Multi-Target Tracking With Target State Dependent Detection

Taek Lyul Song, Darko Mušicki, and Yong Kim
Department of Electronic Systems Engineering
Hanyang University, Ansan
Republic of Korea
Email: tsong@hanyang.ac.kr, darko.musicki@gmail.com, kimyong@hanyang.ac.kr

Abstract— Target tracking in surveillance, guidance, radar, sonar systems plays an important role for prediction and estimation of target’s information. In reality, target tracking is usually attended in imperfect target detection environment and in cluttered environment caused by false alarm, multipath fading, and non-specific noise. Target tracking algorithms usually treat the probability of detection as independent of the target state. In most cases, this assumption is not true, with subsequent degradation in the target tracking performance from both expected and optimal levels. One typical example is the Doppler frequency based clutter rejection, the other is obfuscation (shadowing) of ground based targets, and the third is anti-jamming notch filtering. In this paper, we present an algorithm for multi-target tracking with target state dependent detection.

I. INTRODUCTION

Target tracking in the cluttered environment requires automatic track initiation and maintenance using measurements. Each true track follows a target, and false tracks do not follow targets. False track discrimination is a necessary procedure which tries to recognize and confirm true tracks, and tries to recognize and terminate false tracks. False track discrimination uses various forms of track quality measures. The probability of target existence obtained by utilizing Markov chain propagation models and Bayes update is used as the track quality measure in single scan tracker of [1] and multi-scan tracker of [2]. The integrated track splitting (ITS) filter for single target tracking in clutter, integrates multi-scan target tracking with probability of target existence, which becomes the measure of track quality in ITS.

Critical parameters that affect performance of false track discrimination methods are the probability of target detection \( P_D \) and the clutter measurement density \( \rho \). The probability of target existence and the track score track quality measures [3] are functions of \( P_D \) and \( \rho \). The usual assumption of the vast majority of target tracking algorithms is a constant \( P_D \) determined by the signal to noise ratio and the probability of false alarms. In reality, the probability of detection depends on the target trajectory state due to multipath fading, terrain shadowing, effect of signal processing (clutter rejection filtering), and beam (anti-jamming notch) processing, to name a few. The consequence of wrong \( P_D \) assumptions in practice often results in track loss or inadequate (large rms error) tracking.

In this paper we extend the target tracking with target state dependent detection using Gaussian mixture solution to multi-target tracking using JITS(Joint ITS).

The problem statement is summarized in Section II. The Gaussian mixture solution for the target tracking with target state dependent detection is described in Section III. Section IV presents extension of the Gaussian mixture solution to multi-target tracking environments. The approach is evaluated using simulations in Section V followed by concluding remarks in Section VI.

II. PROBLEM STATEMENT

Target tracking is the process which decides target existence and trajectory by comparing stochastic models with measurements received by the sensor. This section describes models used in target tracking.

A. Target model

Targets randomly enter and exit the surveillance space, thus the existence of targets are a priori unknown. The target existence propagation is modeled as a Markov process. Denote by \( \chi_k \) the target existence event at time \( k \). The probability that the target exists at measurement time \( k \), given that it did exist at \( k−1 \) is

\[
P(\chi_k | \chi_{k−1}) = 1 - \frac{\Delta T_{k−1,k} \chi_k}{T_{ave}}
\]

where \( \Delta T_{k−1,k} \chi_k \) denotes the time interval between measurement times \( k−1 \) and \( k \), and \( T_{ave} \) denotes the average duration of target existence in the surveillance space [2], [4]. The complementary probability \( P(\bar{\chi}_k | \bar{\chi}_{k−1}) \) that the target exists at time \( k \), given that it did not exist at \( k−1 \) is assumed to be zero. The targets which did not exist at \( k−1 \) and then appear at \( k \) are new targets and their tracks will be separately initialized by their measurements.

In this paper we use the Markov Chain One model where, if the target exists, it is always detectable, i.e., the target measurement exists in each scan with a probability of
detection (which in this case depends on the target trajectory state). In each scan, each existing target generates zero or one measurement. Given the sensor trajectory, the probability of target detection is a function of target trajectory state \( x_k \), and is denoted by \( P_D(x_k) \) and assumed known. A maneuvering target is assumed. The target trajectory is assumed to follow one of \( M \) models.

\[
x_k = F_k(\sigma_k) x_{k-1} + v_{k-1}
\]

(2)

where \( \sigma_k \) is target trajectory model between time \( k-1 \) and \( k \), the process noise \( v_k \) is zero mean and white Gaussian sequence with covariance matrix \( Q(\sigma_k) \).

B. Sensor model

At time \( k \) one sensor forms a set of measurements, which may originate from targets or other objects. Non-target originated measurements are termed clutter. The infinite resolution sensors are assumed, where each measurement has only one source. When target is detected, its measurement \( y_k \) is

\[
y_k = h(x_k) + w_k
\]

(3)

where \( h(\cdot) \) can be a nonlinear function, the measurement noise \( w_k \) is zero mean and white Gaussian sequence with covariance matrix \( R \).

C. Clutter model

The general assumption about clutter measurements is the Poisson distribution. However, in non-uniform cluttered environment, the clutter measurement density \( \rho(y) \) at position \( y \) can be estimated. The expected number of selected clutter measurements is

\[
\hat{m}_k = \int y \rho(y) dy
\]

(4)

and its probability density function(pdf) is

\[
p_c(y) = \frac{\rho(y)}{\hat{m}_k}
\]

(5)

and the number of clutter satisfies the Poisson distribution given by following equation:

\[
P_p(m; \hat{m}_k) = \exp(-\hat{m}_k) \frac{\hat{m}_k^m}{m!}.
\]

(6)

III. GAUSSIAN MIXTURE SOLUTION

The target tracking algorithm presented in [5] provides a single target closed-form Gaussian mixture solution. Assuming that the prior target trajectory state pdf is approximated by a Gaussian mixture [6], and that the expression for \( P_D(x_k) \) may be presented by a form of (7) below, we show that updated target trajectory state pdf remains approximated by a Gaussian mixture. The probability of detection \( P_D(x_k) \) may often be naturally modeled by

\[
P_D(x_k) = D_0 - \sum_d W_d N(\delta_d; h_d(x_k), R_d)
\]

(7)

where \( d \) represents a “notch” or a blind spot. This may be due to the Doppler (MTI) clutter rejection. In (7), \( D_0 \) is the maximum probability of detection. Each function \( h_d(x_k) \) is a projection of target trajectory state, e.g., Doppler velocity. Maximum attenuation of the probability of detection occurs when the projection \( h_d(x_k) \) equals \( \delta_d \). The variance \( R_d \) determines the width of the clutter rejection notch. We assume that parameters mentioned in this paragraph \((D_0, W_d, \delta_d, h_d, R_d)\) are a priori known values. Conditions \( P_D(x_k) \geq 0 \) and \( W_d \geq 0 \) should be observed. Jacobian \( H_d \) of function \( h_d(x_k) \) is

\[
H_d(x) = \frac{dh_d(x)}{dx}.
\]

(8)

The updated target trajectory state pdf is represented by

\[
p(x_k | \chi_k, Z^{k-1}) = \frac{p(x_k | \chi_k, h_k, R_k) p(x_k | \bar{Z}, Z^{k-1})}{p(x_k | \chi_k, Z^{k-1})} = \frac{\lambda_k(\chi_k)}{\Lambda_k} p(x_k | \chi_k, Z^{k-1}).
\]

(9)

We can respectively divide the denominator and numerator in (9) by \( p(x_k | \bar{Z}, Z^{k-1}) \). Then (9) becomes

\[
p(x_k | \chi_k, Z^{k-1}) = \frac{\lambda_k(\chi_k)}{\Lambda_k} p(x_k | \chi_k, Z^{k-1}).
\]

(10)

where

\[
\lambda_k(\chi_k) = 1 - P_G P_D(x_k) + P_D(x_k) P_G \sum_{i=1}^{M_k} \frac{f_{k,i}(x_k)}{\rho_{k,i}}
\]

(11)

\[
f_{k,i}(x_k) = \frac{p(Z_{k,i} | x_k)}{P_G} = \frac{N(z_{x,k}; h(x_k), R)}{P_G}
\]

(12)

After mixing and prediction steps of IMM estimation [7] for each track component \( c_{k-1} \), the predicted target trajectory state pdf of the ITS filter becomes

\[
p(x_k | \chi_k, Z^{k-1}) = \sum_{i=1}^{c_{k-1}} \xi(c_{k-1}) \sum_{\sigma_i} \mu_{\sigma_i} N(x_k; \hat{x}_{x,\sigma}, P_{x,x})
\]

(13)

where \( \xi(c_{k-1}) \) is component weight of target trajectory pdf, and the mode probability of the discrete trajectory model \( \sigma_k \) is defined as

\[
\mu_{\sigma_c} = P(\sigma_k | c_{k-1}, Z^{k-1}).
\]

(14)
Ultimately, the updated target trajectory state pdf in (10) is expressed by
\[
p(x_k | \chi_k, Z^k) = \sum_{c_k} \sum_{\sigma_k} \check{\xi}(c_k) \mu_{c_k}(c_k, \sigma_k) \cdot N(x_k; \hat{x}_{c_k}, \sigma_k, P_{c_k}(c_k, \sigma_k))
\]  
(15)

By comparing (15) with (10) the expression for the component weight \( \check{\xi}(c_k) \) is obtained. In the process, \( \Lambda_k \) in (10) can be obtained as
\[
\Lambda_k = \sum_{c_{k-1}} \sum_{\sigma_k} \check{\xi}(c_{k-1}) \sum_{\sigma_k} \mu_{c_k} \cdot \int_{x_k} \check{\lambda}_k(x_k) N(x_k; \hat{x}_{c_k}, P_{c_k}) dx_k
\]  
(16)

The new track component \( c_k \) which is formed after updating measurement can be expressed by \( \{c_{k-1}, i, d, 0\} \) or \( \{c_{k-1}, 0, 0\} \). When measurement \( z_{k,i} \) does not exist, \( c_k \) is \( \{c_{k-1}, 0, 0\} \) or \( \{c_{k-1}, i, 0\} \). When measurement \( z_{k,i} \) does exist, \( c_k \) is \( \{c_{k-1}, i, d, 0\} \) or \( \{c_{k-1}, i, d\} \).

The results of updating process is summarized as follows.
\[
[\hat{x}_{c_{i,0}}^0, P_{c_{i,0}}] = \text{EKF}_u(\hat{x}_{c_{i,0}}, P_{c_{i,0}}, z_{k,i}, R, h)
\]  
(17)
\[
[\hat{x}_{c_{i,0}}^d, P_{c_{i,0}}^d] = \text{EKF}_u(\hat{x}_{c_{i,0}}, P_{c_{i,0}}, \delta_d, R_d, h_d)
\]  
(18)
\[
[\hat{x}_{c_{i,0}}^d, P_{c_{i,0}}^d] = \text{EKF}_u(\hat{x}_{c_{i,0}}, P_{c_{i,0}}, \delta_d, R_d, h_d)
\]  
(19)
where \( \text{EKF}_u(x, P, y, R, h) \) denotes standard EKF update of Gaussian with mean \( x \) and covariance \( P \) by measurement \( y \) and measurement covariance \( R \), and where \( h \) denotes the measurement function. Likelihoods of measurements are
\[
p_{c_{i,0}}^0 = \frac{1}{P} N(z_{k,i}; h(\hat{x}_{c_{i,0}}, \sigma_{c_{i,0}}), s_{c_{i,0}}^0),
\]  
(20)
\[
s_{c_{i,0}}^0 = H(\hat{x}_{c_{i,0}}) P_{c_{i,0}} H(\hat{x}_{c_{i,0}})^T + R
\]  
\[
p_{c_{i,0}}^d = N(\delta_d; h(\hat{x}_{c_{i,0}}), s_{c_{i,0}}^d),
\]  
(21)
\[
s_{c_{i,0}}^d = H_d(\hat{x}_{c_{i,0}}) P_{c_{i,0}} H_d(\hat{x}_{c_{i,0}})^T + R_d
\]  
\[
p_{c_{i,0}}^{d*} = N(\delta_d; h(\hat{x}_{c_{i,0}}), s_{c_{i,0}}^{d*}),
\]  
\[
s_{c_{i,0}}^{d*} = H_d(\hat{x}_{c_{i,0}}) P_{c_{i,0}} H_d(\hat{x}_{c_{i,0}})^T + R_d
\]  
(22)

We can calculate measurement likelihood for \( \delta_d \) by considering the trajectory model \( \sigma_k \) and track component \( c_{k-1} \) such as
\[
p_{c_k}^d = \sum_{\sigma_k} \mu_{c_k} P_{c_k}^{d*}
\]  
(23)
\[
p_0^{d} = \sum_{c_{k-1}} \check{\xi}(c_{k-1}) P_{c_k}^{d*}
\]  
(24)

The average probability of detection of \( p(x_k | c_i, \sigma_k, Z^k) \),
given \( z_{k,i} \) becomes
\[
\tilde{p}_{D, \sigma}^{c} = D_0 - \sum_d W_d p_{c}^{d*}
\]  
(25)
and
\[
\tilde{p}_{k,i}^{c} = \sum_{\sigma} \mu_{c_k} \tilde{p}_{c,k}^{c},
\]  
(26)
\[
\tilde{p}_{k,i}^{c} = \sum_{\sigma} \mu_{c_k} \tilde{p}_{c,k}^{c},
\]  
(27)
\[
\tilde{p}_{k,i}^{c} = \sum_{\sigma} \xi(c_{k-1}) \tilde{p}_{k,i}^{c}
\]  
(28)

Then component weights of updated pdf are obtained:
\[
c_k = \{c_{k-1}, i, d, 0\}; \xi(c_k) = \beta_{k,0} \xi(c_{k-1}) 1 - P_G D_k \frac{1 - P_G D_k}{1 - P_G P_D},
\]  
(29)
\[
\mu_{c_k} = \mu_{c_{k-1}} \mu_{c_k} P_{c_{k}}^{d*}
\]  
(30)
\[
\mu_{c_k} = \mu_{c_{k-1}} \mu_{c_k} P_{c_{k}}^{d*}
\]  
(31)
where
\[
\tilde{p}_k = D_0 - \sum_d W_d p_0^{d*}
\]  
(32)
\[
\beta_{k,0} = \frac{1 - P_G \tilde{p}_D}{\Lambda_k}
\]  
(33)
\[
\beta_{k,1} = \frac{P_G \tilde{p}_k}{\Lambda_k}
\]  
(34)
and \( \Lambda_k \) can be expressed by
\[
\Lambda_k = 1 - P_G \tilde{p}_D + P_G \sum_{i=1}^{\infty} \tilde{p}_{k,i}^{d*}
\]  
(35)

IV. EXTENSION TO MULTITARGET TRACKING

In this section we extend the target tracking with target state dependent detection using Gaussian mixture solution to multi-target tracking using ITS: JITS(Joint ITS). JITS uses
the optimal joint multi-target tracking approach presented in [8].

In an environment of multi-target tracking, each track initialized by 1-point track initiation or 2-point differencing [4] is predicted and updated using the Gaussian mixture solution presented in section III. In the situation that tracks share same measurement, we can replace the single target data association with the data association method of joint multi-target tracking.

JITS implements the “optimal” multi-target data association approach within the ITS framework. In this subsection only final formulae are presented, the full derivation of JITS is presented in [8]. Joint multi-target data association [9, 10] enumerates and evaluates all possible global measurement-to-track allocations. A serious drawback of this approach is that the number of global measurement-to-track allocations is combinatorial in the number of shared measurements and the number of tracks which share these measurements. To limit the computational complexity, in each scan \( k \) tracks are separated into clusters. Joint multi-target data association is then applied to each cluster separately.

The selected measurement set \( z(k) \) becomes the union of measurements validated by each track belonging to the cluster. A feasible joint event \( \varepsilon_k \) is an allocation of measurements to all tracks in the cluster at time \( k \), with \( T_k \) denoting the number of tracks in the cluster and \( m_k \) denoting the number of measurements. The following definition regarding feasible joint event (FJE) \( \varepsilon_k \) are needed.

\[
T_0(\varepsilon_k) : \text{Set of tracks allocated "No measurement"} \\
\text{under } \varepsilon_k .
\]

\[
T_1(\varepsilon_k) : \text{Set of tracks allocated "One measurement"} \\
\text{under } \varepsilon_k .
\]

\[
i(\tau, \varepsilon_k) : \text{Index of the measurement allocated to track} \\
\text{under } \varepsilon_k \tau .
\]

\[
\Xi(\tau, i) : \text{Set of feasible joint events which allocate} \\
\text{measurement } i \text{ to track } \tau .
\]

The a posteriori pdf of FJE \( \varepsilon_k \) can be represented as

\[
P\{ \varepsilon_k | Z^k \} = \Delta_k^{-1} \prod_{\tau \in T_0(\varepsilon_k)} (1 - \tilde{P}_{G}^\tau P_0^\tau P\{ \chi_k^{\tau} | Z^{k-1} \}) \\
\prod_{\tau \in T_1(\varepsilon_k)} P_0^\tau P\{ \chi_k^{\tau} | Z^{k-1} \} \tilde{P}_{k,I(\tau, \varepsilon_k)} \rho_{k,I(\tau, \varepsilon_k)}
\]

where \( \Delta_k \) is normalized coefficient. We can calculate \( \tilde{P}_{k,i} \) by using (28).

The a posteriori probability of the event \( \chi_k \), which represents that measurement \( z_{k,i} \) is allocated to track \( \tau \) is obtained by summing the a posteriori probabilities of all feasible joint events that contain \( \chi_k^{\tau} \). Denote by \( \Xi(\tau, i) \) the set of feasible joint events containing \( \chi_k^{\tau} \). The a posteriori track-to-measurement allocation probabilities are

\[
P\{ \chi_k^{\tau}, \chi_k^{\tau} | Z^k \} = \sum_{\varepsilon_k \in \Xi(\tau, i)} P\{ \varepsilon_k | Z^k \}
\]

\[
P\{ \chi_k^{\tau}, \chi_k^{\tau} | Z^k \} = \sum_{\varepsilon_k \in \Xi(\tau, i)} P\{ \varepsilon_k | Z^k \} \frac{(1 - \tilde{P}_D^\tau P_G^\tau) P\{ \chi_k^{\tau} | Z^{k-1} \}}{1 - P_D^\tau P_G^\tau P\{ \chi_k^{\tau} | Z^{k-1} \}}
\]

Then, the output values of the JITS data association module are

\[
P\{ \chi_k^{\tau} | Z^k \} = \sum_{\tau=0}^{m} P\{ \chi_k^{\tau}, \chi_k^{\tau} | Z^k \},
\]

\[
\beta_k^{\tau} = \frac{P\{ \chi_k^{\tau}, \chi_k^{\tau} | Z^k \}}{P\{ \chi_k^{\tau} | Z^k \}}.
\]

V. SIMULATION STUDY

In this section we show a simulation environment of two sensors and two targets two-dimensional surveillance situation depicted on Fig. 1. The surveillance area is scanned every second by alternate stationary sensors, sampling time \( T = 1 \) s.

![Figure 1. Simulation scenario.](image)

The first target (labeled by target 1) starts from the position and speed of (50, 350) m and (20, 0) m/s, and then progresses as follows:

- 20 s uniformly, then
- decelerates to stop in 5 s, then
- remains stationary for 5 s, then
- accelerates to the initial velocity within 5 s, and

...
• moves with the uniform motion for 15 s.

The second target (labeled by target 2) starts from the position and speed of (58.5, 414.7) m and (20, -20) m/s that heading angle is 15°, and then progresses as follows:

• 25 s uniformly, then
• decelerates to stop in 5 s, then
• accelerates to the initial velocity within 5 s, and
• moves with the uniform motion for 15 s.

These two targets cross at 12.5 s. This trajectory is modeled by two models. Model \( \sigma = 1 \) corresponds to the uniform motion, and model \( \sigma = 2 \) accommodates accelerations. Both models have the same propagation matrix

\[
F_\sigma = I_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \sigma = 1.2
\]  

(41)

where \( \otimes \) is the Kronecker product symbol, and \( I_m \) denotes the \( m \times m \) identity matrix. The plant noise covariance matrices are [20]

\[
Q_\sigma = q_0(\sigma) \cdot I_2 \otimes \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}
\]  

(42)

with \( q_0(1) = 0.006 \) and \( q_0(2) = 6 \).

The probability of detection is a function of target Doppler speed and is modeled by (7), with parameters \( W_1 = 6.8 \) m/s, \( \delta_1 = 0 \) m/s and \( R_1 = 9 \) (m/s)². The maximum probability of detection is 0.95, and there is a clutter rejection notch centered at 0 m/s, with standard deviation of 3 m/s. Minimum probability of detection (at zero velocity) equals 0.05.

The two sensors are located on (300, 150) m and (750, 100) m, respectively. The probability of detection over time is presented in Fig. 2 and Fig. 3. Fig. 2 is the case of target 1 and Fig. 3 is of target 2.

When a target is moving perpendicular to a sensor (at 13 s and 46 s for sensors 1 and 2 in the case of target 1, at 16 s and 43 s for sensors 1 and 2 in the case of target 2), the radial speed with respect to the corresponding sensor drops to zero, and the probability of detection by the sensor decreases. In both cases only one sensor suffers from this decrease of the probability of detection, and the measurements provided by the other sensor can be expected to help with the track maintenance. During the period of low target speed, both sensors experience low probability of detection over a substantial period between 26 and 30 s in the case of target 1. In that case, the absence of measurements is compensated to a large degree by expectations that the target state is within the detection notch. Clutter is uniform with measurement density of \( 10^{-5} \). Target measurement equation is linear, with the covariance matrix of \( 25I_2 \).

In this paper we compare three algorithms. One is the algorithm proposed in this paper, and the curves resulting from its application are labeled by “JITSpd”. The others are ITSpd [5] and standard JITS [8], which is a special case derived using the standard assumption of constant probability of target detection. The JITS algorithm is run with three values of assumed probability of target detection, \( P_D = 0.95, 0.5 \) and 0.1.

Tracks are initialized automatically in each scan using measurements from consecutive scans which satisfy the maximum target speed criterion of 25 m/s. Track trajectory state estimate is initialized using 2-point differencing [4], and the track probability of target existence is initialized as a constant 0.003 on the first measurement, and is updated by using measurements from the following scan. Due to the cluttered environment, both true tracks (which follow the target) and false tracks (which do not) were initialized. Additionally, a true track may become a false track if it stops following the target. Only the probability of target existence is used as the track quality measure to discriminate false tracks. If the probability of target existence falls below a termination threshold 0.001, the track is terminated. If the probability of target existence rises above a confirmation threshold 0.995, the track is confirmed. The simulation
The experiment consists of 1000 simulation runs. Each simulation run lasts for 50 scans. At the end of each simulation run the true tracks are terminated, whereas false tracks remain. True track confirmation success rate over time is shown in Fig. 4.

The result of the proposed algorithm is denoted as JITSpd. Over 1000 runs, the two targets were confirmed 100% during 50 scans. Only 13.5% of true confirmed tracks were terminated or otherwise lost during the no-detection period. On that duration, new tracks were subsequently initialized for track discrimination. On the other hand, confirmed true tracks JITS($P_D = 0.95, 0.5$ and $0.1$) were terminated more than JITSpd. In the case of ITSpd, confirmed true tracks were terminated when two targets crossed at 12.5 s and it maintained low confirmation rate during no-detection period. A direct application of the single target tracking algorithm of [5] to this multi-target tracking environment results in poor performance.

RMS estimation errors of confirmed true tracks over time are shown in Fig. 5 and Fig. 6. As expected, estimation errors increase in the non-detection period. However, these errors are limited due to the use of state dependent probability of detection in trajectory state estimation, as proposed in this paper.

![Figure 4. True track confirmation.](image1)

![Figure 5. RMS estimation errors. (target 1)](image2)

![Figure 6. RMS estimation errors. (target 2)](image3)

VI. CONCLUSIONS

For multi-target tracking environments with target state dependent detection, the previous ITS approach designed for single target tracking is extended with JITS and its performance is tested. Simulation results show that the proposed algorithm has superior performance for crossing-target and non-detectable target environments to the other multi-target tracking filter algorithms.

REFERENCES