 & 0 \\ \Delta t & 0 \\ 0 & \Delta t^2/2 \\ 0 & \Delta t \end{bmatrix} \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

Here Δt is the sampling interval and corresponds to the time interval (scan interval) assumed constant, at which radar measurement data is received.

The system noise sequence $\underline{w}_n$ is a two dimensional Gaussian white sequence for which

$$E(\underline{w}_n) = 0 \quad (7)$$

where E is the expectation operator. The covariance of $\underline{w}_n$ is

$$E(\underline{w}_n \quad \underline{w}_m^T) = Q_n \delta_{nm} \quad (8)$$

where Q_n is a positive semi-definite (2x2) diagonal matrix and δ_{mn} is the Kronecker delta defined as

$$\delta_{nm} = \begin{Bmatrix} 0 & n \neq m \\ 1 & n = m \end{Bmatrix}$$

The measurement noise sequence ν_n is a two-dimensional zero mean Gaussian white sequence with a covariance of

$$E(\nu_n \quad \underline{\nu}_m^T) = R_n \delta_{nm} \qquad (9)$$

where R_n is a positive semi-definite symmetric (2x2) matrix given by

$$R_n = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \qquad (10)$$

σ_x^2 and σ_y^2 are the variances in the errors of the x, y position measurements, and σ_{xy} is the covariance between the x and y measurement errors. It is assumed that the measurement noise sequence and the system noise sequence are independent of each other, that is

$$E(\nu_n \quad w_m^T) = 0 \qquad (11)$$

The initial state $\underline{x}_0$ is also assumed independent of the $\underline{w}_n$ and ν_n sequences that is

$$E(\underline{x}_0 \quad w_n^T) = 0 \qquad (12)$$
$$E(\underline{x}_0 \quad \nu_n^T) = 0 \qquad (13)$$

$\underline{x}_0$ is a four dimensional random vector with mean $E(\underline{x}_0) = \hat{\underline{x}}_{0/0}$ and a (4x4) positive semi-definite covariance matrix defined by

$$P_0 = E\left[(\underline{x}_0 - \overline{\underline{x}}_0)(\underline{x}_0 - \overline{\underline{x}}_0)^T\right] \qquad (14)$$

where $\overline{\underline{x}}_0$ is the mean of the initial state $\underline{x}_0$. The Kalman filter is an optimal filter as it minimizes the mean squared error between the estimated state and the true (actual) state provided the target dynamics are correctly modeled.

The standard Kalman filter equations for estimating the position and velocity of the target motion described by equations [1] & [2] are;

$$\hat{\underline{x}}_{n+1/n} = \Phi \hat{\underline{x}}_n \qquad (15)$$
$$\hat{\underline{x}}_{n+1} = \hat{\underline{x}}_{n+1/n} + K_{n+1}\underline{\nu}_{n+1} \qquad (16)$$
$$K_{n+1} = P_{n+1/n} H^T B_{n+1}^{-1} \qquad (17)$$
$$P_{n+1/n} = \Phi P_n \Phi^T + \Gamma Q_n^F \Gamma^T \qquad (18)$$
$$B_{n+1} = R_{n+1} + HP_{n+1/n} H^T \qquad (19)$$
$$P_{n+1} = (I - K_{n+1} H) P_{n+1/n} \qquad (20)$$
$$\underline{\nu}_{n+1} = \underline{z}_{n+1} - H\hat{\underline{x}}_{n+1/n} \qquad (21)$$

Where $\hat{\underline{x}}_{n+1/n}$, $\hat{\underline{x}}_{n+1}$, K_{n+1}, $P_{n+1/n}$, B_{n+1} and P_{n+1} are the predicted state, estimated state, the Kalman gain matrix, the prediction covariance matrix, the covariance matrix of innovation, and the covariance matrix of estimation respectively. Q_n^F is the covariance of the measurement noise assumed by the filter which is normally taken equal to Q_n. In a practical situation, however, the value of Q_n is not known so the choice of Q_n^F should be such that the filter can adequately track any possible motion of the target. To start the computation an initial value is chosen for P_0. Even if this is a diagonal matrix, then clearly from the above equations the covariance matrices B_{n+1}, P_{n+1}, $P_{n+1/n}$ for a given n do not remain diagonal when R_n is not diagonal.

4. Random Scenario Generation

The targets move in the X-Y plane, and the positions of the targets are considered with respect to the tracker located at the fixed origin of coordinates. The initial target positions are randomly selected in a predefined tracking window such that they are uniformly distributed inside that space. The directions of the targets are also randomly selected between 0 - 2π. The initial velocity of the targets is taken from a random distribution by specifying a mean and standard deviation for the velocity. The targets follow a random velocity path given by eqn. (1). The data for different target scenarios is generated by specifying the target density, the mean value and variance for the initial velocity and the probability of detection. Other parameters such as the radar resolution, acceleration noise in the target model and measurement noise can also be changed according to the requirements of the scenario. The density of the targets for a complete run remains constant by replacing those targets which leave the tracking window by other targets whose initial positions, velocity and heading are again selected randomly as described earlier.

5. Performance Measure

A single parameter for the performance evaluation of a multiple target tracking algorithm is difficult to obtain. The target tracking problem is statistical in nature and many factors enters performance assessment. For example, one tracking algorithm may be computationally efficient but lose true tracks for a significant time. On the other hand another algorithm may perform better in tracking accuracy and rarely lose the true tracks but require more computation time. A practical approach for the assessment of a multiple target tracking algorithm is to use simulation studies, typically analytical methods are some what complicated. We are investigating three parameters which seem logical for the described situation:

- **T_{error}:** The average tracking error that is the difference between true target positions and estimated positions.

- **N_b:** The possible number of branches.

- **C_o:** Correct Observation, the average number of time a correct observation is selected for update.

As it is said earlier we are using Kalman filters, although a less expensive α-β algorithm in terms of space and computation is more attractive but simulations have shown that the trade-off in using Kalman filter is; better measurement prediction ellipse and support function assessment, which are important factors when multiple target exist [6][7][8][9]. The average tracking error for the x coordinate is given by;

$$T_x(\nabla) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\frac{1}{b}\sum_{j=1}^{b}(X_t(\nabla)^i - \hat{X}_e(j)^i)^2} \quad (22)$$

$$T_y(\nabla) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\frac{1}{b}\sum_{j=1}^{b}(Y_t(\nabla)^i - \hat{Y}_e(j)^i)^2} \quad (23)$$

where k is the number of scans, b is the number of branches belonging to the tree of track ∇ (maxima for b is 3), $X_t(\nabla)^i$ is the true position of track ∇ (noise free measurement known from the measurement generation program) at scan i and $\hat{X}_e(j)^i$ is the track estimate of branch j at scan i. The global average tracking error is then defined as;

$$T_{error} = \sqrt{\frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}[T_x(i) + T_y(i)]^2} \quad (24)$$

Even when all measurements are correctly taken by a filter from one target an average tracking error will exist due to the statistical nature of the problem. In a multiple target tracking environment, however, since incorrect measurements may also be taken by a specific filter this also affect the tracking accuracy.

N_b is another important parameter which tells actually what kind of activity present in the tracking area. A value of unity **N_b**

indicates that there is no track splitting. In a multiple target tracking environment with crossing targets, maneuvering targets and false measurements, a unit value for N_b will not be possible at various stages of tracking. Another parameter we selected provides information about system performance is C_o which gives statistics that how many time systems was able to select a correct measurement/observation. In a multiple target tracking scenario a correct target motion model is not enough to ensure successful track maintenance due to the presence of multiple measurements, since the acceptance of a measurement from any neighboring target may result in termination of the true track through similarity. Because of the track splitting process the lost target may be absorbed by the neighboring targets.

6. Implementation & Simulation

To simulate the output of a radar, a data generator routine was written in C to run on a TMS320C6713 DSK board and the parameters describing the simulation can entered interactively by the user or defined as default prior to compiling. The trajectories for the targets are generated using the kinematics described in eqn. (1) namely a constant velocity motion with acceleration noise. Simulation results for three random scenarios with target densities giving 8, 15 and 25 targets respectively have been obtained. The normalized performance evaluation parameters for these scenarios are given in table 1. The tracking window space is 25 by 25 and the results are for a single run of 100 scan intervals. From table 1 it can also be seen that the performance evaluation parameters do not vary a lot if the density of targets in the tracking window is reasonable, meaning as long as the targets are spread over the whole space.

Figure 1 shows observation paths for 15 targets. At some places targets are very close to each other and giving us the illusion of crossing or moving parallel to each other. In fact the tracking algorithm only perform badly if the crossing or parallel movement is at the same scan. However, this is not the case in this particular scenario, therefore, the tracking algorithm is able to track it very efficiently and the track paths (filtered) are shown in Figure 2 (joining lines indicate the re-appearing of targets after they leave the window to keep the density constant inside the window).

Performance Parameters			
	8 Targets	15 Targets	25 Targets
T_{error}	0.09	0.12	0.15
N_b	0.000	0.002	0.0028
C_o	0.999	0.988	0.975

Table 1: Tracking Window (25 x 25)

Here, we would like to point out that as the target position, target heading are all randomly selected so depending upon the seed they can appear any where in the tracking window.

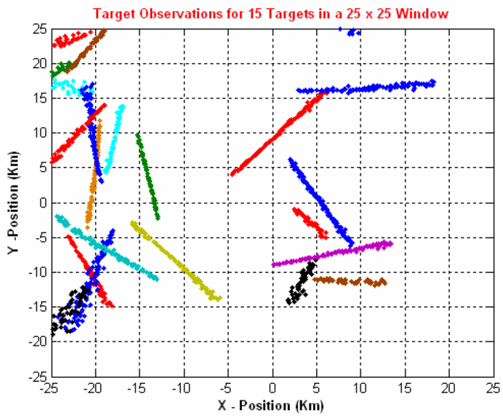

Figure 1: Target Observations

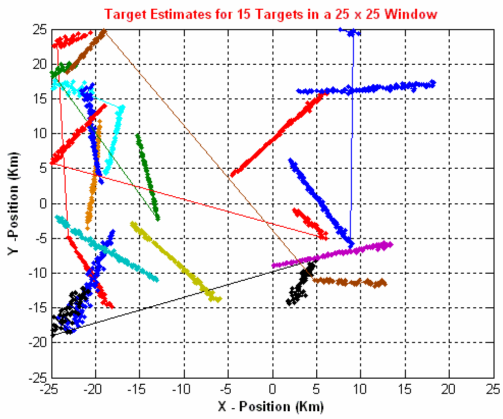

Figure 2: Track Paths

In our second simulated scenario with 25 targets appearing in 10 x 10 windows, the performance is degraded as one might expect. The probable reason being targets are in close proximity at the same scan number. Table 2 gives the normalized performance parameters for this scenario and it can be seen as the number is increased performance parameters are showing degradation.

Performance Parameters			
	8 Targets	15 Targets	25 Targets
T_{error}	0.09	0.14	0.20
N_b	0.000	0.004	0.006
C_o	0.999	0.980	0.961

Table 2: Tracking Window (10 x 10)

Figure 3 shows the targets observation paths and it can be seen that not only targets are close to each other but also they are re-appearing at various positions randomly at the same scan time. Figure 4 shows the track paths for these targets the joining lines are kept to show target belongings once they disappear/re-appear.

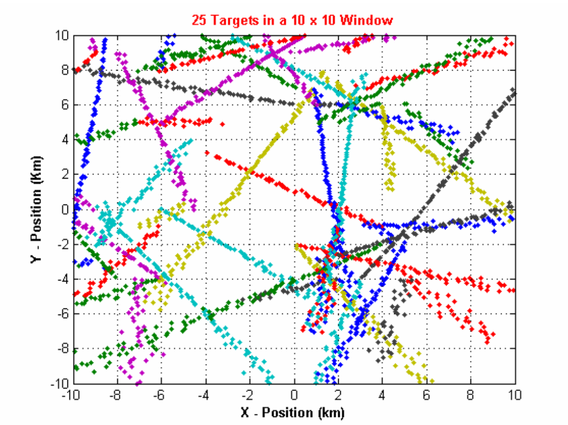

Figure 3: Target Observations

7. Conclusions

In this paper the performance of a selective track splitting algorithm, using a random scenario has been studied. The track splitting approach requires a large number of tracking filters so not only less accurate tracking is observed but also memory and computational requirements grow exponentially if multiple targets exist in the same vicinity for a longer period of time. As expected the study here has found that when the tracking window becomes denser all the performance parameters deteriorate exponentially. However, the performance of the system with selective track splitting filter is still satisfactory for a moderated number of targets in the same vicinity, which is the most likely situation to occur. The chaotic or un-realistic number of targets in the same vicinity represents an unrealistic situation. The obtained parameters values can help in the design and development of a tracking system. Also, obtaining empirical values for various performance parameters provide a more in depth vision to understand the situation. This study has used a simple simulated approach instead of more complicated analytical and finds that former may be the best approach.

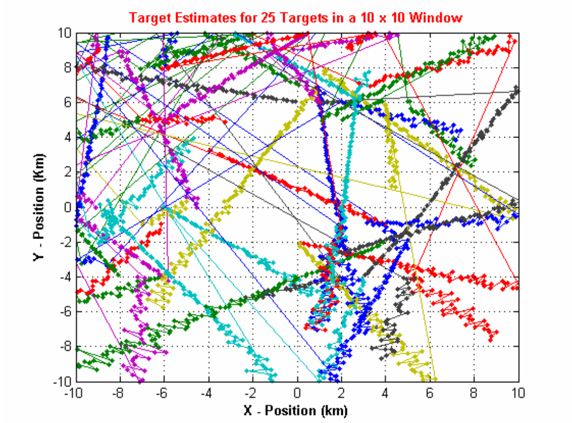

Figure 4: Target Paths

8. References

[1] D. P. Atherton, E. Gul, A. Kountzeris and M. Kharbouch; "Tracking Multiple Targets Using Parallel Processing", Proc IEE, Part D, No. 4, July 1990, pp 225-234.

[2] P. L. Smith and G. Buechler; "A Branching Algorithm for Discriminating and Tracking Multiple Objects ", IEEE Trans. Automatic Control, Vol. AC-20, February-1975, pp. 101-104.

[3] D. P. Atherton, Deacon, C; "Tracking studied of two crossing targets IFAC identification and system parameter estimation symposium, university of York, July 1985, pp 637-642.

[4] Y. Bar-Shalom and T. E. Fortmann; Tracking and Data Association, Academic Press, Inc. 1988.

[5] S. S. Blackman; Multiple-Target Tracking with Radar Applications, Artech House, Inc. 1986.

[6] M. Kharbouch; "Some investigations on target tracking" D Phil thesis, Sussex University 1991.

[7] D. M. Akbar Hussain: Tracking multiple objects using modified track-observation assignment weight approach for data association, INMIC-2003, International Multi-topic Conference, 08 – 09 December 2003, Islamabad Pakistan.

[8] D. M. Akbar Hussain, Michael Durrant, Jeff Dionne: "Exploiting the Computational Resources of a Programmable DSP Micro-processor (Micro Signal Architecture MSA) in the field of Multiple Target Tracking" SHARC International DSP Conference 2001, September 10 – 11, 2001, North Eastern University Boston USA.

[9] D. P. Atherton, D. M. A. Hussain, E. Gul; "Target Tracking using transputers as parallel processors", 9th IFAC Symposium on Identification and system Parameter Estimation, Budapest, Hungary July 1991.