5-8-2017

Using ML-PDA and ML-PMHT to Track Two Unresolved Moving Objects

Katherine Domrese
katherine.domrese@uconn.edu

Using ML-PDA and ML-PMHT to Track Two Unresolved Moving Objects

Katherine Ruth Domrese

B.S. Electrical Engineering, University of Connecticut, Storrs, CT, 2014

A Thesis

Submitted in Partial Fullfilment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2017

To my mother and father

ACKNOWLEDGEMENTS

First, I would like to thank my major advisor Dr. Peter Willett for recommending me to the Graduate School, inviting me to be his graduate student, and advising me the last two years. Next I would like to thank my associate advisor Dr. Yaakov Bar-Shalom for his expert advice and careful editing of my paper and presentation drafts. I am also thankful to my other associate advisor Dr. Steven Schoenecker for kindly agreeing to serve on my advisory committee.

Next, I would like to thank all of my graduate professors at UConn – Dr. Peter Willett, Dr. Yaakov Bar-Shalom, Dr. Shengli Zhou, Dr. Shalabh Gupta, Dr. Krishna Pattipati, and Dr. Robert Lynch from the ECE Department and Sherry Shamash, my Hebrew professor. They patiently taught, advised, and inspired me and showed that they cared about my academic success and well-being. I am also thankful to my labmates, fellow students, and friends at UConn for studying and collaborating on school projects with me.

Finally, I would like to thank Marcia Beaulieu, who is like an aunt to me, Robert Aleksiewicz and Richard Wilton, who have been like uncles to me throughout my life, my fiancé Shane Tobey, and my father and late mother, Robert and Claudia Domrese for their love and support.

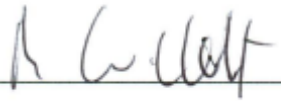
APPROVAL PAGE

Master of Science Thesis

Using ML-PDA and ML-PMHT to Track Two Unresolved Moving Objects

Presented by

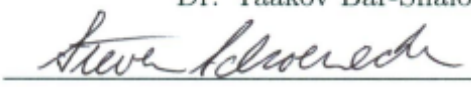
Katherine Ruth Domrese, B.S.

Major Advisor 

Dr. Peter Willett

Co-Major Advisor 

Dr. Yaakov Bar-Shalom

Associate Advisor 

Dr. Steven Schoenecker

University of Connecticut

2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Using ML-PDA and ML-PMHT to Track Two Unresolved Moving Objects

Both the Maximum Likelihood Probabilistic Data Association (ML-PDA) track extractor and the Maximum Likelihood Probabilistic Multi-Hypothesis (ML-PMHT) track extractor are extended in this work to handle the scenario of two unresolved moving objects in the field of gravity. The original ML-PDA and ML-PMHT log-likelihood ratios are modified to use the probability that the objects being tracked are unresolved i.e., their measurements are merged. The performances of the modified ML-PDA and ML-PMHT, which we denote ML-PDA-M and ML-PMHT-M (M for merged), respectively, are compared with those of the ML-PDA and the ML-PMHT in a notional scenario in which two moving objects appear initially unresolved to two space-based passive sensors observing them and become resolved first by one and then by both. Simulation results for the original track extractors and the modified track extractors are presented. While in many tracking situations the performances of the ML-PDA and ML-PMHT are indistinguishable (and the ML-PMHT therefore selected for its other features), this case of challenged resolution appears to be one situation where the more arduous ML-PDA ought to be favored. There does seem to be some reason to favor the "M" versions of both, but the results there are less compelling.

1. Introduction

Optical sensors have a finite resolution, making it possible for objects in the field of view of the sensors to be unresolved. In this case, it is not straightforward to ascertain the number of moving objects in view nor estimate tracks for each of them. Hence, a target-tracking algorithm that does not consider the possibility of unresolved (merged) measurements may yield poor results.

If the probability that moving objects in a sensor's field of view are resolved can be modeled, this can be incorporated to a target-tracking algorithm to improve its performance. Various target-tracking algorithms have been extended to handle the case of unresolved measurements. These include particle filtering **1632067** the Joint Probabilistic Data Association (JPDA) algorithm **1103597** the Nearest Neighbor Joint Probabilistic Data Association (NNJPDA) algorithm **blair** and the Probabilistic Multi-Hypothesis Tracker (PMHT) **4252476**

Both the Maximum Likelihood Probabilistic Data Association (ML-PDA) and the Maximum Likelihood Probabilistic Multi-Hypothesis (ML-PMHT) track extractors are batch algorithms that maximize the target motion parameters' log-likelihood ratio (LLR) based upon some clear assumptions. The main difference between the two algorithms is that ML-

PDA allows only at most one measurement per frame per sensor to be assigned to a single target whereas ML-PMHT allows any number of measurements obtained in a frame to be so assigned; more on this later. The ML-PMHT LLR turns out to be more easily formulated for multiple-target scenarios than the ML-PDA LLR **schoenecker˙pda** hence despite its adulterated model it is often of great interest.

In this paper, we propose the inclusion of the probability of resolution of the objects being tracked to the ML-PDA and ML-PMHT LLR's in order to detect more accurately the number of targets extant and to track them. We denote the new algorithms ML-PDA-M and ML-PMHT-M, respectively (M for merged measurements). Section 2 of this paper provides a description of the LLR's. Section 3 shows how a resolution probability model can be incorporated into the track extractors. In Section 4 a notional scenario in which the original algorithms and the modified algorithms are applied and compared is described. Section 5 details the results of 100 Monte Carlo runs, and Section 6 contains a summary of this work.

2. The Log-Likelihood Ratios

2.1. The Log-Likelihood Ratio for ML-PDA

The ML-PDA LLR for the single-target scenario was developed using the following five basic assumptions **schoenecker˙pda**

1) At most one measurement per frame per sensor originates from a target.

2) The target exhibits deterministic motion. (Its trajectory is often parameterized as a straight line.)

3) False alarms are uniformly distributed in each observing sensor's field of view, and their number is Poisson distributed with known density. (This can be relaxed.)

4) The target measurement noise is a known Gaussian. (Gaussianity can be relaxed.)

5) Frames of data, conditioned on the parameterized state, are independent. Naturally the first assumption implies strong dependence between measurements within a frame.

For our purposes, we did not consider the amplitudes of the target(s) and false alarms. The ML-PDA LLR for the single-target scenario in which multiple sensors are observing the

target is based upon that given in **schoenecker˙pda** and is as follows:

$$\Lambda_{\text{mlpda}}(\mathbf{x}_1;\mathbf{Z}) = \ln\left(\prod_{k=1}^{N_w}\prod_{s=1}^{s_n}\left((1-P_d)+\frac{1}{\lambda}P_d\sum_{j=1}^{m(k,s)}p(z_j(k,s)|\mathbf{x}_1)\right)\right) \tag{2.1}$$

where N_w is the batch length, s_n is the number of sensors, $m(k,s)$ is the number of measurements (a pair of azimuth and elevation angles counts as one measurement) obtained at frame k by sensor s, λ is the spatial clutter density, P_d is the probability of detection, and $z_j(k,s)$ is the jth measurement vector containing the azimuth and elevation angle measurements obtained at frame k by sensor s. The LLR in (2.1) can be extended to the two-target scenario as follows:

$$\begin{aligned}\Lambda_{\text{mlpda}}(\mathbf{x};\mathbf{Z}) = \ln\Bigg(\prod_{k=1}^{N_w}\prod_{s=1}^{s_n}\Bigg((1-P_d)^2 + (1-P_d)\frac{P_d}{\lambda}\sum_{j=1}^{m(k,s)}\big(p(z_j(k,s)|\mathbf{x}_1)\\ +\ p(z_j(k,s)|\mathbf{x}_2)\big) + \left(\frac{P_d}{\lambda}\right)^2\sum_{\substack{i=1\\ i\neq j}}^{m(k,s)}\sum_{j=1}^{m(k,s)}p(z_j(k,s)|\mathbf{x}_1)p(z_j(k,s)|\mathbf{x}_2)\Bigg)\Bigg)\end{aligned} \tag{2.2}$$

where

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \\ \mathbf{x}_2\end{bmatrix} \tag{2.3}$$

represents the combined state of the two targets to be estimated.

2.2. The Log-Likelihood Ratio for ML-PMHT

The assumptions used to develop the ML-PMHT LLR are the same as those for the ML-PDA with one exception. Measurement association events (i.e., whether a measurement comes from any particular target or clutter) are independent. The ML-PMHT LLR for the single-target scenario in which there are multiple sensors is based upon that given in **schoenecker˙2011** and is as follows:

$$\Lambda_{\text{mlpmht}}(\mathbf{x}_1;\mathbf{Z}) = \sum_{k=1}^{N_w}\sum_{s=1}^{s_n}\sum_{j=1}^{m(k,s)} \ln\left\{\pi_0(k,s) + V\pi_q(k,s)p[z_j(k,s)|\mathbf{x}_1]\right\} \tag{2.4}$$

where V is the search volume. The values of $\pi_0(k,s)$, the prior probability that a measurement obtained in frame k by sensor s is from clutter, and its complement $\pi_q(k,s)$, the prior probability that a measurement obtained in frame k by sensor s is from a target, depend upon P_d. Here we give approximations of the probabilities in **pmht˙prob˙sol** and **pmht˙crouse**

$$\pi_0(k,s) = 1 - \pi_q(k,s) \tag{2.5}$$

where $\pi_q(k,s)$ is defined by

$$\pi_q(k,s) = \frac{P_d}{m(k,s)} \tag{2.6}$$

and $m(k,s)$ is the number of measurements obtained in frame k by sensor s.

The LLR in (2.4) can be easily extended to the multiple-target scenario. For the sake of simplicity and the purpose of our present research, we give here the LLR, based upon that

in **schoenecker˙2011** which assumes that there are two targets in existence and that the measurements are resolved

$$\Lambda_{\text{mlpmht}}(\mathbf{x};\mathbf{Z}) = \sum_{k=1}^{N_w}\sum_{s=1}^{s_n}\sum_{i=1}^{m(k,s)} \ln\{\pi_0(k,s) + V\left(\pi_1(k,s)p[z_j(k,s)|\mathbf{x}_1]\right. \\ \left. + \pi_2(k,s)p[z_j(k,s)|\mathbf{x}_2]\right)\} \tag{2.7}$$

where $\mathbf{x}$ is defined as in (2.3). The prior probability that a measurement obtained in frame k by sensor s originates from object n for $n = 1, 2$ is denoted $\pi_n(k, s)$ and can be expressed as follows:

$$\pi_n(k,s) = \frac{1}{2}\pi_q(k,s). \tag{2.8}$$

3. Unresolved Objects and the Algorithms

3.1. Object Resolution Probability Model

The object resolution probability model used was based upon that proposed in **chang** and is as follows:

$$P_{r(m,n)} = \begin{cases} 1 & d_{mn} \geq DW \\ f(d_{mn}, SNR_m, SNR_n) & \frac{1}{2}\ DW \leq d_{mn} \leq DW \\ 0 & d_{mn} \leq \frac{1}{2}\ DW \end{cases} \tag{3.1}$$

where d_{mn} is the distance between object m and object n, DW is the detection width, and $f(d_{mn}, SNR_m, SNR_n)$ for high SNR can be approximated as

$$f(d_{mn}, SNR_m, SNR_n) = \frac{2d_{mn}}{DW} - 1. \tag{3.2}$$

For our purposes, we chose to define DW as twice the sensor pixel width.

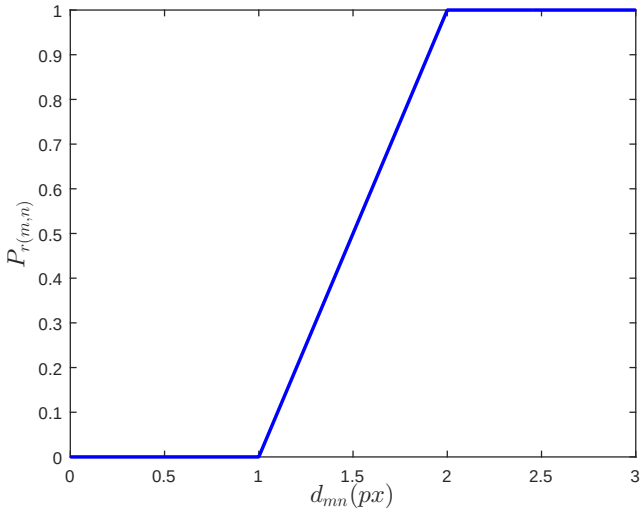


Figure 3.1: Probability of resolution $f(d_{mn}, SNR_m, SNR_n)$ for $DW = 2px$.

3.2. The ML-PDA-M and ML-PMHT-M

Using the object resolution probability model, the LLR in (2.2) can be extended as follows to form the two-target assumption LLR for the ML-PDA-M (M for merged):

$$\begin{aligned}\Lambda'_{\text{mlpda}}(\mathbf{x};\mathbf{Z}) = \ln\Bigg\{\prod_{k=1}^{N_w}\prod_{s=1}^{s_n}\Bigg[\Big(1-P_r(k,s,\mathbf{x})\Big)\left((1-P_d)+\frac{1}{\lambda}P_d\sum_{j=1}^{m(k,s)}p_\rho(z_j|\mathbf{x}_q)\right)\\ +\ P_r(k,s,\mathbf{x})\Bigg((1-P_d)^2+(1-P_d)\frac{P_d}{\lambda}\sum_{j=1}^{m(k,s)}\left(p(z_j|\mathbf{x}_1)+p(z_j|\mathbf{x}_2)\right)\\ +\ \left(\frac{P_d}{\lambda}\right)^2\sum_{\substack{i=1\\ i\neq j}}^{m(k,s)}\sum_{j=1}^{m(k,s)}p(z_i|\mathbf{x}_1)p(z_j|\mathbf{x}_2)\Bigg)\Bigg]\Bigg\}\end{aligned} \tag{3.3}$$

where the state to be estimated is defined as in (2.3), $\mathbf{x}_q$ denotes the unresolved state

$$\mathbf{x}_q = \frac{1}{2}(\mathbf{x}_1+\mathbf{x}_2) \tag{3.4}$$

and $P_r(k, s, \mathbf{x})$ is the probability that objects 1 and 2 are resolved at frame k when observed by sensor s. We estimate this probability by transforming $\mathbf{x}_1$ and $\mathbf{x}_2$ into azimuth and elevation angles using the position and orientation of each observing sensor, computing the distance between those angles as d_{12}, and finally applying (3.1). The one-target assumption LLR for the ML-PDA-M is the same as (2.1). The probability density function $p_\rho(\cdot)$ refers to the inflated covariance (by the factor ρ) when targets are unresolved.

Similarly, the object resolution probability model can be used to extend the LLR in (2.7) to form the two-target assumption LLR for the ML-PMHT-M (M for merged) as follows:

$$\begin{aligned}\Lambda'_{\text{mlpmht}}(\mathbf{x}; \mathbf{Z}) = \ln \Bigg\{ \prod_{k=1}^{N_w} \prod_{s=1}^{s_n} \Bigg[\Big(1 - P_r(k, s, \mathbf{x})\Big) \prod_{j=1}^{m(k,s)} (\pi_0(k, s) \\ + V \pi_q(k, s) p_\rho[z_j(k, s)|\mathbf{x}_q]) + P_r(k, s, \mathbf{x}) \prod_{j=1}^{m(k,s)} (\pi_0(k, s) \\ + V(\pi_1(k, s) p[z_j(k, s)|\mathbf{x}_1] + \pi_2(k, s) p[z_j(k, s)|\mathbf{x}_2])] \Bigg\} \end{aligned} \tag{3.5}$$

The one-target assumption LLR for the ML-PMHT-M is the same as (2.4). As in (3.3), the probability density function $p_\rho(\cdot)$ refers to the inflated covariance (by the factor ρ) when targets are unresolved.

3.3. Determining the Number of Tracks

Any maximum likelihood algorithm can fall prey to over-fitting, and this is especially acute when a choice must be made between models of differing order. To be specific, a model of higher complexity – for example, one that attempts to fit the data to a two-target model

versus one with only a single target – is mathematically bound to find a maximized likelihood that is no smaller. It is typical to define a "penalty" on model order complexity, and several exist.

A minimum description length (MDL) penalty term was applied to the LLR's in (2.1), (2.2), (2.4), (2.7), (3.3), and (3.5). The resulting LLR's are defined as follows:

$$\Lambda_{MDL}(\mathbf{x};\mathbf{Z}) = \Lambda_{\text{mlpda,mlpmht}}(\mathbf{x};\mathbf{Z}) - \frac{p}{2}\log M \tag{3.6}$$

$$\Lambda'_{MDL}(\mathbf{x};\mathbf{Z}) = \Lambda'_{\text{mlpda,mlpmht}}(\mathbf{x};\mathbf{Z}) - \frac{p}{2}\log M \tag{3.7}$$

where $\Lambda_{MDL}(\mathbf{x};\mathbf{Z})$ denotes the LLR for either ML-PDA or ML-PMHT, $\Lambda'_{MDL}(\mathbf{x};\mathbf{Z})$ denotes the LLR for either ML-PDA-M or ML-PMHT-M, p is the total number of parameters being estimated, equivalent to the number of state variables being estimated multiplied by the number of moving objects assumed to exist, and M is the number of measurements in a batch assumed to be associated with a target, where each azimuth and elevation angle pair is considered two measurements.

$$M = \begin{cases} 2N_w \text{ when } p = 6 \\ \\ 4N_w \text{ when } p = 12 \end{cases} \tag{3.8}$$

4. Simulation Setup

4.1. Notional Scenario

For the purpose of testing the performance of the original algorithms and modified algorithms against that of each other, a notional scenario was created using the Systems Tool Kit (STK11) **stk11** The scenario consisted of two moving objects that launched from the same location but had different destinations. Two orbiting passive sensors traveling in lower earth orbit (LEO) at different angles and heights observed the moving objects. Table 4.1 lists the altitude, inclination, and the right ascension of the ascending node (RAAN) of the sensors. A view of the scenario is shown in Figure 4.1. Coordinates in the earth-centered inertial (ECI) frame J2000 were exported from STK11 for both sensors and the moving objects, as well as the yaw, pitch, and roll information for the sensors.

Table 4.1: Specifications for the space-based sensors.

Sensor	Altitude (km)	Inclination (deg)	RAAN (deg)
1	800	80	145
2	700	90	150

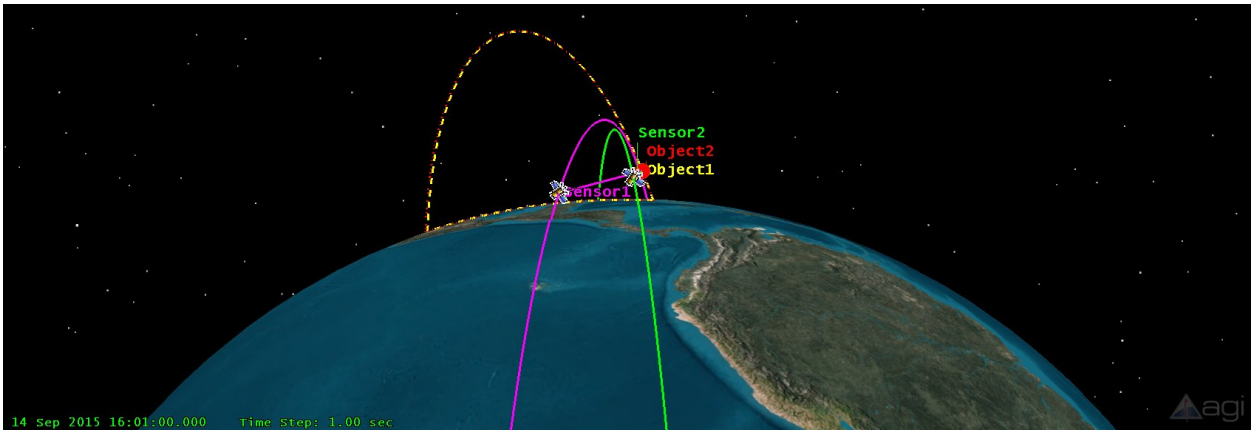


Figure 4.1: The notional simulation setup.

4.2. State Space Models

The position of each of the sensors in 3D Cartesian coordinates is defined by $\boldsymbol{\xi}_s(k)$ where $s = 1, 2$ is used to denote the sensor

$$\boldsymbol{\xi}_s(k) = [\xi_s(k), \eta_s(k), \zeta_s(k)]' \tag{4.1}$$

and the orientation is defined by $\boldsymbol{\omega}_s(k)$ where $\phi_s(k)$ is the yaw, $\rho_s(k)$ is the pitch, and $\psi_s(k)$ is the roll.

$$\boldsymbol{\omega}_s(k) = [\phi_s(k), \rho_s(k), \psi_s(k)]' \tag{4.2}$$

The state of moving object n at frame k is defined in terms of position and velocity in 3D Cartesian coordinates.

$$\mathbf{x}_n(k) = \begin{bmatrix} x_n(k) & y_n(k) & z_n(k) & \dot{x}_n(k) & \dot{y}_n(k) & \dot{z}_n(k) \end{bmatrix}' \tag{4.3}$$

The motion of each moving object is expressed by the following state space model

$$\mathbf{x}_n(k) = \mathbf{F}\mathbf{x}_n(k-1) - \mathbf{G}g\frac{\mathbf{x}_n^p(k-1)}{\|\mathbf{x}_n^p(k-1)\|} \tag{4.4}$$

where the state transition matrix $\mathbf{F}$ is

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

and the input gain matrix $\mathbf{G}$ is

$$\mathbf{G} = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^2}{2} \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \tag{4.6}$$

and the acceleration due to gravity is assumed constant. The position-only coordinates of object n are defined as

$$\mathbf{x}_n^p(k) = \begin{bmatrix} x_n(k) & y_n(k) & z_n(k) \end{bmatrix}' \tag{4.7}$$

The measurement model is defined by the azimuth and elevation angles, $\alpha_n^s(k)$ and $\epsilon_n^s(k)$ measured from sensor s to object n.

$$\mathbf{z}_n^s(k) = \begin{bmatrix} \alpha_n^s(k) \\ \epsilon_n^s(k) \end{bmatrix} = \mathbf{h}(\mathbf{x}_n^p(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k)) + \mathbf{w}_s(k) \tag{4.8}$$

Measurement noise $\mathbf{w}_s(k)$ is zero-mean and Gaussian with variance $\mathbf{R}_s$.

$$\mathbf{R}_s = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix} \tag{4.9}$$

We defined σ_w as half the sensor pixel width. When the moving objects are unresolved, we assume that $\mathbf{w}_s(k)$ has covariance $\mathbf{R}_q = \rho \mathbf{R}_s$, where $\mathbf{R}_s$ is the covariance of the resolved measurements and ρ is the multiplier of the variances of the resolved measurements.

Using the locations of the sensors and the moving objects in ECI coordinates and the orientation of the sensors, the azimuth and the elevation angles measured by the sensors can be computed for the purpose of simulating measurements. First, the position of each moving object n in ECI coordinates is mapped to the appropriate position in the reference frame of each sensor s as follows:

$$\mathbf{x}_n^s(k) = T_s(\boldsymbol{\omega}_s(k))(\mathbf{x}_n^p(k) - \boldsymbol{\xi}_s(k)) = \begin{bmatrix} x_n^s(k) & y_n^s(k) & z_n^s(k) \end{bmatrix}' \tag{4.10}$$

where the rotation matrix T_s is defined by the rotations around the z, y, and x axes

$$
\begin{aligned}
T_s(\psi_s, \rho_s, \phi_s) &= T_z(\psi_s) T_y(\rho_s) T_x(\phi_s) \\
&= \begin{bmatrix} \cos\psi_s & \sin\psi_s & 0 \\ -\sin\psi_s & \cos\psi_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\rho_s & 0 & -\sin\rho_s \\ 0 & 1 & 0 \\ \sin\rho_s & 0 & \cos\rho_s \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_s & \sin\phi_s \\ 0 & -\sin\phi_s & \cos\phi_s \end{bmatrix}
\end{aligned} \tag{4.11}
$$

Then, the azimuth and elevation angles are computed using the position of the moving objects in each sensor's reference frame.

$$
\begin{bmatrix} \alpha_n^s(k) \\ \epsilon_n^s(k) \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{x_n^s(k))}{z_n^s(k))} \\ \tan^{-1} \frac{y_n^s(k))}{\sqrt{x_n^s(k)^2 + z_n^s(k)^2}} \end{bmatrix} \tag{4.12}
$$

4.3. Generating the Measurements

Using (4.10)–(4.12) and (4.8), simulated measurements were generated. The detection width DW, which we considered to be twice the pixel width as defined by the sensor's field of view, was determined by defining the field of view to be a square of width 0.2 rad, or 512 pixels. The probability that the objects were resolved was computed based upon truth (azimuth and elevation angles without noise). If the objects were unresolved with a probability between 0 and 1, a uniform random number was drawn, and if the resulting number was less than the probability of resolution, the targets were assumed resolved. Otherwise, they were assumed unresolved. The average of the azimuth and elevation angles of the moving objects considered

unresolved was computed to simulate a single measurement and then noise was added with variance $3\mathbf{R}_s$. Resolved measurements were simulated by adding noise with variance $\mathbf{R}_s$ to the true azimuth and elevation angles.

Clutter was generated uniformly in the field of view of each sensor, which we chose to be an area of 0.04 rad^2. The field of view was considered to be centered around the moving objects. The number of clutter measurements was determined by generating a Poisson random variable with $\lambda = 4$ for each sensor frame.

Table 4.2: Parameters used to generate results.

Parameter	Value
DW	7.8x10^{-4} rad
σ_w	2.0x10^{-4} rad
P_d	0.7 or 0.9
V	0.04 rad^2
s_n	2 (sensors)
N_w	10 or 20 (frames)
T	1 s
λ	4/frame/sensor
ρ	1 or 3

4.4. Algorithm Parameters

The parameters used are given in Table 4.2. The parameter ρ refers to the inflation of the *assumed* measurement error covariance when two targets end up unresolved from the perspective of the sensor. The true inflation is 3; but the algorithms can assume different values of ρ.

4.5. Maximizing the Likelihoods

In order to maximize the LLR's given above, we minimized the negative LLR using the locally-biased form of the DIviding RECTangles algorithm for global optimization as explained in **jones** and **gab** with a maximum runtime of 20 s. The upper and lower bounds were defined by the initialized state with ±25 km for the position components and ±1 km/s for the velocity components to obtain about half of the results in this thesis. The upper and lower bounds were defined by the initialized state with ±40 km for the position components to obtain the results in Figures 5.7 – 5.9. The optimization was performed in Matlab using code from the NLopt library **nlopt** We then subsequently applied the Matlab command `fmincon` for fine search with the default settings for local optimization and used the same bounds as for the global optimization. The Matlab Parallel Computing Toolbox was utilized to decrease overall computation time for considering several starting frames in each of 100 Monte Carlo (MC) runs.

5. Results

5.1. For Purposes of Illustration: No Noise, No Clutter

First, we considered the highly idealized scenario in which there was no noise, and no clutter was present. The algorithms were initialized on truth and P_d was assumed to be 1. As shown in Figures 5.1 & 5.2 as the measurement error covariance inflation factor ρ was assumed to be greater than 2, the ML-PMHT-M's estimated tracks were "pushed apart" even though the objects were actually unresolved from the point of view of the sensor. This is interesting, and deserves some explanation. Why do the algorithms get a wrong answer even in this most clairvoyant case?

We notionally explore (3.5) for the case of a *single measurement* z and no clutter ($\pi_0 = 0$).

$$\begin{aligned} f(z|\mathbf{x}_1, \mathbf{x}_2) = P_r(k,s,\mathbf{x})\frac{1}{2}\mathcal{N}(z;\mathbf{x}_1,\mathbf{R}_s) + P_r(k,s,\mathbf{x})\frac{1}{2}\mathcal{N}(z;\mathbf{x}_2,\mathbf{R}_s) \\ + (1 - P_r(k,s,\mathbf{x}))\mathcal{N}\left(z;\frac{\mathbf{x}_1+\mathbf{x}_2}{2},\rho\mathbf{R}_s\right) \end{aligned} \tag{5.1}$$

Suppose we compare (5.1) in the two cases $\mathbf{x}_1 = \mathbf{x}_2$ versus $\mathbf{x}_2$ somewhere far away; and under

the (notional!) assumption that $z = \mathbf{x}_1$. We have

$$f(z; \mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2} \frac{1}{\sqrt{|2\pi \mathbf{R}_s|}} \tag{5.2}$$

in the case $\mathbf{x}_2$ is "far away" from the truth (and hence the targets are resolved) and is undetected, versus

$$f(z; \mathbf{x}_1, \mathbf{x}_1) = f(z; \mathbf{x}_2, \mathbf{x}_2) = \frac{1}{\sqrt{|2\pi \rho \mathbf{R}_s|}} \tag{5.3}$$

when the targets are perfectly unresolved. Bringing ρ out of the determinant and square root in (5.3), we see that when $\rho^{n_{dim}/2} \geq 2$, the expression in (5.2) will be of value equal to or greater than that in (5.3). Therefore, in our FPA case, when ρ is assumed to be greater than 2, even in this completely ideal case the ML-PMHT-M will tend to favor the resolved scenario and yield a result in which the estimated tracks are "pushed apart" – one track that is perfectly aligned with the measurement and another that is undetected and far away from truth actually results in a more favorable likelihood situation than the intuitively more appealing outcome of two unresolved tracks. The moral is that the parameter ρ must be chosen with care.

5.2. Examples of Tracks

Figure 5.3 shows the combined estimated tracks from a batch with starting frame 85 and a batch with starting frame 105 from a single run. Most of the frames in these batches had three or more measurements. Each figure is shown from the point of view of sensor 1 or 2

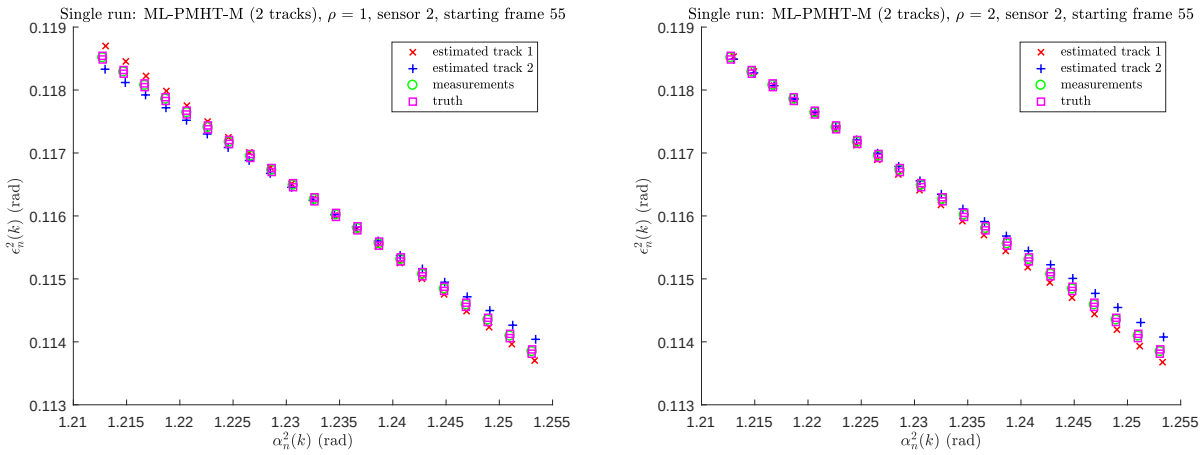


Figure 5.1: Estimated tracks from one sensor's point of view. Left: $\rho = 1$. Right: $\rho = 2$.

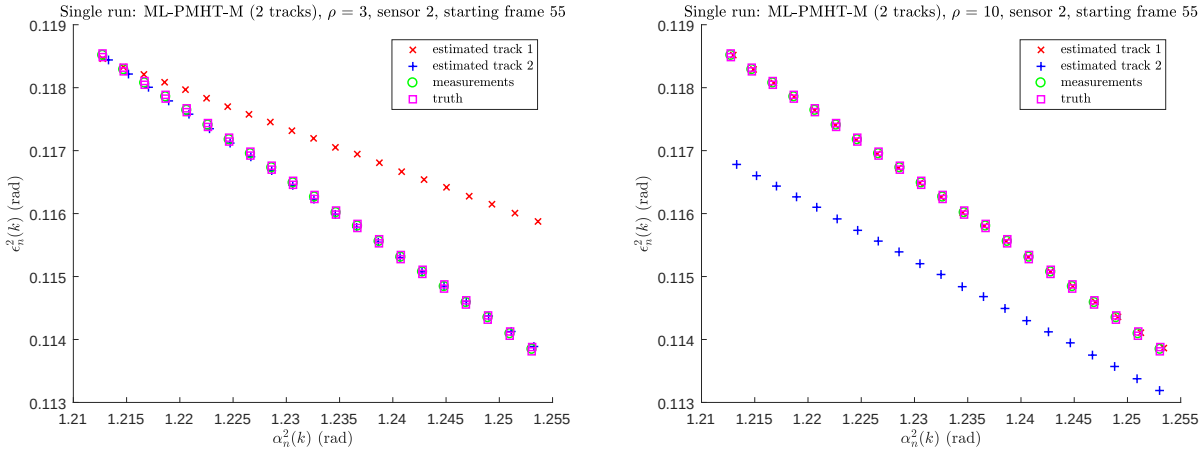


Figure 5.2: Estimated tracks from one sensor's point of view. Left: $\rho = 3$. Right: $\rho = 10$.

and contains the estimated tracks from a single run, the target-based measurements, and truth. These plots show tracks for the ML-PMHT algorithms; those from the ML-PDA are similar.

5.3. Detection of Number of Tracks

Next we considered the more typical scenario in which both noise and clutter are present. Our initial results were generated by considering the data in frames $k = 25 - 184$, during

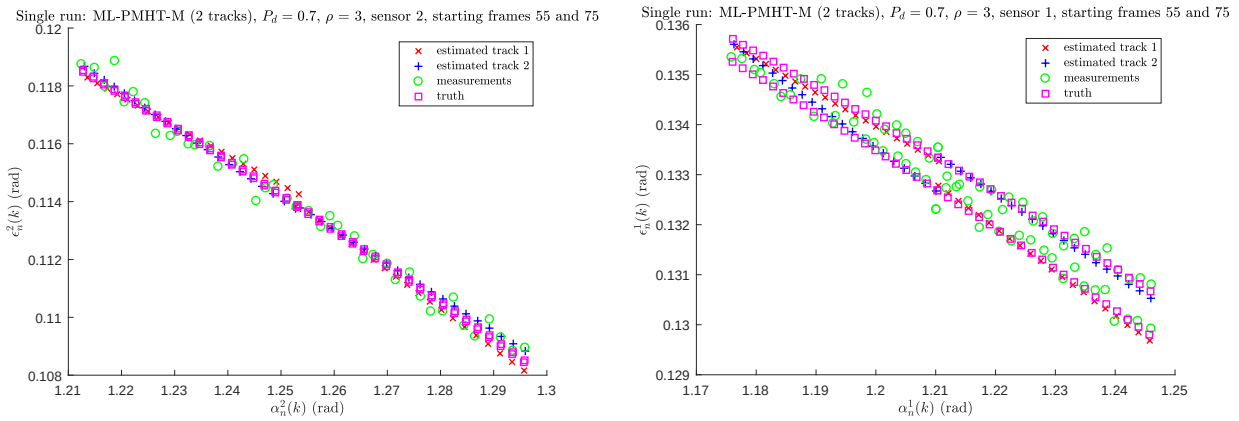


Figure 5.3: Tracks estimated by the ML-PMHT-M (2 tracks) from from a batch with starting frame 85 and a batch with starting frame 105 from sensor 2's point of view for $P_d = 0.7$ and $\rho = 3$. Left: Sensor 1. Right: Sensor 2.

which the objects were unresolved and became resolved. We compared the algorithms by initializing the LLR's at various frames in the interval $k = 25 - 165$ with positions 10 minutes of latitude north of the true positions (approximately $18 - 20$ km away) with $\dot{x}_n = -3$ km/s, $\dot{y}_n = 3$ km/s, and $\dot{z}_n = 2$ km/s. Two different values of the unresolved measurement covariance were considered, namely those for which $\rho = 1$ and $\rho = 3$.

In Figure 5.4, the average number of detected tracks from 100 Monte Carlo (MC) runs is given for $\rho = 3$ and $P_d = 0.9$ when only one object is present; the corresponding (noisy) truth is also shown. The key observation here is that all algorithms seem to be capable of knowing that only one target is present. The PDA algorithms are slightly more stable.

In Figures 5.5 & 5.6, the average number of detected tracks and corresponding numbers of (true but noisy) measurements is shown for $P_d = 0.7$ and values of $\rho = 3$ and $\rho = 1$, respectively. The number of detected tracks is indicated at the starting frame of the tracks. Two tracks were considered detected if the LLR with the MDL penalty term for the estimator that assumed two targets was greater than that assuming only one – naturally estimation

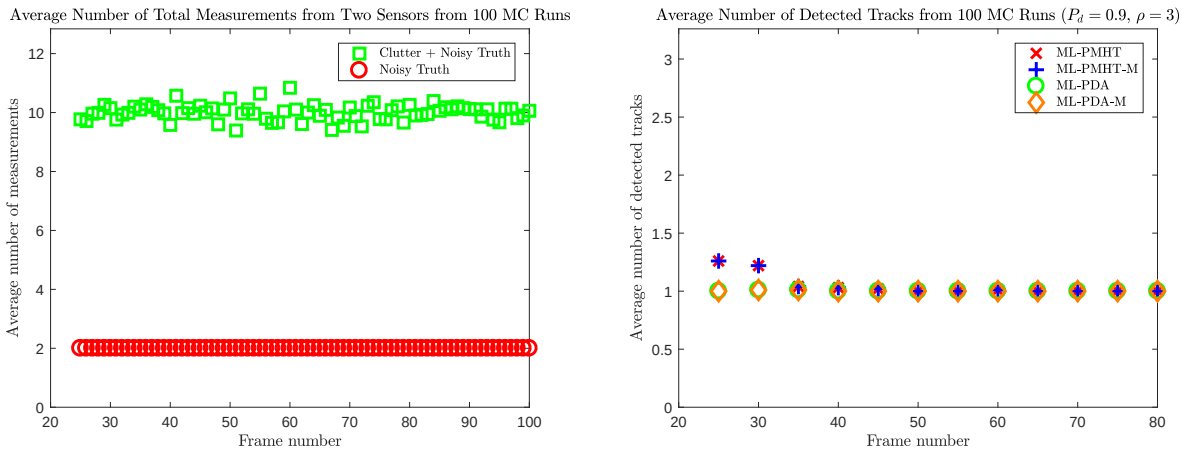


Figure 5.4: Left: Average number of total measurements from two sensors. Right: Average number of detected tracks for $N_w = 20$, $\rho = 3$, $P_d = 0.9$, and 1 object present.

under both regimes has to be performed in both cases. To be specific, (3.6) with $\Lambda(\mathbf{x}; \mathbf{Z})$ equal to (2.1) was compared to (3.6) with $\Lambda(\mathbf{x}; \mathbf{Z})$ equal to (2.2) to make a decision on the number of tracks detected using ML-PDA, and (3.6) with $\Lambda(\mathbf{x}; \mathbf{Z})$ equal to (2.1) was compared to (3.7) with $\Lambda(\mathbf{x}; \mathbf{Z})$ equal to (3.3) to make a decision on the number of tracks detected using ML-PDA-M. Decisions on the number of tracks detected using ML-PMHT and ML-PMHT-M were made similarly.

When ρ was set to 3 so that the modeled unresolved covariance matched the true unresolved covariance, the ML-PDA-M and ML-PMHT-M were able to detect two tracks almost immediately – note that this appears to happen even before the (noisy) number of measurements increases from 2 to 3 both since the algorithms are batch-oriented and because the merged measurements from the two sensors do not "make sense" under the single-target assumption. This result is good since those same algorithms detected one track the majority of the time for the same value of ρ when only one object was present as shown in Figure 5.4. When ρ was set to 1, the ML-PDA and ML-PDA-M algorithms detected two tracks more

quickly and more accurately than the ML-PMHT and ML-PMHT-M. Overall the ML-PDA and ML-PDA-M seem to be more accurate in predicting the number of tracks when fewer than four measurements originating from the objects across two sensors are obtained. However, both of the modified algorithms seem to perform better than their original counterparts with a good selection of ρ to characterize the unresolved measurements as shown in Figure 5.5. The ML-PDA and ML-PDA-M appear to provide more stable estimates of the target number than do the ML-PMHT and ML-PMHT-M.

We then considered the effect of slightly altering the MDL penalty term and the effect of using shorter batch lengths. In considering these scenarios, the optimization position bounds were increased to be the initialized state ± 40 km and additional batches in frames $k = 25 - 165$ were considered. Every fifth frame was the first frame of a batch.

In an attempt to take false alarms into account in the MDL penalty term, we considered the effect of altering the MDL penalty on the likelihood functions in (3.6) and (3.7) such that P_d was multiplied by p. Results were obtained using the modified MDL penalty when $N_w = 20$, $\rho = 1$, and $P_d = 0.7$. The result in Figure 5.7 does not appear much different from the result before the change as shown in Figure 5.6.

Last, we considered what effect changing $N_w = 20$ to $N_w = 10$ would have on the results. Figures 5.8 & 5.9 show that the number of tracks detected with the shorter batch length corresponded better with the number of true measurements. In the first few batches considered (those where the first frame was $k = 25, 30, 35, 40$), the number of estimated tracks was much closer to 1. The increase in the number of tracks detected from 1 to 2

appears to follow closely the increase in the number of true measurements from 2 to 3 both when $\rho = 1$ and when $\rho = 3$.

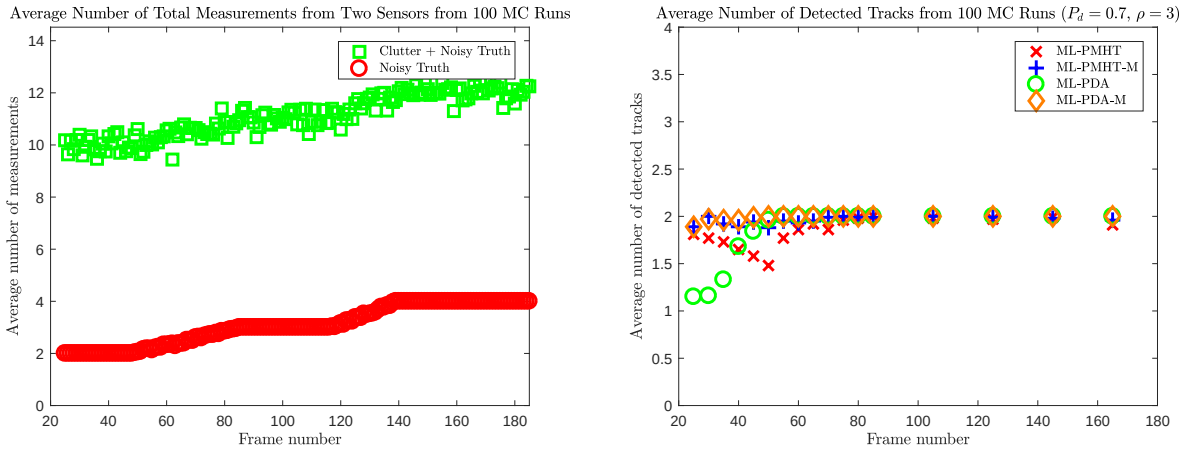


Figure 5.5: Left: Average number of total measurements from two sensors. Right: Average number of detected tracks for $N_w = 20$, $\rho = 3$, $P_d = 0.7$, and 2 objects present.

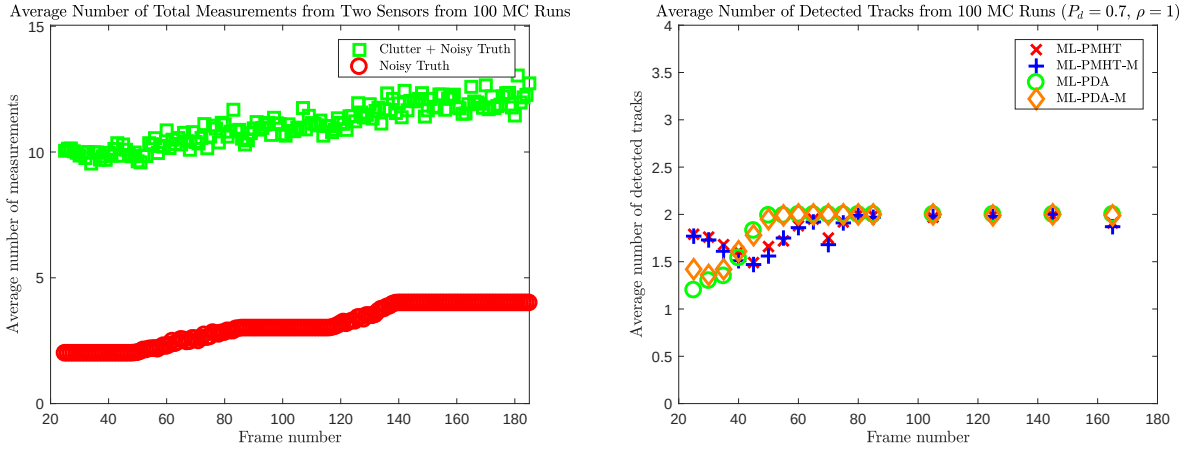


Figure 5.6: Left: Average number of total measurements from two sensors. Right: Average number of detected tracks for $N_w = 20$, $\rho = 1$, $P_d = 0.7$, and 2 objects present.

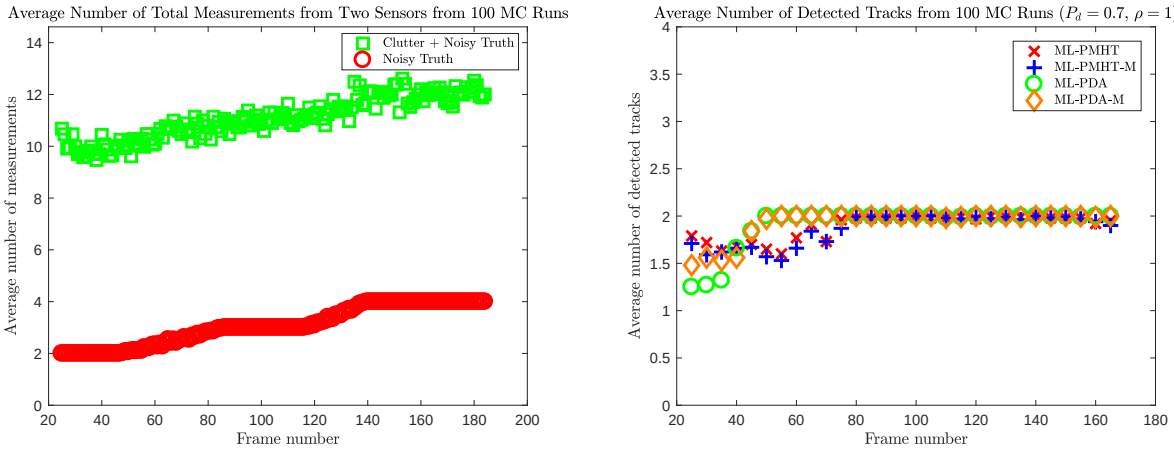


Figure 5.7: Results when P_d is multiplied by p in the MDL penalty term. Left: Average number of total measurements from two sensors. Right: Average number of detected tracks for $N_w = 20$, $\rho = 1$, $P_d = 0.7$, and 2 objects present.

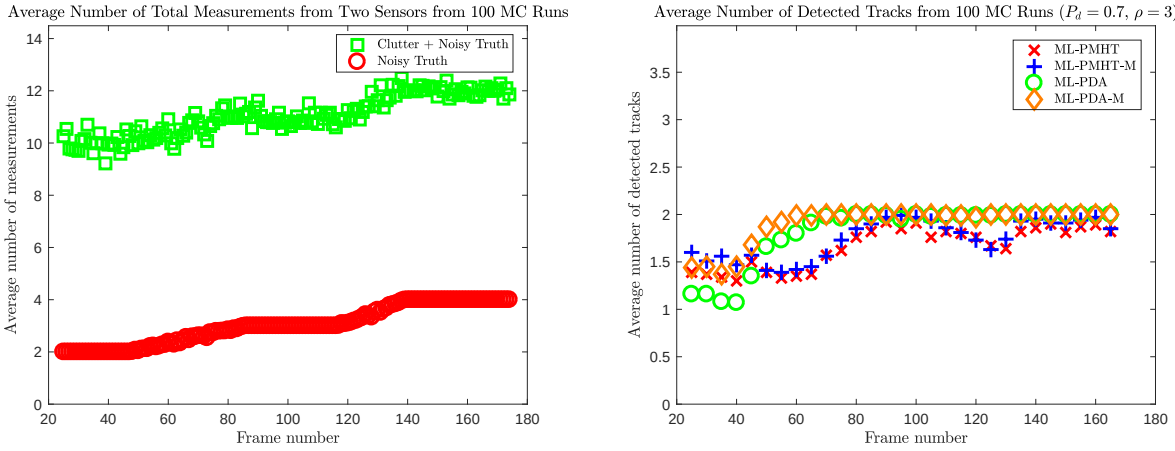


Figure 5.8: Left: Average number of total measurements. Right: Average number of detected tracks for $N_w = 10$, $\rho = 3$, $P_d = 0.7$, and 2 objects present.

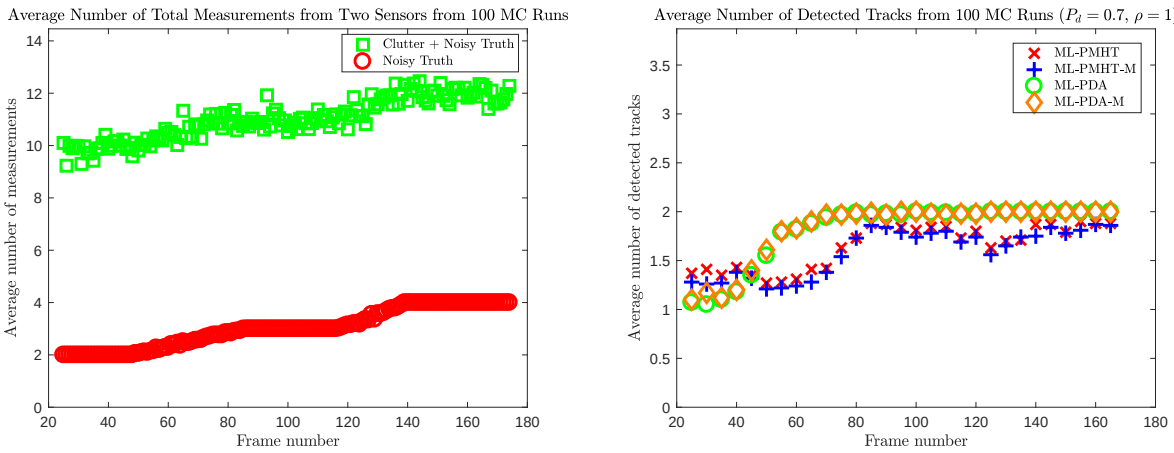


Figure 5.9: Left: Average number of total measurements. Right: Average number of detected tracks for $N_w = 10$, $\rho = 1$, $P_d = 0.7$, and 2 objects present.

5.4. Position Errors

Figure 5.10 gives the average position root median square errors for each track starting frame when $P_d = 0.7$ and $\rho = 3$. The errors are given for the ML-PMHT and for the ML-PMHT-M in one figure and for the ML-PDA and the ML-PDA-M in another figure. To calculate the errors for each algorithm, first the number of detected tracks made by the algorithm in a given run was considered. If that number was 1, the errors were calculated by finding the squared errors between the estimate and each true state. If the number of detected tracks was 2, the squared errors between each estimate and each true state were found. Specifically, the estimated initial position for track 1 was compared to the true initial position for track 1 and the estimated initial position for track 2 was compared to the true initial position for track 2. Then the estimates for track 1 and track 2 were compared to the truth for track 2 and track 1, respectively. The sums of the position errors squared for each of these track

comparisons were compared. The lower of the sums was used to decide which estimated track corresponded to which true track as follows:

$$\min(e_{11}^2(k) + e_{22}^2(k), e_{12}^2(k) + e_{21}^2(k)) \tag{5.4}$$

where $e_{ln}(k)$ denotes the position error between object n and the object associated with the estimated track l ($l = 1, 2$) at frame k. The position of object n at frame k is denoted $\mathbf{x}_n^p(k)$, and the estimated position at frame k of the object associated with track l is denoted $\hat{\mathbf{x}}_l^p(k)$.

$$e_{ln}(k) = \|\hat{\mathbf{x}}_l^p(k) - \mathbf{x}_n^p(k)\|_2 \tag{5.5}$$

If the minimum of (5.4) was equal to the sum on the left of the expression, then the estimated track 1 was determined to be associated with object 1 and the estimated track 2 with object 2. If the minimum was the sum on the right of the expression, then the estimated track 1 was determined to be associated with object 2 and the estimated track 2 with object 1. Then the position and velocity squared errors were computed and recorded for that MC run. This method was repeated until position and velocity squared errors were found for each initial state used and for each MC run. Finally, the root of the median of the squared errors was computed.

While these figures are noisy – reflecting the fact that tracks are sometimes far away from where they should be – the message is that the "M" algorithms provide significantly better estimation performance than do the algorithms that do not consider the merged measurement model.

Next an error analysis was done in which the the Optimal Subpattern Assignment (OSPA) metric with order 2 was used to make decisions on the object and track associations **ospa** In this metric, a function of the error and a cutoff parameter $c_n(k)$ is used and we replace (5.4) with the following expression to determine which track l is associated with which object n:

$$\min\Big(\big(d(e_{11}(k),c_1(k))+d(e_{22}(k),c_2(k))\big),\big(d(e_{21}(k),c_1(k))+d(e_{12}(k),c_2(k))\big)\Big) \tag{5.6}$$

where the function $d(e_{ln}(k),c_n(k))$ is defined as follows:

$$d(e_{ln}(k),c_n(k)) = \begin{cases} e_{ln}^2(k) & e_{ln}(k) < c_n(k) \\ c_n^2(k) & \text{otherwise} \end{cases} \tag{5.7}$$

We chose the cutoff parameter to be equal to four times the longest cross range $r_n^c(k)$ relative to object n and sensor s at frame k.

$$c_n(k) = 4r_n^c(k) \tag{5.8}$$

$$r_n^c(k) = \sigma_w(\max_{s=1,2}\|\boldsymbol{\xi}_s(k)-\mathbf{x}_n^p(k)\|_2) \tag{5.9}$$

The results obtained using the OSPA metric are contained in Figures 5.11 – 5.15. In Figure 5.11, the percentage of tracks in the Monte Carlo analysis in which the ML-PDA-M and ML-PMHT-M estimates were less than the metric $\kappa r_n^c(k)$ where κ is an integer are plotted for each value of κ up to the value for which 95% of the initial estimates of the tracks were within this metric. The maximum number of tracks for the left hand figure is 3400 (100

MC runs with 17 batches per run and 2 tracks per batch); the maximum number of tracks for the right hand figure is 5800 (100 MC runs with 29 batches per run and 2 tracks per batch). The results were obtained using $\rho = 3$ and $P_d = 0.7$. There was greater estimation accuracy when $N_w = 20$ than when $N_w = 10$.

In Figures 5.12 & 5.13, the number of runs for which the estimate at the first frame k of a batch was within the metric when $\kappa = 10$ is shown for the ML-PDA-M and ML-PMHT-M, respectively. In the lefthand figures $N_w = 20$ and in the righthand figures $N_w = 10$. The accuracy of the estimates for object 2 decrease significantly in batches with starting frames later in the data set. This result indicates a problem in the optimization technique. Likely as the objects move farther apart, the initialized state for object 2 becomes close to the true state of object 1 and convergence is easily obtained with the optimization routine for object 1. The initialized state for object 1 is far away from the true state for object 2 and convergence is not easily obtained. A better optimization technique needs to be used with the ML-PDA-M and ML-PMHT-M algorithms to handle this problem and achieve greater accuracy.

In Figure 5.14, we consider the percentage of successfully estimated tracks for each value of κ for $N_w = 10$, $P_d = 0.7$, and $\rho = 1$. The results on the left were obtained by initializing the optimization on states about $18 - 20$ km north of each of the true states whereas the results on the right were obtained from initializing on truth. When initializing on truth, it took a value of $\kappa = 13$ for 95% of the estimates to be within the metric. An ideal value would be $\kappa = 4$ or less. This result confirms that there is a problem with convergence in the

optimization routine because even when initializing on truth the accuracy of the estimates is not desirable.

In Figure 5.15, the number of successful estimates out of 100 MC runs for each batch with starting frame k are shown for $\kappa = 10$. Again, the results on the right were obtained when initializing on truth. The results confirm the problem with convergence in the optimization routine because the estimates of the initial frames of batches considered later in the data set converged well when initializing on truth where as they did not when initializing on states about 18-20 km north of each of the true states.

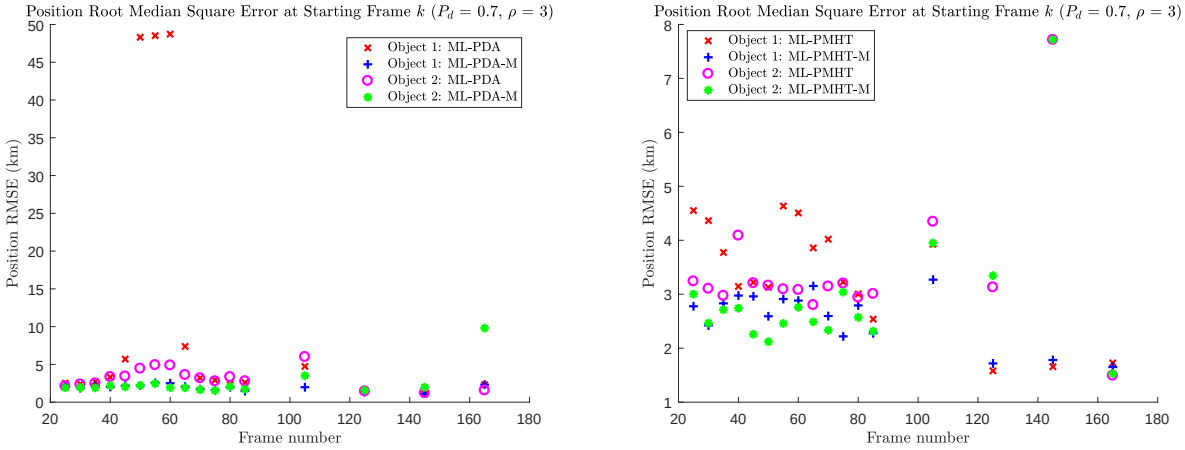


Figure 5.10: Position Root Median Square Errors. Left: ML-PDA and ML-PDA-M. Right: ML-PMHT and ML-PMHT-M ($N_w = 20$, $P_d = 0.7$, $\rho = 3$).

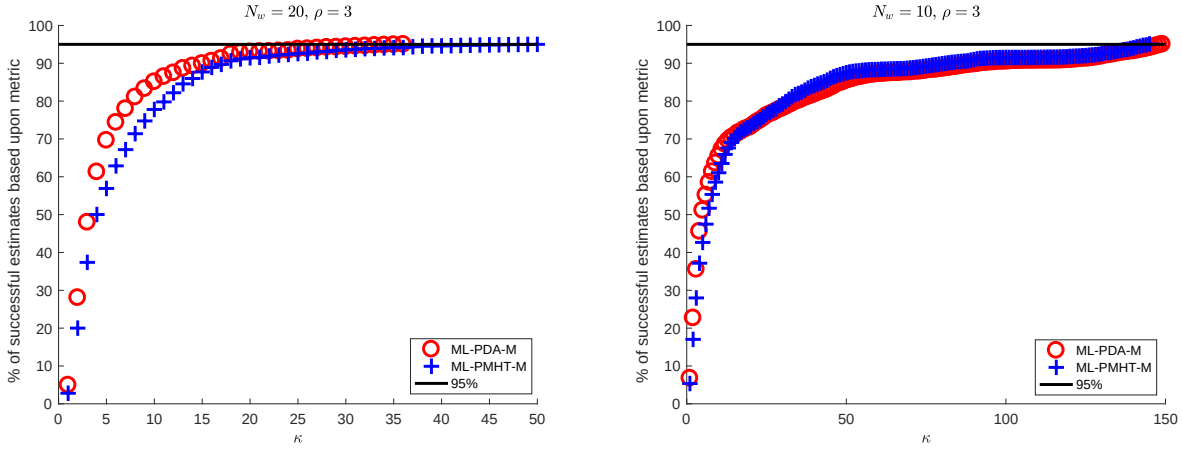


Figure 5.11: Percentage of successfully estimated tracks vs. κ. Left: $N_w = 20$. Right: $N_w = 10$.

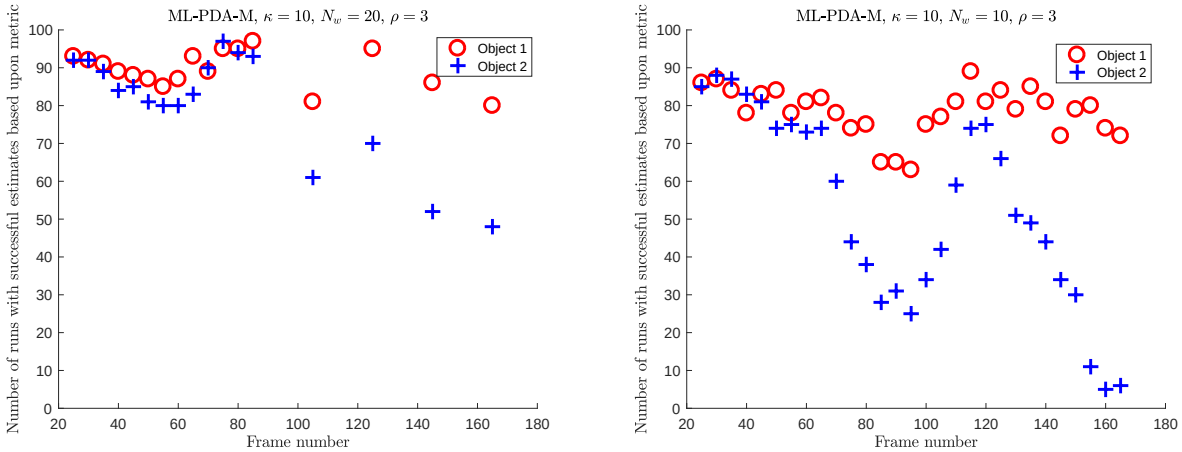


Figure 5.12: Number of runs with successfully estimated tracks when $\kappa = 10$ using ML-PDA-M. Left: $N_w = 20$. Right: $N_w = 10$.

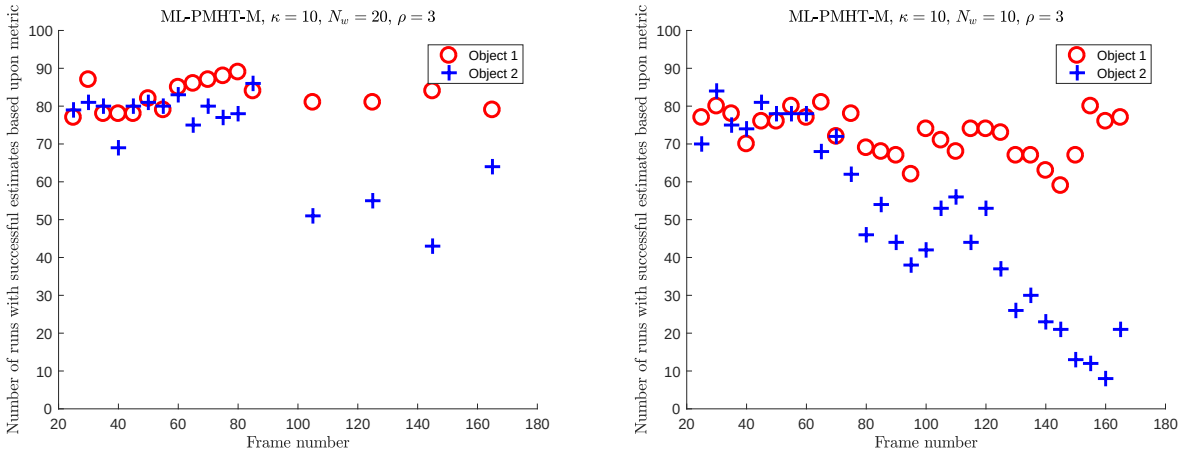


Figure 5.13: Number of runs with successfully estimated tracks $\kappa = 10$ using ML-PMHT-M. Left: $N_w = 20$. Right: $N_w = 10$.

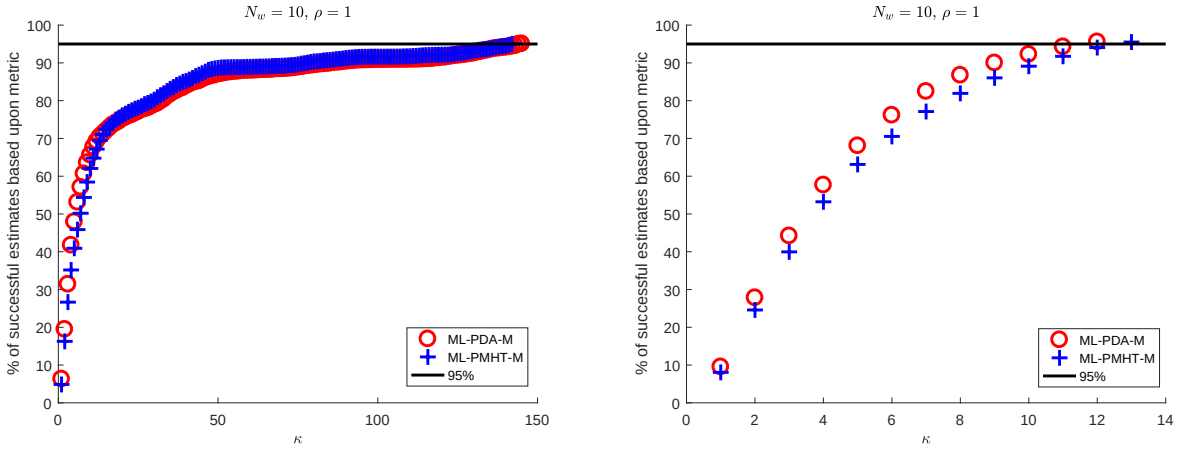


Figure 5.14: Percentage of successfully estimated tracks vs. κ. Left: Initialization on states about 18-20 km north of each true state. Right: Initialization on truth.

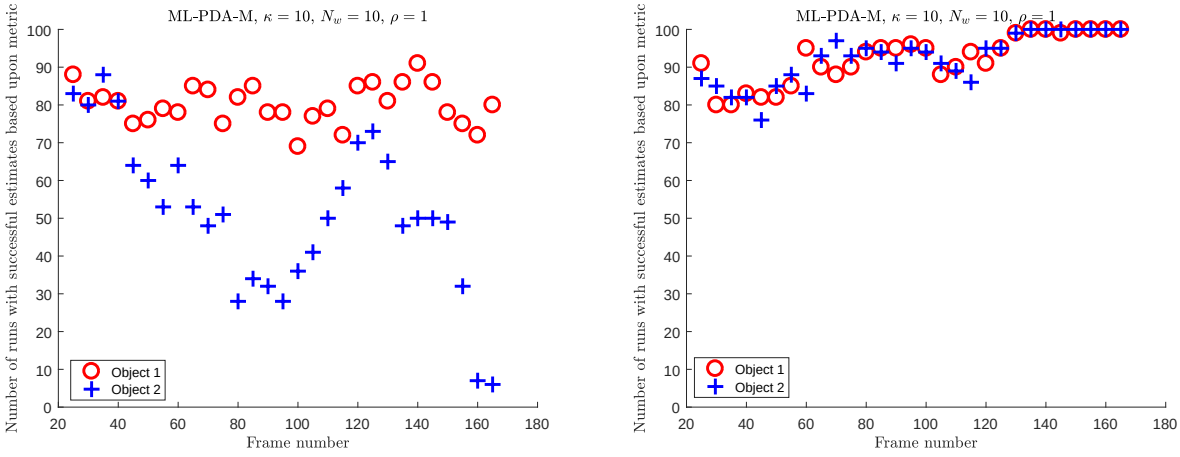


Figure 5.15: Number of runs with successfully estimated tracks when $\kappa = 10$ using ML-PDA-M. Left: Initialization on states about $18 - 20$ km north of each true state. Right: Initialization on truth.

6. Conclusion

We extended the original ML-PDA and ML-PMHT algorithms to handle the case of unresolved measurements by including an estimate of the probability of resolution in the LLR's and denoted these new algorithms ML-PDA and ML-PMHT-M, respectively. The performance of the modified algorithms was compared to that of the original algorithms in a notional space-based scenario in which two moving objects are initially unresolved and are observed by two space-based passive sensors. We draw several conclusions:

- In much previous work (e.g., **schoenecker˙2011**) it had been claimed that the ML-PMHT is preferable to the older ML-PDA since the new ML-PMHT is far simpler and more flexible and since the PDA's more-realistic data association model does not influence the performance when full non-convex optimization must be performed anyway. We stand by that claim in most cases. However, in *this* case of extremely closely-spaced objects and with a need to discern quickly the *number* of objects, the PDA offers advantages.

- Insertion of the modified measurement model to the ML-PDA and ML-PMHT (that is, to the ML-PDA-M and ML-PMHT-M) seems to be advantageous in terms of position error.

- Our model for unresolved measurements includes a term (ρ) that inflates the measurement error covariance when measurements truly *are* unresolved. We have found that incorporation of knowledge of ρ to the estimation algorithms requires care: an injudicious choice can make a wildly inappropriate track a theoretically appealing choice. It appears that setting $\rho = 1$ (that is: ignoring the issue) is a good strategy.

As part of our investigation we have had gratifying success in application of the MDL as a penalty term to determine the number of tracks.

Much future work remains. Foremost, as evidenced by the errors on the estimates when initializing on truth, a better method of optimization needs to be explored and implemented to reduce estimation error. Next is investigation of a better resolution model: the one we have used **chang** is very old and quite approximate. Another key challenge is deeper consideration of the MDL for the measurement-origin uncertainty (MOU) case, since the term M in (3.6) & (3.7) refers to the number of measurements – what should that be when the available measurement suite includes false alarms?