REPORT DOCUMENTATION PAGE		AFRL-SR-AR-TR-03- 0302
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork		
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE**	**3. REPORT TYPE AND DATES COVERED** 15 MAR 01 TO 29 JUN 02
4. TITLE AND SUBTITLE ~~Distributed Data~~ Fusion ~~Research to~~ Support ~~Range Test and Evaluation~~		**5. FUNDING NUMBERS** F49620-01-1-0281
6. AUTHOR(S) Dr. James Llinas		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Calspan UB Research Center Calspan Corporation P.O. Box 400 Buffalo, NY 14225		**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 4015 Wilson Blvd, Room 713 Arlington, VA 22203-1954		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** F49620-01-1-0281
11. SUPPLEMENTARY NOTES		20030818 006
12a. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED		**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 words)*

This report is a part of research conducted at the Center for Multisource Information Fusion (CMIF) at the State University of New York at Buffalo (SUNY at Buffalo) during the second arid third year of a three-year Air Force Office of Scientific Research (AFOSR)-funded research grant to Calspan-UB Research Center, Inc. (CUBRC). The overarching research objective of this grant is to provide understanding about the nature of multi-platform and distributed data fusion and the influence that such methods might have on flight-testing of future multi-platform systems at major range facilities such as, in particular, Edwards Air Force Base, and also with a special focus on Electronic Warfare (EW) aspects and impacts. Ihe effort stems from the visions for future combat depicted in various DoD forward-looking documents such as Joint Vision 2010 (JV2O 10), the Advanccd Battlespace Information System (ABIS), and New World Vistas (NWV), among other similar reports. In those documents, sensibly all views of the future theater environment show a highly distributed but highly connected information environment, with the backbone data-linking infrastructure generally labeled as the "lnfosphere" or "Cybersphere". In essence the backbone for information reconnaissance in future theater environment will be a set of complex distributed data fusion systems consisting of (but not limited to) onboard and off-board fusion systems.

14. SUBJECT TERMS			**15. NUMBER OF PAGES** 130
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**

CMIF

Center for Multisource Information Fusion
State University of New York at Buffalo
421 Bell Hall
Buffalo, NY 14260

CUBRC

CALSPAN—UBRESEARCH CENTER, INC.

FINAL TECHNICAL REPORT

A Frame Work for Performance Evaluation of Multi Target Tracking Systems

Grant No. F49620-01-1-0281

Author:

Sanjay Rawat

Technical Advisors:

Dr. James Llinas

Dr. Christopher Bowman

Dr. Christopher Rump

Prepared for:

Air Force Office of Scientific Research (AFOSR)

and

Air Force Flight Test Center (AFFTC), Edwards AFB, CA

August 2002

Acknowledgements

This report reflects a research effort that would not have been possible without the support of both AFOSR and AFFTC, and that support is gratefully acknowledged herein, especially the programmatic support of Dr. Leonidas Sakell of AFOSR and the technical guidance of Mr. Pete Burke and Mr. Martin Welch. This research was directly dependant on the use of simulator called CASE_ATTI, graciously provided by Defense Research Establishment (Valcartier) of the Canadian Ministry of Defense. We also acknowledge the specific technical support of Mr. Nicolas Duclos-Hindié regarding CASE_ATTI code and operations, and of Dr. Éloi Bossé who authorized the use of CASE_ATTI.

Foreword

This report is a part of research conducted at the Center for Multisource Information Fusion (CMIF) at the State University of New York at Buffalo (SUNY at Buffalo) during the second and third year of a three-year Air Force Office of Scientific Research (AFOSR)-funded research grant to Calspan-UB Research Center, Inc. (CUBRC). The overarching research objective of this grant is to provide understanding about the nature of multi-platform and distributed data fusion and the influence that such methods might have on flight-testing of future multi-platform systems at major range facilities such as, in particular, Edwards Air Force Base, and also with a special focus on Electronic Warfare (EW) aspects and impacts. The effort stems from the visions for future combat depicted in various DoD forward-looking documents such as Joint Vision 2010 (JV2010), the Advanced Battlespace Information System (ABIS), and New World Vistas (NWV), among other similar reports. In those documents, sensibly all views of the future theater environment show a highly distributed but highly connected information environment, with the backbone datalinking infrastructure generally labeled as the "Infosphere" or "Cybersphere". In essence the backbone for information reconnaissance in future theater environment will be a set of complex distributed data fusion systems consisting of (but not limited to) onboard and offboard fusion systems.

As the employment of Data Fusion system increases it becomes pertinent that the systems being employed be dependable in terms of efficiency and accuracy. In this context, rigorously evaluating the performance of the Data Fusion systems is of prime importance. There are several motivating factors for conducting an evaluation of the Data Fusion system. First, the system designer has numerous choices when selecting a particular design concept for a given application. Making an intelligent selection demands the comparison of candidate algorithms, which leads to evaluating them on the same basis without bias. Some examples of systems level issues raised in this process are:

- What is the best combination of sensors to meet a given set of detection probability, Target discrimination, and Target location requirements.

- What level of detection, discrimination, and location performance can be achieved by fusion of a given set of existing sensors? What improvements are accrued by adding sensors or improving the performance of individual sensors?

Second, the evaluation process can also be used to refine the performance of the selected Data Fusion concept, especially the Tracking algorithms. The designer has the mandate to find ways to use the available resources as efficiently as possible. Hence, the evaluation process must allow the measurement of performance criteria for the candidate algorithms, providing quantitative inputs to support their optimization.

Unfortunately, no widely accepted scheme for characterizing the performance of Data Fusion systems / Multi Target Tracking system is currently in use. This document proposes such a formal framework for performance evaluation of such systems in computer simulations.

In this report we present A Formal Frame Work for Performance Evaluation of Multi Target Tracking Systems, which is one of the main focal points of this research effort. In doing so we heavily draw upon the existing (and widely excepted) frameworks, models and techniques of the target tracking community. This has two major advantages – (1) Facilitates reusability of existing s/w and h/w components and (2) Using standard frameworks and norms makes it easier for the tracking community to adopt it – thus giving this aspect of tracking a highly needed jumpstart.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

1.1 Background

Multi-Target Tracking Systems (MTTS) are important components of surveillance systems used mainly by the Department of Defense (DOD). The function of a Multi-Target Tracking System is to employ the measurements from single or multiple sensors about the location and/or kinematic parameters of moving objects present in the space of interest (or objects entering or leaving it), and compute a "best" estimate of their location and related kinematic parameters, in real time. For example, an MTTS monitoring a given air space would be expected to track man-made objects flying within the range of the sensor systems of the MTTS; this study is in fact focused on aircraft type objects. Also the MTTS would usually need to report the latest kinematic information such as position, velocity etc from time to time, depending on the ratio between sensor sampling time and the time dynamics of the observed phenomenology i.e. the rate of change of object location and/or direction. (The reporting time of an MTTS depends on many factors such as system capabilities, mission requirements etc).

1.1.1 Sensors

In the military applications of interest here, these are devices (example Radars, Sonar, Infrared detectors etc) capable of detecting distant objects and measuring their kinematics. But getting high quality measurements from these devices is complicated due to:(a) underlying sensor limitations regarding precision and accuracy, reliability, etc., (b) natural phenomena that complicate the observing process (weather effects, terrain clutter, etc) and (c) in the defense-problems of interest, from the possible use of sensor counter-measures employed (covertly) by an adversary. In short, the output of these devices cannot be accepted at face value and has to be processed in order to bring it within acceptable limits of error. Figure 1.1 depicts a typical problem scenario of interest to this research project. There are multiple, moving/maneuvering (possibly hostile) aircraft being observed at relatively long ranges by a sensor system array comprised of disparate sensor types. Sensor data are processed in an MTTS which incorporates "Data Fusion" techniques that are capable of exploiting the multiple observations on a given aircraft from several of the sensors to produce an improved estimate of true aircraft kinematic behavior (i.e. better than an estimate based on any single sensor data stream).

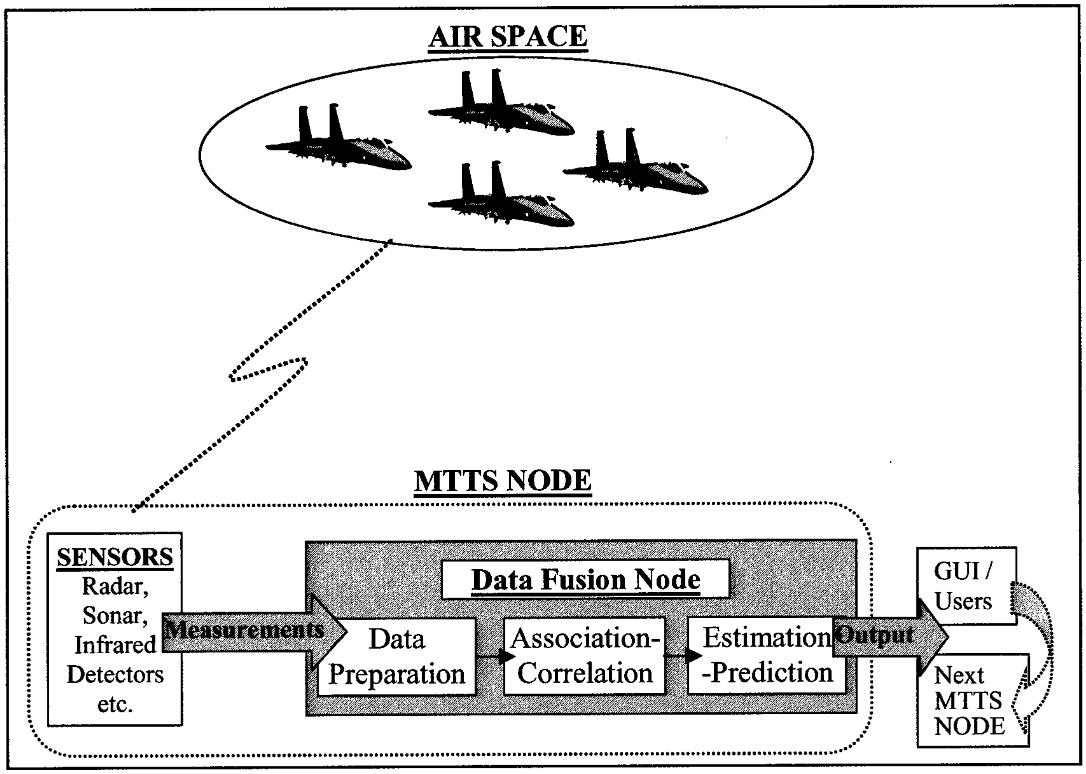


Figure 1.1 Concept of a Multi-Target Tracking System

1.1.2 Data Fusion Node:

The raw data from the sensors is processed to reduce errors and eliminate "noise" (irrelevant data). The output of the data fusion node is an estimate of position of the objects but may also include estimates of velocity, acceleration and other attributes of the objects (depending on the system capabilities and mission requirements). Figure 1.2 shows the conceptual design of a typical Data Fusion "Tree" (data fusion processing is usually partitioned according to various design criteria that leads to a network, or tree-type overall processing architecture).

The main components of Data Fusion Node are:

1. Data Preparation (aka Common Referencing): The data from different sensors is aligned against a common time, position and directional reference. This function may also account for inter-sensor biases and also filter out "obviously" bad data.
2. Data Association: In this step or function, the many measurements are associated with many objects that were previously detected or they are labeled as related to "new objects". This process determines the allocation of the observations to the estimation algorithms that estimate parameters of interest for every object. Different algorithms and systems designs can be used for this process. An in-depth discussion of Data Association techniques is given in the concluding part of this chapter.
3. State Estimation and Prediction: Optimal estimators such as the Kalman filter are used in this step to obtain an optimal estimate of the kinematics of the objects based on all of the observations assigned to the algorithm in the Data Association step. These estimators or "Trackers" are typically recursively-structured optimization algorithms, often based on satisfying a minimum-variance optimization criterion. These estimators also make a prediction of the kinematics of the objects for the next time point. References [1, 2] are suggested as further readings on optimal estimators for interested readers.

The output from an MTTS node is often fed to a succeeding MTTS node, or may possibly be provided to a user via some sort of interface. The user may react to this output and provide his inputs to the next node by modifying or tuning some parameters.

In all Data Fusion (DF) processing applications, the Data Association (DA) step is crucial. This function is usually achieved by applying a 3-step process. The steps of this process involve a "Hypothesis Generation (Hg)" step, a "Hypothesis Evaluation (He)" step, and a "Hypothesis Selection (Hs)" step to produce a "good" allocation of the sensor measurements to the appropriate object-specific optimal estimation algorithms. The problem being addressed is most typically a problem in Combinatoric Optimization, since there are often many measurements and many feasible objects at any time point. The problem we are addressing is that of "Track Maintenance" wherein Track initiation has been accomplished and the Trackers are maintaining optimal kinematic estimates for all initiated Tracks (or objects). Thus the Combinatoric Optimization problem being addressed here is that in which we are comparing measured kinematics with predicted kinematics. Often, to reduce the combinatorics, what are called "gating" techniques have be used either as part of or prior to DA in order to prevent all measurements from having to be compared to all objects, ie to reduce the combinatoric dimensionality. Gating techniques typically employ some type of a priori domain knowledge that, e.g., may suggest that objects cannot move faster than some maximum velocity, in which case object-to-measurement associations, which exceed this criterion, are eliminated from consideration.

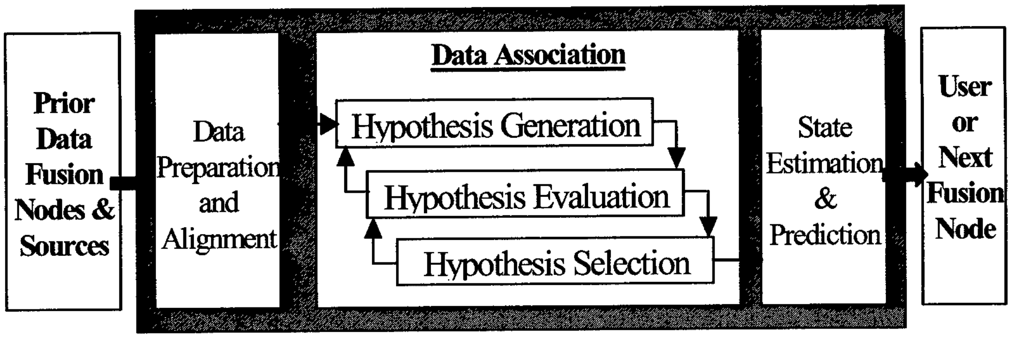


Figure 1.2 Data Fusion Tree Node

1.1.2.1 Data Association

Data association uses models of sensor errors and estimation errors to determine which data should be associated for improved state estimation (i.e., which data belongs together representing the same physical object or collaborative unit--such as for situation awareness). Mathematically, data association is a labeled set covering decision problem. This means that, given a set of prepared sensor data (for example, the designated data shown in Figure 1.3), the problem is to find the best way to collect this data into subsets where each subset contains the data to be used for the next state estimation about a given object. This collection of subsets must cover all the input data and each must be labeled as to which object they are associated to. In Figure 1.3, the red symbols denote the object locations predicted from the so-called Track Filter. The Data Association process, as noted above, takes advantage of prior knowledge of sensor errors and errors in state estimation (typically built into the estimation algorithms; e.g., the Kalman Filter inherently has a built-in process which estimates its own prediction error) to determine which observations are "closest" to the predicted location based on an assessment of the composite errors.

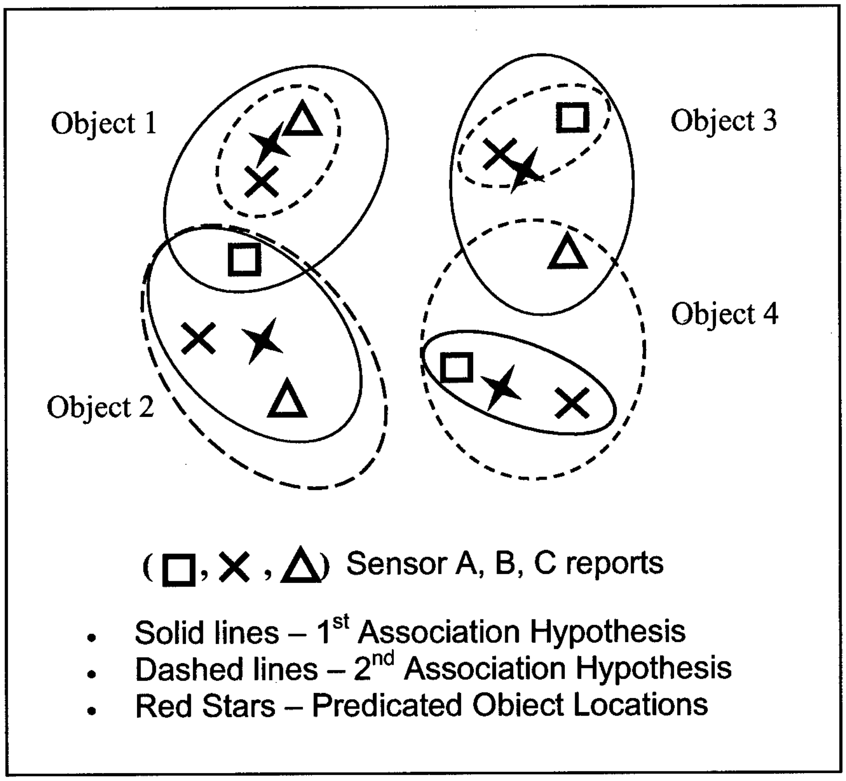


Figure 1.3 Data Association – A labeled set-covering problem

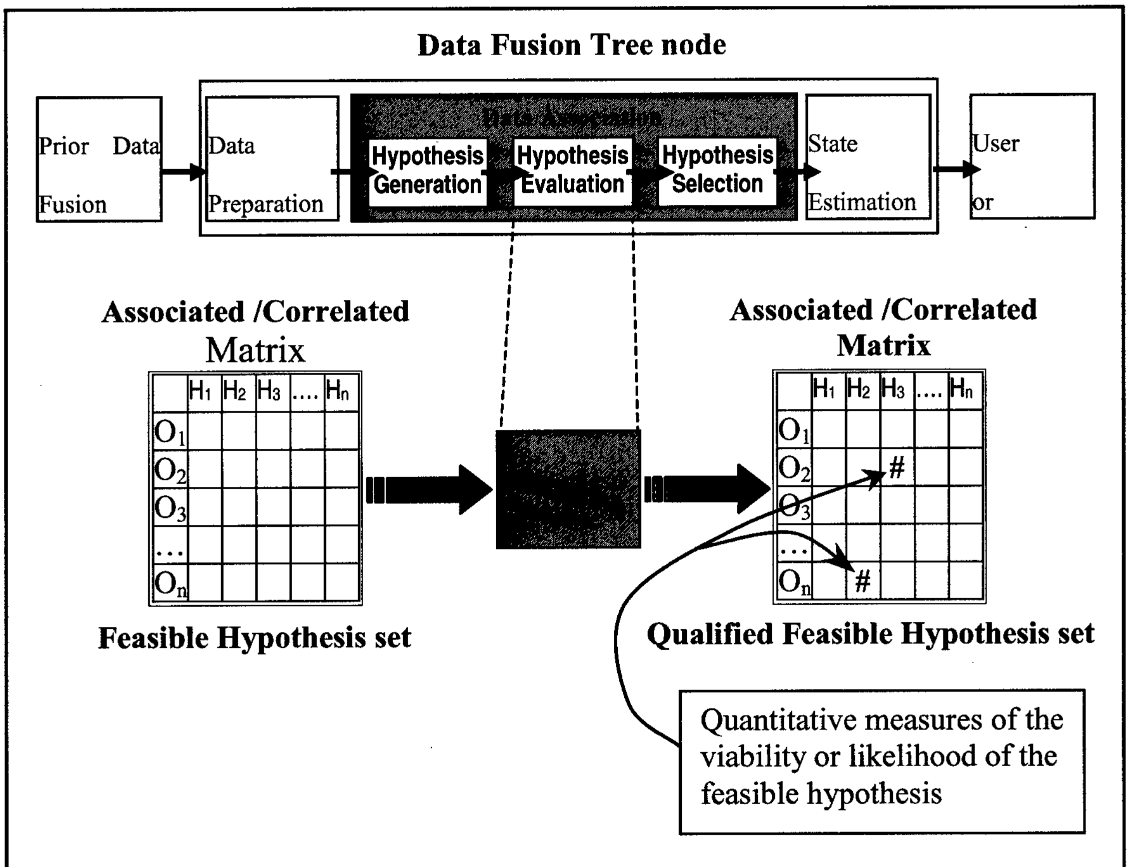
Data Association is segmented into three parts. These are discussed below.

1.1.2.1.1 Hypothesis Generation (HG)

In this step all possible measurement – Track pair hypotheses are formed. As a result of this step, the search space is reduced for the subsequent functions by limiting the feasible data association hypotheses. Further limitations in the feasible hypotheses are achieved by use of the previously mentioned "Gating Techniques" to eliminate infeasible hypotheses. Conceptually, the idea here is to nominate the minimally-sized list of all possible "causes" that could give rise to any expected measurement or observation. This list forms the constrained-set of possible hypotheses to which any observation will be allowed to associate. Examples might be: Object Track, Natural Clutter (e.g. weather causes), hostile deception, etc.; of course, it is hoped that the sensor set employed will primarily yield observations about the Object Track, and be resistant to clutter effects and to hostile deception measures.

1.1.2.1.2 Hypothesis Evaluation (HE)

Even with the employment of an HG function and Gating, ambiguities will arise in the sense of observations being feasibly associated to more than one hypothesis. In order to deal with this in a quantitative way, the feasible hypotheses are evaluated or scored using kinematics, parametric, attribute, ID, and a priori data [3]. The output of the HE function is a quantified evaluation or ranking of the feasible hypotheses, based on a statistically-developed "closeness" metric, as described in relation to Figure 1.3. Thus, the hypothesis evaluation task processes the association matrix, and develops quantitative measures to determine how feasible or likely these explanations are. Many different measures can be used including probabilistic metrics, similarity measures, distance calculations, likelihood functions, and many others. The HE function populates or fills in the association matrix with numerical values representing the feasibility or relative likelihood of the alternative hypotheses explaining the incoming data. The gist of the processing operations in HE is shown in Figure 1.4. As part of this processing, an "Association Matrix" is formed that relates observations ("O") to hypotheses (Hi). The matrix cells contain the computed scores for each possible cell relationship or association. As noted above, in spite of these scores there may still be ambiguity in determining a "best" set of associations; this ambiguity is addressed in the next step.



The Hypothesis Evaluation Function of Association computes quantitative measures to allow comparison amongst feasible hypothesis

Figure 1.4 An overview of Hypothesis Evaluation Function of Association.

A discussion on scoring techniques for Hypothesis Evaluation is out of the scope of this thesis. However the scoring technique adopted for this work will be discussed later.

1.1.2.1.3 Hypothesis Selection (HS)

Once the HG and HE processes have been completed, the overall association process has reached a point where the "most feasible" set of both multi-sensor measurements and Tracks (State estimation process) exists and these measurements remain to be assigned to the appropriate estimation processes that can exploit them for improved computation and prediction of the states of interest. This process is called Hypothesis Selection, where the hypothesis set comprises of all feasible assignment permutations of the inputs to the estimation process. So the input to this process is a two dimensional (can be greater than 2D but usually its two dimensional) matrix (or matrices). The two dimensions being the measurements or observations "O" as noted above, and the Tracks (state estimation processes) and any other feasible hypotheses designated at the HG step, "Hi" in Figure 1.4. The matrix is populated with "costs" of assigning any single measurement to any single estimator. These costs also known as "scores" in Tracking terminology are computed in the Hypothesis Evaluation process. Basically these costs are likelihood of measurement to estimation association and are in fact stochastic in nature. Depending on the HE technique these costs can be integer values or real values.

The usual optimal strategy is to find the set of hypotheses with lowest total cost assignment; depending on the scoring technique it the objective may be to maximize the cost. The cost of assigning any given input to any of a few or several estimators may be reasonable within the thresholds of acceptable costs. If this condition exists across many of the inputs, the identification of the total-lowest-cost assignments may be a complex problem, due to the existence of many alternative patterns within which to allocate the measurements. The central problem to be solved in HS is that of defining a method of selecting the hypothesis-set with minimum total cost out of all feasible hypotheses. This problem is specifically what is called an "Assignment problem" in the domain of combinatorial optimization. The selection of assignment technique for HS is a very critical part of System design. The important issues governing selection of assignment technique for HS are

- Timeliness (Governed by mission goals)
- Optimality desired. (Governed by mission goals)
- Structure of Matrix (2D or N-D. Depends on system design)
- Nature of costs. (Real or Integer values. Depends on HE technique employed)
- Size of Matrix (Depends on the Environment)
- Sparseness (Depends on HG process and Environment)

1.1.2.2 State Estimation

When the HS processing is completed, an optimal assignment of the multiple observations to each "causal" factor or hypothesis will have been completed. For those observations assigned to objects being tracked, the observations are processed by each Tracking filter which is maintaining track on each particular object. That is, the observations are in effect assigned not to an object but to a particular estimation algorithm which is recursively computing (estimating) a best estimate, based on all assigned observations, of the object location and kinematics. The "object" is conceptually instantiated in the system by invoking a Tracking algorithm, which designates its presence. For those observations assigned to other causal factors, such as deception, other algorithms will be employed that exploit the information contained in the observations for other purposes, e.g., to determine the best deception-countermeasure to employ. Our interests here are of course only on the Tracking process, not "other" hypotheses in the system.

1.1.2.2 Data Fusion Node and Process Design

The design of an overall Data Fusion Process for target tracking will possibly involve a series of Data Fusion Nodes, each having processing operations of the type discussed above. As mentioned previously, such processes typically are structured in a network or tree framework, and the overarching strategy to develop the overall design can be imagined to be quite complex, given that each node's operations involve the steps just discussed. Our focus here is to begin exploring structured techniques for evaluating a candidate or prototype Data Fusion-based target tracking system.

1.2 Introduction: The Performance Evaluation Problem

The design and development of algorithmic techniques for estimating the "best" location and related kinematic parameters of moving objects which are observed by single or multiple sensors is a complex process [4]. It is complicated in part by: (1) the difficulty of obtaining high-quality measurements from sensor systems due to underlying sensor limitations regarding precision and accuracy, reliability, etc., from natural phenomena that complicate the observing process (weather effects, terrain clutter, etc), and in the defense-problems of interest, from the possible use of sensor countermeasures employed (covertly) by an adversary. Another complicating factor is: (2) the inaccuracy associated with the estimation algorithm being used. Virtually all estimation algorithms are model-based, and employ a priori models of Target motion, sensor errors, system noises etc in order to estimate the Target kinematics.

However, if the inter-object/target spacing is large enough and other factors that degrade DA are not present, then most current-day Data Association processes will separate the observations into the correct sets, ie into observation-sets correctly aligned with the objects. Said otherwise, the observations will be grouped into sets that are "caused" by the objects[1]. Our concern is for those problem cases where the factors influencing the Data Association process for the --"System Under Test (SUT)" fusion process are such as to result in ambiguous associations for the SUT. Among the factors influencing DA are inter-target separation distance and also sensor sampling rate;

Thus, for many cases of interest, a perfect match between the a priori models and the true behavior of the objects, the sensor observational processes, and the nominated data associations from the DA of the SUT cannot occur due to the stochastic behavior of the elements involved. As a result, there is a divergence in the *estimated (from the SUT) and the real ("truth") picture of the composite multi-object kinematic behavior.* The goal of a

[1] It can be argued that this never happens in an entirely correct way but we are discussing degrees of problem cases in a general way here.

Multi Target Tracking System designer is to develop a fusion-based tracking system that yields a composite, estimated kinematic picture which is in some sense considered a "good enough" estimate of the composite, true object behavior. Hence at various stages in development of a tracking system it is necessary to evaluate the performance of the system in order to see how close the system's estimate is to the true picture. This is the fundamental issue addressed here: given all the components of a typical tracking system (whose design, as a network of separate fusion processing nodes, is often referred to as a "Data Fusion Tree"), along with the overarching stochastic characteristics of the problem, on what basis can an equitable approach to evaluation of a candidate-design tracking system—the "SUT"-- be based?

The process of performance evaluation allows the analysis of the absolute and relative performance of SUT's, through comparisons of the measured and estimated tactical pictures with what has come to be called the "Air Truth" picture. There are several motivating factors for conducting an evaluation of the Tracking System. First, the system designer has numerous choices when selecting a particular design concept for a given application. Making an intelligent selection demands the comparison of candidate algorithms, which leads to evaluating them on the same basis without bias. Some examples of systems level issues raised in this process are [5]:

- What is the best combination of sensors to meet a given set of detection probability, Target discrimination, and Target location requirements.

- What level of detection, discrimination, and location performance can be achieved by fusion of a given set of existing sensors? What improvements are accrued by adding sensors or improving the performance of individual sensors?

Second, the evaluation process can also be used to refine the performance of the selected Tracking System concept, especially the Tracking algorithms. The designer has the mandate to find ways to use the available resources as efficiently as possible. Hence, the evaluation process must allow the measurement of performance criteria for the candidate algorithms, providing quantitative inputs to support their optimization.

Finally, the quantitative performance assessment can be applied to determine the relative contribution of the Tracking System to the success of the mission for which it is being employed. In the context of a Tracking System design problem that starts from scratch, with design choices open for sensors, operating conditions, and algorithms, many factors are involved. Table 1.1 shows some of these factors and the design choices

	Factors Affecting Tracking Algorithm Design	Effects on Tracking System Design
Target Characteristics	• Target Signature (example, Radar Cross Section) • Target Motion (Maneuvers) • Target Scintillation Characteristics	Models are required for each expected Target; thus this requires understanding the Target-set expected.
Scenarios	• Sensor-Target Geometry • Target Density (Many Objects) • Highly Dynamic environment • Target Proximity (Closely spaced Objects)	This defines the sensor-to-Target geometries and range of expected Target kinematic behaviors—this establishes models of detection probability, sensor performance, and complexity of Target kinematic models

Physical Environment	• Clutter • Propagation Effects (Complementary Sensors) • Electronic Countermeasures (Enemy interference) • Weather conditions • Day/night operation	This defines the physics of sensor operating conditions and affects the sensor performance models but also affects the feasible operating conditions for the algorithms being studied, eg viability to operate under different weather conditions and night/day conditions
Measurement Process	• Sensor Power • Sensor Sensitivity • Sensor Detection threshold • Sensor Resolution • Sensor Noise • Sensor Misalignment • Sensor Failures • Limited processing capabilities	Determines what sensor design parameters can be modified by the system designer, and thereby overall sensor performance
Estimation Process	• Bad Assumptions in Target Models • Bad tuning of parameters • Limited processing capabilities	Affects the overall quality of the estimation algorithms and their processing speed

Table 1.1 Factors affecting the performance of Tracking System [6]

affected by each. The focus of this study however is not so broad; we focus here on evaluating the Tracking algorithms per se, given the overall Tracking System design. Our focus in particular is not to conduct the evaluation of these algorithms with algorithm performance improvement in mind but with the general purpose of understanding methodological options in forming a performance evaluation approach since, as we argue below, such basic understanding has not yet been developed in the data fusion community.

1.3 Approaches to Performance Evaluation

There are two distinct approaches to the performance evaluation problem [5], Analytical and Simulation-based.

1.3.1 Framework for Tracker Performance Evaluation

Figure 1.5 shows a depiction of the overall framework for evaluating object-Tracking algorithm performance.

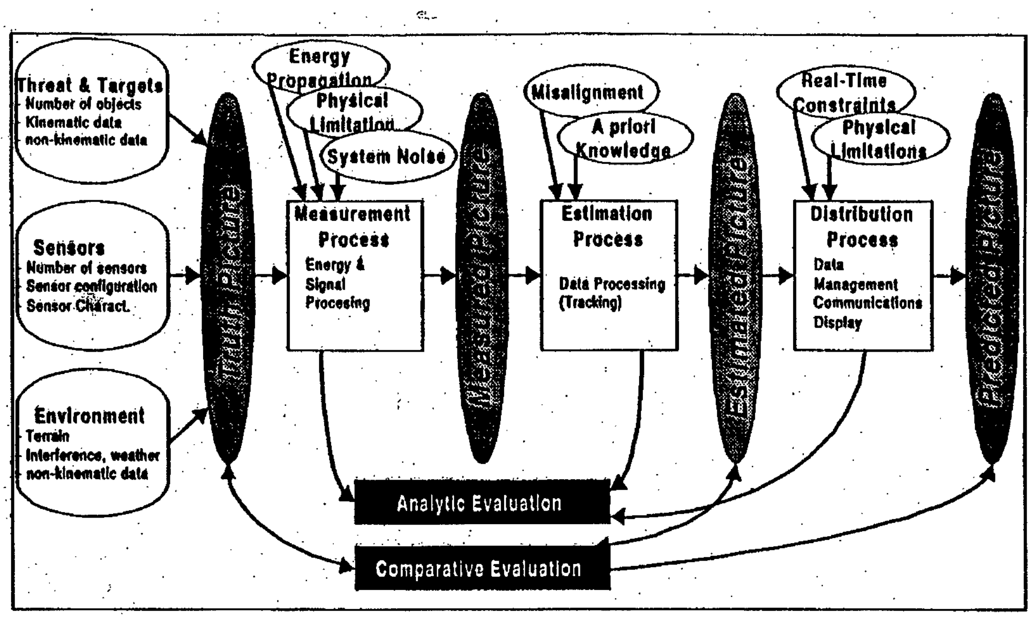


Figure 1.5 Overall Framework for Tracker Performance Evaluation [7]

This Figure, along with Table 1.1, provides a context for understanding the issues, factors, and complexities in defining and carrying out a performance analysis of object-Tracking algorithms in a data fusion-based context, i.e. in situations where the algorithm also depends on a Data Association process, and wherein there are multiple observations for each object. In reference to Figure 1.5, we see that there is the notion of the "Truth Picture", meaning the actuality of object positions over time and the actuality of the sensor data/observations/measurements, and also of the environment. In the cases of interest here, this Truth Picture is known when the analysis is simulation-based but for real-world cases the Truth Picture may itself need to be estimated from observations; this is a separate matter not addressed herein. The Measured Picture is developed from the Measurement Process and the factors, noted in Table 1.1 and in Figure 1.5, which affect that process. In the data fusion case, these multiple measurements need to be correlated to (assigned to) particular estimation algorithms operating on each objects kinematics. Subsequent to association processing, the Tracking algorithms produce estimates of object locations and other kinematic factors over time, producing the Estimated Picture. For algorithms that have a recursive structure, they can also propagate the kinematics estimates forward in time, and this produces the Predicted Picture.

As noted in Figure 1.5, analytically based evaluations typically compare the effectiveness of estimation algorithms in extracting all available information content from the measurements. Simulation-based evaluation methods typically compare the Truth Picture (requiring a composite, "all-Tracks" metric) with either or both of the Estimated Picture and the Predicted Picture.

1.3.2 Analytical Methods

A number of investigations have been carried out on the development of performance prediction (and thus performance analysis) of a Tracking System. For example Sea and Singer [8, 9] analyzed the performance of nearest-neighbor type data association algorithms. They provide a method for predicting Track accuracy as a function of Track history. Blackman [10] derived some analytic expressions for computing the probabilities for correct correlation, false correlation and correct decision. Another example is the use of covariance analysis to evaluate the effects of mismatch between the Kalman filter model and the true Target dynamic model, offering a reduction of the model state space dimensions and the variation of the sampling interval of the sensor.

Analytical evaluations are also helpful in estimating the upper bounds for the performance and to identify the key factors contributing to the Tracking performance [7]. These bounds are typically based on the information content of the data and are not related to any particular estimation technique. These techniques are very useful in performing sensitivity analysis. However these bounds are optimistic and no estimation technique can guarantee to achieve them. Hence to get the real picture Simulation evaluation becomes a necessity, Figure 1.5.

A complexity and limitation of Analytical methods is that the various processes shown in Figure 1.5 are not connected by an integrated set of closed-form mathematics. Thus, the interdependencies among each succeeding processing operation cannot be expressed in a connected set of mathematics; this aspect is further complicated by the stochastic nature of the mathematics. There is also the time-dependency aspect. As a result, Analytical forms are typically applied using assumptions of various types and also framed for the steady-state case, reflecting average performance rather than temporally dependent performance.

1.3.3 Simulation-Based Evaluation

The Performance evaluation through computer simulations refers to the examination of output results of Tracking algorithms running against different scenarios established in a simulation system (i.e. computer-based system) under a set of pre-established measures. The major advantage of this method is that the simulations can be designed to reflect the real application environment as closely as possible and the models employed can be much more complete that those used in the analytical evaluations. A generic model for evaluation through simulations is as shown in Figure 1.6.

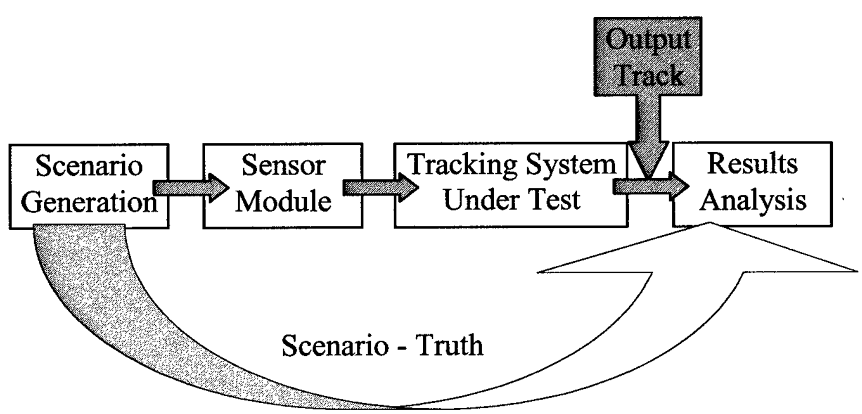


Figure 1.6 General model for Simulation based Testing and Evaluation [5].

1.3.4 Issues And Difficulties In Tracking System Evaluation.

Wolff and Fixen [11] discuss a wide range of difficulties that are encountered in designing a performance evaluation system. These are enumerated below.

1.Lack of Global Measures of Performance

The currently available measures of performance are not applicable to all kinds of algorithms. There is a need for *Global* measures which can be applied to most of the algorithms.

2.Diversity of Aspects Involved

The Tracking System provides answer to many different questions for example location, velocity, attributes, identity etc. It is necessary to know how well does the system performs in each of its regimes. This requires a set of measures of performance that cover all the aspects of performance of these systems.

3.Complexity and Stochastic Behavior

The Target estimates are a blend of random distributions, hard to evaluate because they jump around too much. Even if the same scenario is repeated many times, each repetition may be expected to produce a different set of answers. Hence in order to evaluate at any confidence level one needs to conduct several runs and then take an average.

4.Location Effects

Sensors accuracies are always dependent on range and angle of the Target. Hence while a sensor might be very accurate in one scenario and very in accurate in another scenario. This makes it very hard to establish the accuracy expectation of a Tracking System. Hence it becomes necessary to test the system against a set of scenarios so as to bring out to ascertain its accuracies in different conditions.

5.Target Priorities

Some Targets are more important than others. Trackers spend more time on high priority Targets; consequently their performance against such Targets might be better than against normal priority Targets. This should be taken into account while designing the evaluation system.

6. Model Matching Issues

There are several types of sensor models. In lab simulation its important that the sensor model used by the simulator concurs with the one used in the estimation algorithm other wise the algorithm gets penalized for the mismatch.

7.Model Fidelity Issues

In an ideal Tracking System very high fidelity models would enable very accurate Tracking. But in tactical systems, the Tracker is often called upon to respond to high data rates and time-critical requirements under severed limitations of size, power and weight.

These constraints militate against high fidelity and force the designer to compromise. Hence it's prudent that the tactical algorithms evaluated "at speed".

8.Differences Between Algorithms

Difference between algorithms interferes in comparative evaluations and makes the process very expensive.

9. Performance evaluation Ambiguities

In order to compute measures of performances such as Position estimation error the real position of the object or the Air Truth[1] (hence forth referred to as *Truth*) and the estimated position or the Output Track[2] (hence forth referred to as *Track/s*) have to be compared. An advantage of laboratory simulations is that the Truth is easily available. Knowing the Truth would be enough in Single Target environments. In such a situation there would be just one set of Truth data and one set of Track data for comparison. But in case of Multi Target environments there would be many sets of ground Truth data and almost as many (or more) sets of Track data. In order to compare Track to Truth one needs to know which Track represents which Truth. The process of correlating Truth to the representative Track is termed as '*Track to Truth Association*'.

Track to Truth Association is a complex problem in Multi Target Multi sensor environments; the complexity increases in dense environments of clutter, false alarms, and closely spaced Targets. Often in such environments Tracks get updated from more than one source for example consider a scenario wherein two Targets are maneuvering as shown in Figure 1.7(a)

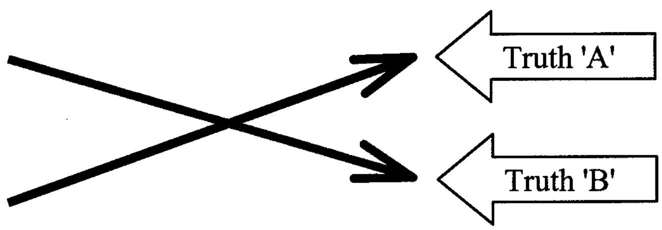


Figure 1.7(a) Two crossing Targets

Figure 1.7(b) shows how the measurement generated from Truth-A gets assigned to Track-1, which had been representing Truth-B till this time point. Similarly the measurement generated from Truth B gets assigned to Track-2 that had been representing Truth-A till this time point. Thus a "Switching" has occurred at this point i.e. that Target/Truth that was earlier being represented by some Track is now being represented by another Track. This kind of Switching is frequently observed in environments where Targets are maneuvering closely. Several factors contribute to Switching; some of which are listed in table 1.1.

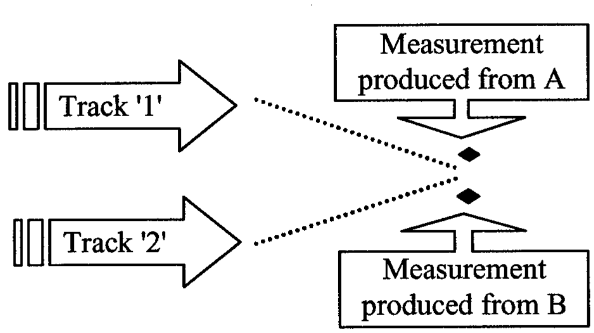


Figure 1.7(b) Origin of Switching

A possible output of the Tracking System for the Targets is shown in Figure 1.7(c). Thus for some part of the scenario Track-1 represented Truth-A and for the other part of the scenario Track-2 represented Truth-A. So the question arises which Track should represent which Target for evaluating the Tracking System.

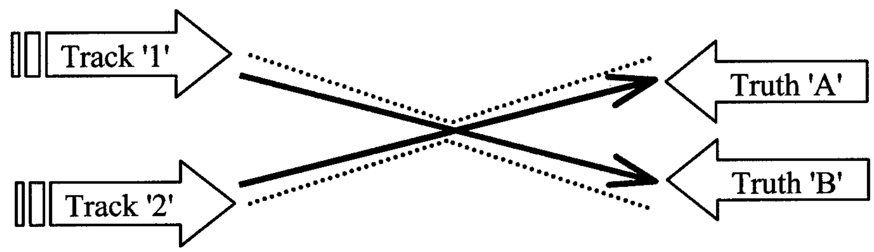


Figure 1.7(c) Concept of Switching

1.4 Methods Of Evaluation

Some of the different methods of simulation evaluation that have been observed in the Tracking community are discussed below.

1.4.1. Drummond's work

Drummond along with others has discussed some of the aspects of Performance evaluation in his various publications [12, 13, 14, 15]. The focus of these publications has been around ambiguities involved in evaluating the performance of Tracking System and metrics for evaluation of performance. Some methods to assigning Track to Truth have been suggested and metrics for evaluating the performance have been presented.

In his work Drummond presents a two-step methodology for performance evaluation of Tracking System. As shown in Figure 1.8 the first step of this methodology is about assigning Track to Truth while the second step deals with computing the performance metrics give the Track to Truth assignment.

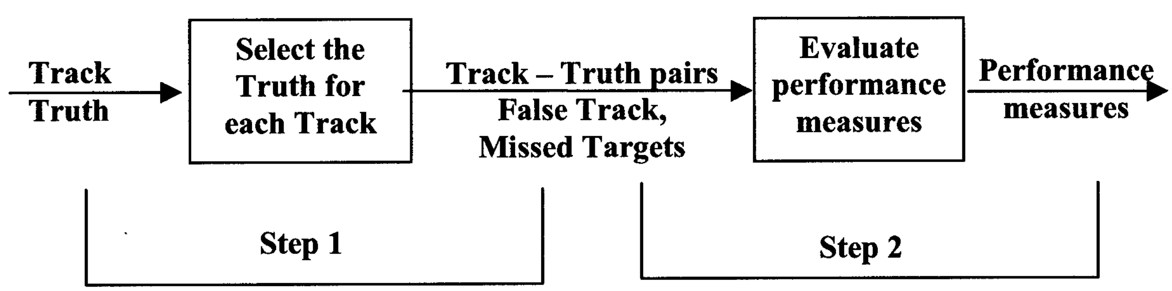


Figure 1.8 The Two steps of the PE Methodology

Two alternatives to computing scores for Track to Truth assignment have been discussed in [14]. The first one is based on using the statistical distance between Track and Target locations. Different functions can be used to measure the distance. For instance, the sum of the squares of the difference between each component of the Track state vector and the Truth state vector is one measure of this distance. Another measure could involve the position components of the Track and Truth state vectors. Yet another one suggested is based on the difference between the estimated state of the Track and the Truth state and weighted by the error covariance matrix of the Track. They also suggest the use of a threshold value on this measure to perform gating.

Another method is based on Track purity. In this method the objective behind the assignment of Track to Truth is to maximize Track purity. Kovachich and Chong first discussed this method [16]. In their 1991 paper [13] Drummond and Fridling discuss this method but prefer the distance based scoring method to it.

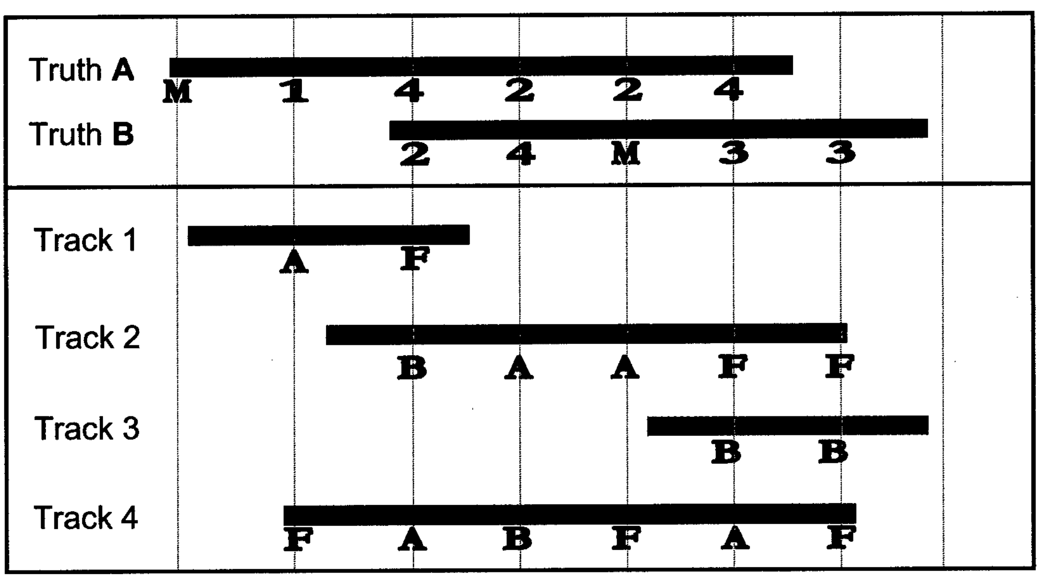
Overall preference has been given to the distance based method because it does not require the evaluator to know anything about the measurements and their source nor about the Tracking algorithms.

Four different methods for dealing with ambiguities involved in assigning Track to Truth have been presented. These methods have been distinguished on the basis of Track-Truth Association "strategy". These strategies dictate whether *Track Switching* is allowed or not. From the perspective of evaluation, Track Switching refers to changes in the assignment of Track to Truth over time.

In these methods Tracks are either declared as valid in which case they are assigned to some Truth or else they are declared as invalid or False Track. Truth, which is not assigned to any Track, is declared as Missed Truth. Depending on the method a Track may be assigned to more than one Truth during it life time but it is never assigned to more that one Truth at a time i.e. unique assignment of Track and Truth is done.

a. Method 1: The Switching Strategy

In this method there is no limit on Track Switching. The assignment of Track to Truth is computed at time points of interest independent of the assignments at other time points. As result a Truth can be associated with some Track at a time point and the same Truth can get associated to a different Track at the very next time point. Figure 1.9 depicts the assignment of Truth to Tracks at various time points of the scenario. For example Truth 'A' was declared as missed for the first time point. It was then assigned to Track '1' at the next time point i.e. to say Track '1' represented Truth 'A' at that time point. At the very next time point Track '4' represented Truth 'A'; resulting in a *Track Switch.*

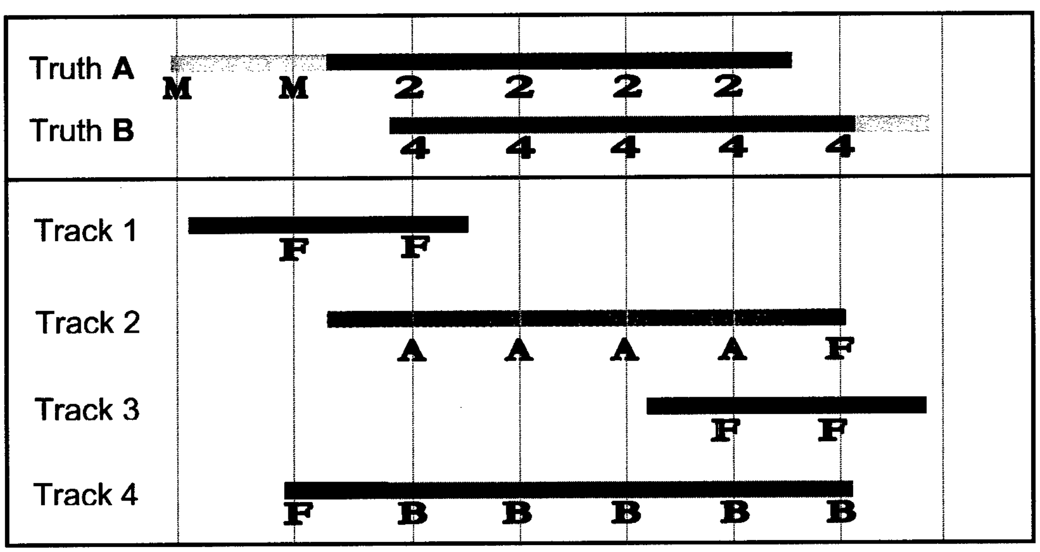


MISSED TRUTH - **M** FALSE TRACK -**F** ASSIGNED TO TRACK 1-**1**

Figure 1.9 Hypothetical assignments of Truth to Track using Switching Strategy

b. Method 2: The No Switch Strategy

This method does not allow Track Switching within a run; however a Truth may be associated with a different Track for different Monte Carlo runs. Track to Truth assignment is done only once for a single run. Scores are calculated at time points of interest for all candidate Track-Truth pairs. Then sum is taken over the time from the earliest occurrence of the Track or Truth until the latest occurrence of the Track or Truth. Thus there is a penalty imposed for Missed Truth or False Track. Figure 1.10 shows a typical Track-Truth association using No Switch Strategy. Here Truth 'A' was represented by Track '2' for all time points when Track '2' existed.



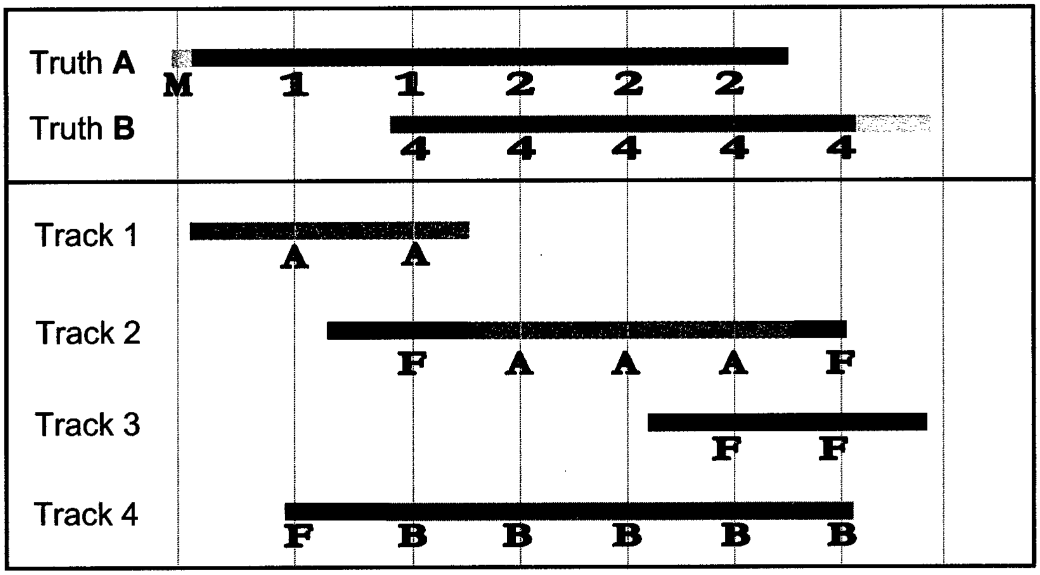
MISSED TRUTH - **M** FALSE TRACK -**F** ASSIGNED TO TRACK 1-**1**

Figure 1.10 Hypothetical assignments of Targets to Track using No Switch Strategy

c. Method 3: Restricted Switch Strategy

This method allows Track Switching under very limited conditions. A sequence of Tracks can be assigned to a Truth if the sequence is feasible. In a "feasible Track sequence" no two Tracks exist at the same time. Thus a Track cannot be started before the prior Track is terminated i.e. only after the termination of the Track that had associated to a Truth other Tracks get a chance to get associated with that Truth. Figure 1.11 shows a typical Track-Truth association using Restricted Switch Strategy. Here Truth 'A' is initially represented by Track '1' for all time points when Track '1' exists. Once Track '1' ceases to exist a new representative Track for Truth 'A' is selected which is Track '2' in this case.

First the assignment cost for each candidate Track-Truth pair is computed as described in method-2. Then for each Truth, all feasible sequences of Tracks are enumerated. The assignment cost for each feasible Truth-Track sequence is computed as described in method-2. Finally the assignment of Truth to Track or Truth to Track sequence is done by solving N-Dimensional assignment problem (N>2). The major drawback of this method is the computational complexity involved in solving N-D assignment problems.



MISSED TRUTH - **M** FALSE TRACK -**F** ASSIGNED TO TRACK 1-**1**

Figure 1.11 Hypothetical assignments of Targets to Track using Restricted Switch Strategy

d. Other Methods

Some ad-hoc methods to discourage Track Switching have been suggested.

1. Reduce the cost of current candidate Track-Target pair that was assigned last time.
2. Compute a "recent cost " using a moving window of the instantaneous costs to compute a weighted average of the costs.
3. Computed three tentative recent costs and choose the lowest of the three. The three recent costs are computed using
    a. A past cost window.
    b. A future cost window (past cost window computed backwards in time).

c. A centered window having equal number of past and future time points.

4. Uniquely assign a Track to Truth when the Track is first available and continue that assignment until the Track is terminated. Then uniquely assign another Track to the Truth.

1.4.2 Track Purity Method

Kovacich and Chong have described the use of Track purity for Track–Truth association. Track purity has been defined in strict sense and loose sense. In the strict sense Track purity measures the degree to which a Track is consistently updated with measurements from a single Target. In loose sense, it measures the degree to which a Track is consistently updated with measurements from a set of Targets.

In this method the score for an association hypothesis between Track 'i' and Target 'j' is computed as the ratio of # of measurements in the Measurement set (measurements which have cleared gating) for Track 'i' whose Truth set contains Truth 'j' to # of measurements in the Measurement set for Track 'i'. These scores are then used for Track –Truth assignment.

1.4.3 Measurement Source

Another possible way to solve the Track–Truth association problem is the use measurement source as the assignment criteria. This requires the evaluator to identify, which observations were used by which Track and which was the originating Target for those observations. This is not an easy task especially when the sensor fidelity in the simulation is high. It might require the evaluator to weigh the observation – Target and observation – Track relationship and then solve that assignment problem. Another drawback with this method is that it cannot be applied for on-field testing or lab testing where the real Truth data is being used.

1.4.4 Trial and Error Methods

It has been observed that often the user/system designer manually assigns Track to Truth for computation of metrics. These methods again enforce No Switch strategy.

1.4.5 Selection of Association Strategy

It can be observed from above discussion that Track-Truth association strategy has a deep influence on the design of Performance evaluation process. The Track-Truth association strategy can be divided into three categories the No-Switch strategy, the Switching Strategy and the Restricted Switching strategy. So which strategy is the best? There is no single opinion as to which of these strategies is the best because each of these strategies optimizes some aspect of Tracking. For example let us consider the scenario described in Figure 1.12(a)

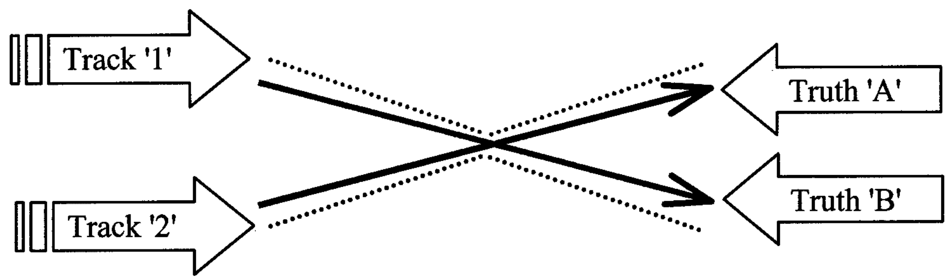


Figure 1.12 (a) Track to Truth Association Problem

Now methods adopting No-Switch strategy could yield a solution shown in Figure 1.12(b) for this problem. This solution would give a very poor overall accuracy while giving very high Track purity.

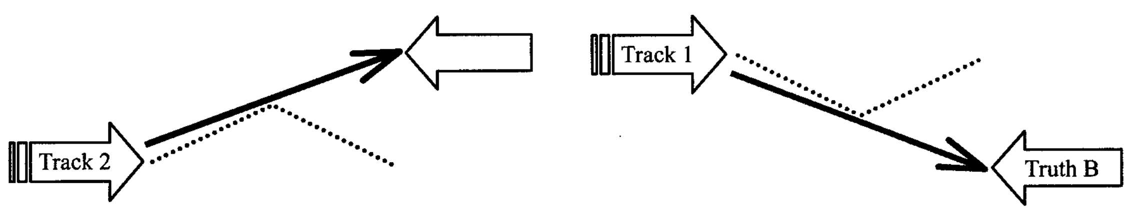


Figure 1.12 (b) Probable Track to Truth association using No-Switch Strategy

Methods adopting Switching Strategy might associate the Track and Truth as shown in Figure 1.12(c). Thus Truth 'A' gets represented by Track '1' part of the scenario and by Track '2' for rest of the scenario. It can be observed that such an assignment would result in better kinematic accuracy when compared to the assignment shown in Figure 1.2 (b). Using the Switching strategy one would get much better overall accuracy but one would have no control whatsoever over Track purity or Switching. On the other hand adopting the No Switch strategy could yield very poor ambiguity metrics.

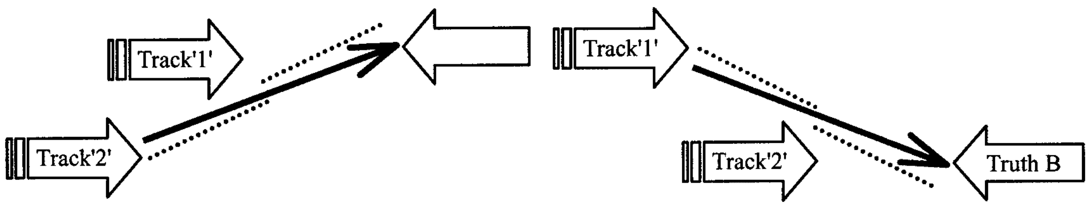


Figure 1.12 (c) Probable Track to Truth association using Switching Strategy

Also one could take the middle path of Restricted Switch strategy, which might yield better purity and Switch metrics than the Switching strategy while compromising the overall accuracy.

In short no particular strategy is the "best". Each strategy optimizes some metrics at the cost of other metrics. So the question arises what factors determine the selection of the Track to Truth association methodology for evaluation of Tracking System? The answer lies in the definition of the word "evaluation". The evaluation of a Tracking System is the process of determining the value of the Tracking System. The value of a Tracking System is something that has to be measured in context of the mission of which the Tracking System is a part. Hence it is the mission goal that dictates what the Tracking System should try to achieve. For example with reference to the No-Switch strategy Drummond states [14], "This methodology might be useful for a Tracking System that is used to feed a combat identification processing function". It is obvious that the basis for this statement stems from the interpretation that such missions require a Tracking System that does not Switch i.e. the system consistently updates Tracks from same Target. But then how do we confirm whether the system has a tendency to Switch or not? The only way to the study the Switching characteristic of a system is by evaluating it using a Switching strategy rather than using No-Switch strategy as Drummond suggests.

1.5 Performance Metrics

Abundance of literature discussing this aspect of evaluation of Tracking Systems is available. A brief summary of these from the perspective of Level1 data fusion is discussed here.

1. *Accuracy*

This criterion quantifies the accuracy of the Kinematics estimates of the system output. For example

a. Radial Miss distance (RMD)

It is measured as the Pythagorean distance between the Track position and the source position

b. Track Accuracy

As assessed for a particular object that should be Tracked, the root mean sum squared error (RMSE) history in position, RSME in velocity, the root sum squared average error

(RSSAE) history in position and RSSAE in velocity of the Track assigned to that object compared to the Truth states for that object.

c. Track Covariance Consistency

As for a particular object that should be Tracked, the mean normalized Chi- squared statistic of the Track assigned to that object.

2. Association Performance

This deals with the ability of the system to correctly associate the Track with their sources. For example

a. Number of Track Switch

The number of times the source of a Track Switched during the lifetime of the Track.

b. Track purity

Track purity is defined as the degree to which a Track uniquely represents a single Target. This can further be classified as loose sense of purity and Strict sense of purity.

3. Tracking System Responsiveness

This criterion deals with the time that the Tracker needs to come to a decision. For example

a. Track initiation time

Time taken by the Tracker to initiate the Track once the Target has entered the volume of interest

b. Track confirmation time

Time taken by the Tracker to confirm the Track after the time of initiation

c. Track deletion

Time taken by the Tracker to delete the Track once the Target has left the volume of interest

4. Ambiguity

This criterion measures the redundant and spurious Tracks in the output. For example

a. Redundant Track Mean Ratio

In a gated non-unique assignment, the number of declared Tracks that are assignable to real objects, divided by the number of valid declared Track.

b. Spurious Track mean ratio

In a gated non-unique assignment, the number of declared Tracks that are un-assignable to real objects, divided by the number of valid declared Track.

5. System Loading

This criterion measures computational load necessary for the system. For example

a. Communications Data Loading

Sum of rates input from all platforms into the communications function for distribution to other platforms.

b. Processor Loading

Peak number of floating point operations per Tracker per scenario per second.

1.6 Summary

In section 1.3.3 we discussed several issues and problems associated with simulation based performance evaluation. Following this in section 1.4 we presented a critical discussion of the various performance evaluation methods and philosophies that have been observed to date in the tracking community. Now keeping in sight the various methods of Performance Evaluation and issues surrounding them, let us summarize the steps generally involved in the performance evaluation process design.

1. Selection of Performance Measures

As mentioned in the previous chapter the assessment of delivered value of the MTTS should be done in the light of mission goals and objectives i.e. the mission goals and objectives need to be translated into measures of performance. The process of this translation gives the system designer an insight to the critical tracking performance measures. An important aspect of the overall evaluation process is that the underlying problems are temporally dynamic; thus, some sort of strategy needs to be defined to account for these dynamics in the evaluation approach and in the computation of performance measures. The definition of performance measures should therefore comprise what is to be measured, the frequency at which it has to be computed and how it is to be aggregated (i.e. over time or over various simulation runs, etc.).

2. Selection of Track-Truth Association strategy

The critical tracking performance metrics play an important role in the selection of the Track-Truth association strategy. For example, for applications involving the evaluation of combat identification processing as affected by association ambiguities, switching is discouraged [14]; hence the only option (of those explored in this thesis) is to adopt the No Switch approach. If for some mission purity is critical, then the emphasis would be to achieve high purity. This would require the system to be rigorously tested for switching behavior, which cannot be done using the No-Switch approach.

3. Selection of the Gating Criterion

In the same way as for regular (i.e. systems-under-test) Association processes, gating is used in Performance Evaluation to eliminate improbable Track-Truth Associations. Usually selection of gating criteria is an easy task since such criteria are not affected by other factors

4. Selection of the Scoring Method

The performance metrics and the association strategy affect the selection of a scoring method. For example, if the No Switch approach has been selected, and purity is more important than accuracy, then the scoring method could be based on Kovachich and Chong's Track purity approach [16]. But if accuracy is important then distance based scoring could be used. Also, depending on the association strategy, the computation of average scores might be required.

5. Assignment Technique Selection

Selection of the assignment technique for Track-Truth assignment depends on many factors (refer to Section 1.1.2.1.3 on Hypothesis Selection). The most critical factor affecting assignment technique selection for Track-Truth association is the association strategy. For example, Drummond's Method #3 based on a Restricted Switching approach requires an N-D assignment formulation.

6. Data Flow Design

Data flow design is the process of defining the data and control flow within the various components of the PE module. The data flow design process also specifies how the PE module is to be integrated with the components of fusion system under test.

Thus, *the task of designing the Performance Evaluation process is as complicated as the original task of designing a Data Fusion-based Tracking System.* A close look at the steps summarized above reveals that the mains steps involved in designing the PE process are similar to those involved in designing the data fusion process for the SUT. In this research effort we propose a formal design framework for the Performance Evaluation process based on the data fusion tree paradigm. Such a framework would facilitate reusability of the software/hardware components designed for the data fusion process for the SUT—ie; to some degree it should be possible to share SUT and PE process components.

The remainder of this thesis is organized as follows: Chapter 2 presents a Formal Framework for PE System Design followed by a Case Study Implementation of the Framework in Chapter 3. In Chapter 4 we present the Case Study implementation results and analysis. We conclude with conclusion and recommendations in Chapter 5.

Chapter 2

Performance Evaluation System Design Framework

2.1 Introduction

In section 1.4 we presented a critical discussion of the various performance evaluation methods and philosophies that have been observed to date in the tracking community. Following this in section 1.6 we presented a summary of steps involved in designing a PE system. In this chapter we present a formal framework for PE system design, which is one of the main focal points of this research effort. In doing so we heavily draw upon the existing (and widely excepted) frameworks, models and techniques of the target tracking community. This has two major advantages – (1) Facilitates reusability of existing s/w and h/w components and (2) Using standard frameworks and norms makes it easier for the tracking community to easily adopt it – thus giving this aspect of tracking a highly needed jumpstart.

2.2 The PE System Design Framework

The PE System Design Framework is shown in the Figure 2.1. The Framework is partitioned into 4- hierarchical levels each of which looks at the PE system from a different depth i.e. the very first level looks at the PE system with reference to its environment whereas the lowest level deals with the intrinsic of the PE system

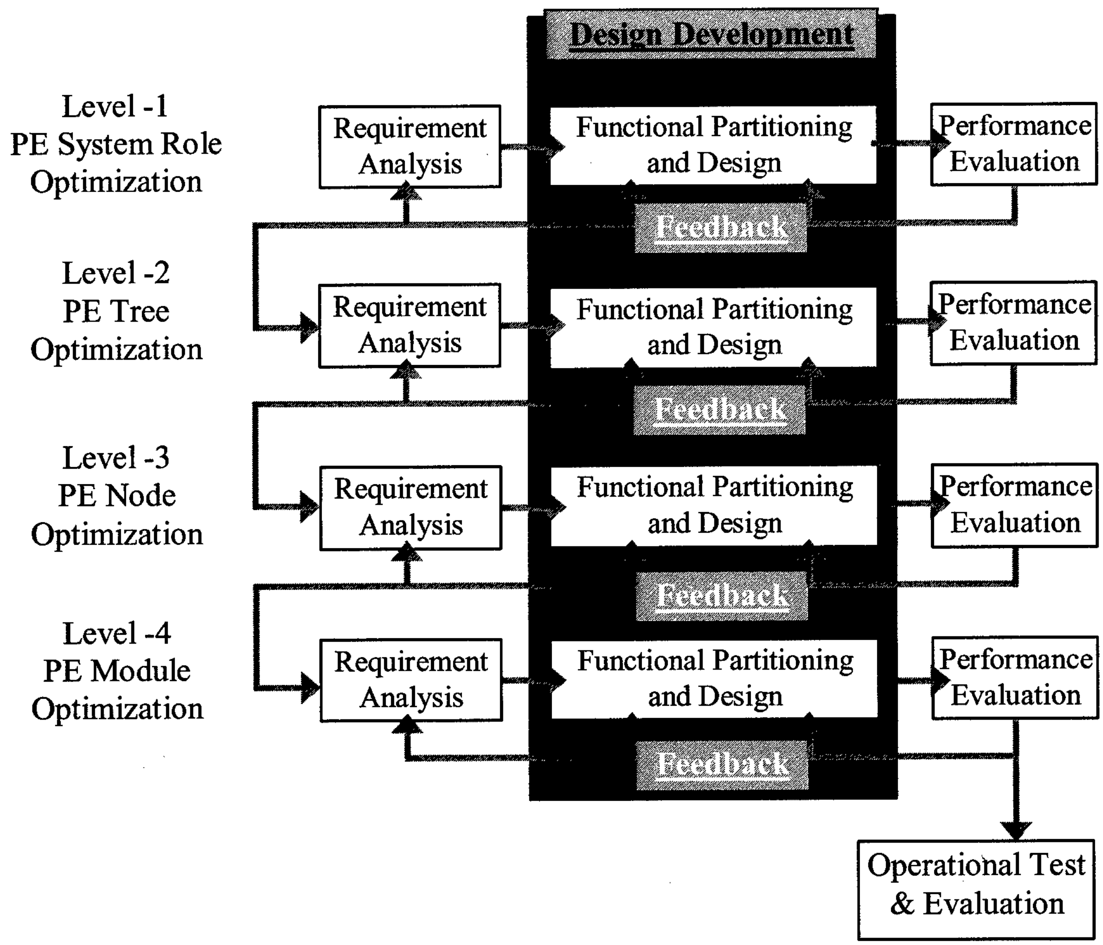


Figure 2.1 PE System Design Framework

components. Each level goes through three phases, namely the requirements analysis phase, the actual design phase, and the evaluation phase. Each sub-level provides feedback to the predecessor level, which might trigger another set of design iterations through the hierarchy.

2.2.1 PE System Role Design

PE System Requirements Analysis

This phase begins with an in-depth analysis of the role of the SUT within the C3I infrastructure, i.e., within the intended application context. This analysis gives insight to the critical aspects of SUT performance, which the PE system should estimate. Said in other words this analysis leads to a broad definition of the MOPs, which the PE system should be capable of estimating.

The complete picture of the PE system needs is expressed as the concept of operations (CONOPS) for the system. The CONOPS [and the resulting PE system design criteria, and constraints] defines the problem for the PE system design development. In summary, the CONOPS provides the following:

- Objectives of the SUT,
- Description of SUT operational scenarios and activities,
- Description of the SUT output data

- Objectives of the PE system i.e. the description of the characteristics of the SUT to be evaluated by the PE system.

PE System Functional Role Design

The PE system is just a subset of the MTTS development process. This second step in PE system role design development has two main goals, (1) to optimize the functional role of the PE system within the system environment and (2) to facilitate the seamless integration of the PE system as a "black box" within the system environment. This is an iterative process that culminates in a PE System Specification, which includes:

- System functional capabilities [to include capabilities implied from the CONOPS and the modes of the system required in support],
- Data bases [to include requirements on data bases that must be incorporated into system]
- External I/O interfaces [to include descriptions of physical interfaces, communication media, nature of data, and security issues to external systems, support environments, data bases, and users],
- HW/SW environment [to include physical/environmental characteristics. computer equipment, and support SW that must be used by the system],
- Software life-cycle cost and complexity (i.e., affordability);
- Robustness to errors/mismodeling (i.e., graceful degradation);
- Ease of user adaptability (i.e., operational improvements and personalization);

- Result explanation to user (i.e., responds to queries to justify result).
- Documentation [to include requirements for manuals, test plans, procedures, training, and other descriptive materials in various hard copy, electronic, and audio/video forms].

This "black box" design determines the relationship of the proposed system with respect to the other supporting systems.

2.2.2 PE Tree Optimization

After the role for the PE system (i.e., as a black box) is defined as described above, the component phase design is optimized using a similar feedback process to that described above. Namely, a further refinement of the requirements defines the problem, from which a PE tree design solution is developed, and then iterated upon.

PE Tree Requirements Analysis

The PE system role requirements are further refined as necessary to optimize the batching of the data for optimal MOP estimation. Note that data batching is a main driver to the particular structure of the PE Tree, i.e. that the layers of a PE Tree reflect layers of PE process partitions, largely defined by how the SUT results will be aggregated for MOP estimation. These requirements mainly focus on the input data quality, availability, and timeliness, as well as the corresponding output requirements – mainly the MOPs. A crucial point to be noted is that the relationship between the input data characteristics (note that PE inputs are SUT outputs, i.e. Track estimates) and the MOPs characteristics is a driving factor in the selection of Track – Truth Association strategy which in turn can affect the PE Tree design.

More specifically, requirements refinement includes more detailed descriptions of the range of tracking-problem scenarios that the SUT might be subjected to, SUT output data description, system output requirements, system functional capability requirements, the SW/HW system environment I/O, and user interfaces. Sufficient description of the problem is provided in this step to enable the PE tree design development trades on performance vs cost/complexity.

PE Tree Design Development

The Performance Evaluation MOP estimation requires the association of the SUT Tracks to the Truth entity states, since the basic evaluation context is based on differences between SUT output and the truth states. A trade-off of evaluation system performance versus complexity must be made to design the Track-Truth Association strategy and resulting MOP estimation approach and associated software design.

The PE tree design process describes how the SUT Track data is to be batched (e.g., over time, scenarios, platforms, object ID data, sensors, reports, etc.) for processing by PE nodes. For the range of problems addressed in this thesis, the PE tree design is mainly governed by the Track to Truth association strategy adopted. However, in more complex applications, the tree design may be influenced by factors similar to those affecting the design of the Fusion Tree for the system-under-test (ie, processing speed, other cost-driving factors such as ease of interpretation of results, etc). Thus, on a case-by-case basis, the factors influencing the design of a PE tree will be more or less similar to the design considerations for the application.

The selection of an effective PE tree requires understanding the data inputs, the SUT fusion requirements and above all the PE system requirements. The tree structure provides the designer with a formal mechanism for an understanding of how difficult the problem is, how accurately he needs to solve it, and under what processing and cost constraints. To reduce complexity/cost he wants to accomplish as much as possible as early and as easy as possible in the processing.

Those PE trees with simpler PE node processing are those, which use smaller batches of data. However too small batch size may not create a sufficiently broad perspective for MOP estimation. On the other hand too large batch size may create unnecessary computational overheads (generally, computational complexity increases exponentially with relation to input data size), with hardly any improvement in the MOP estimation process quality. For example, when partitioning input data based on the timestamp it bears, batch size is determined in accordance with a time window of size "Δt" i.e. the n^{th} batch consists of all the input data generated between time "$T + (n-1)*\Delta t$" and "$T + n*\Delta t$". Now if "Δt" is too small then it might happen that some of the Truth data gets assigned to one batch and the SUT Track data representing that particular set of Truth data might get assigned to the next batch. (In high fidelity simulations there is always a time lag between the timestamps on the truth data and track data). But a high value of "Δt" will result in very big batch size with no extra advantage. In other words the size of the batch of input data – in this case governed by the choice of "Δt", should be chosen to be only as large as necessary to enable the consideration of the other data needed to achieve sufficiently accurate Track-to-Truth association and MOP estimation results. Thus, the size of the batches to be selected must trade-off performance versus computational complexity/cost. Whenever a batch of data can be segmented from other data for association and MOP estimation and still achieve the performance requirements then a simpler PE node type can be inserted with reduced overall solution complexity.

Figure 2.2{a, b, c, d} shows some examples of simple PE trees. The circles denote a data source, which could be a database or a simulation process providing the truth data and the output track data. The boxes denote the PE tree nodes.

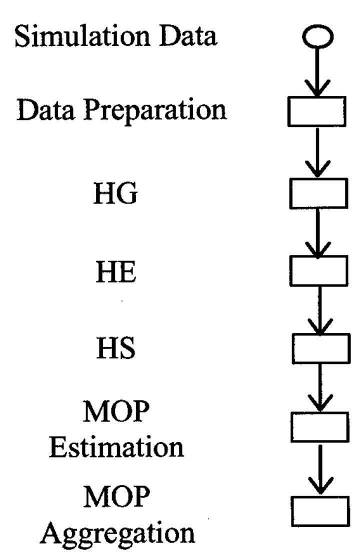


Figure 2.2(a) A Simple Single Node PE Tree

One approach to structuring a PE tree would be to take all the data from a scenario in one large batch and then perform an optimal association of the tracks to truth. Figure 2.2(a) displays such a simple PE tree wherein all the incoming data is processed together at one go. The data is aligned in the data preparation node prior to being sent to Hypotheses Generation and Hypothesis Evaluation nodes; in that order. The scored hypotheses are then passed on to the Hypothesis Selection node, which is the last node of the Track – Truth association stage. From the Hypothesis Selection node the Track-Truth pairs are directed to the MOP estimation node. These estimates are then sent to the MOP aggregation node, which creates an aggregate picture of the MOPs. Such an approach is usually infeasible, as it requires extensive computation and perhaps even unworkable computation; it also implies ND assignment problem solutions that are NP-hard and very complex, even to structure the solution.

Figure 2.2(b) shows a simple "recursive" PE tree wherein the incoming data is batched according the timestamp it bears (Δt as described above). Thus for a given time window all the data obtained during that period is processed together by invoking a new instance of the PE Node. The PE Node is recursively instantiated in this type of Tree design. The Track-Truth pairs are from the HS node are directed to the local MOP estimation node. The local MOP estimation node generates estimates of measures with reference to the data input for the given window only. These estimates are then sent to the MOP

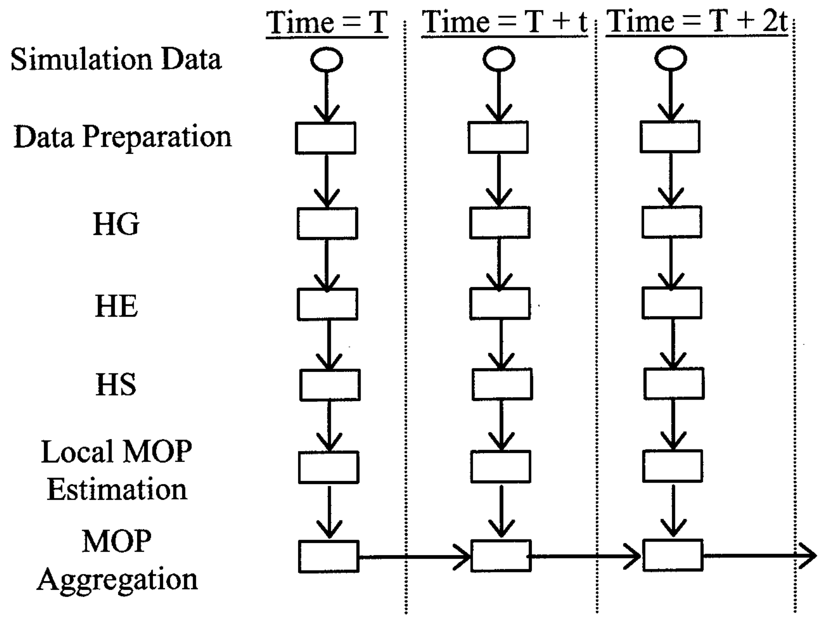


Figure 2.2(b) A simple "recursive" PE Tree

aggregation node, which creates an aggregate picture over all the local MOP estimates till the current time window. The aggregate MOP picture is transmitted from one time window to the next.

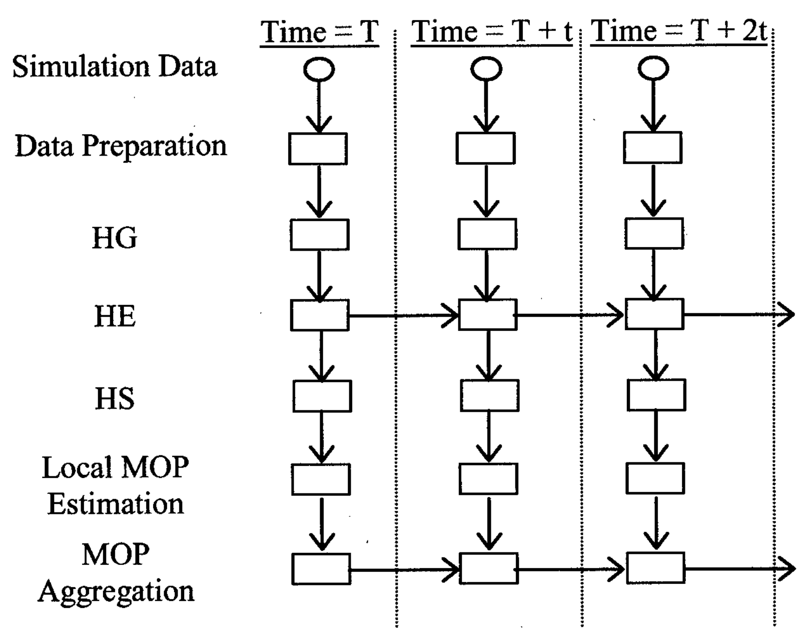


Figure 2.2(c) A Recursive PE Tree with memory

Figure 2.2(c) shows a time-batched recursive PE tree wherein the hypothesis scores from the HE node of one time window are passed on to the HE node of the next time window. Such a tree would be used when the HE process is designed to "learn" from the past. As an example the HE node could use the previous hypothesis scores to "average out" the effects of erratic data. Figure 2.2(d) shows another example of a PE tree wherein the HE node for a given window learns from previous windows. In this tree the MOP aggregate picture is passed on to the succeeding window's HE node. The HE node refines its scoring equations based on the aggregated MOPs. For example the probability of False Alarm estimated in the MOP aggregation node of a preceding window could be used to update relevant probabilities in Hypothesis scoring equations of the current window.

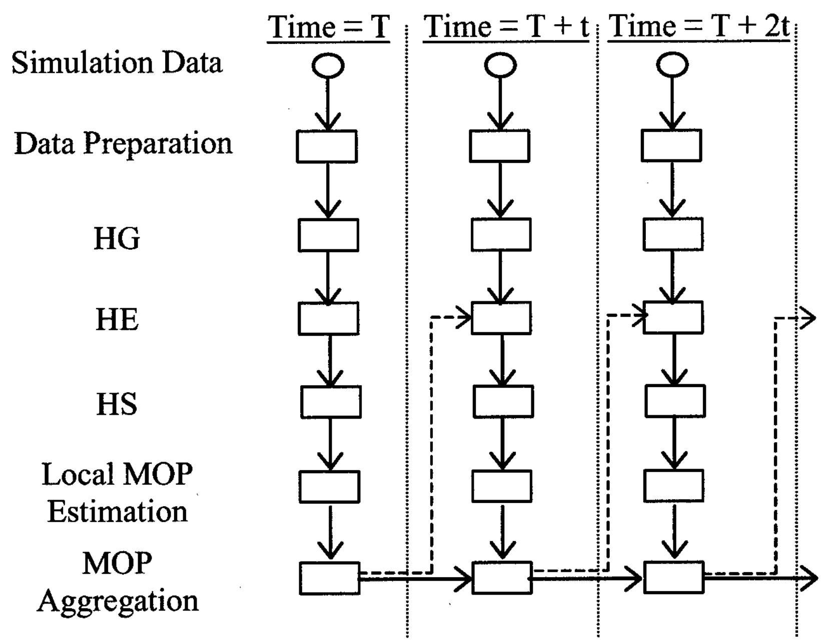


Figure 2.2(d) A Recursive PE Tree with memory

The PE trees in Figure 2.2{a, b, c, d} are "single node" trees i.e. these tree designs contains only a single PE Node which is invoked recursively for each time window.

Figure 2.3 shows a complex PE Tree consisting of four different PE Nodes, which are recursively invoked for each data batch. This PE Tree is designed to study the quality of estimates obtained from different sensors and fusion nodes of the SUT. Each of the four PE Nodes processes the data independent of the other nodes and reports its MOP estimates separately.

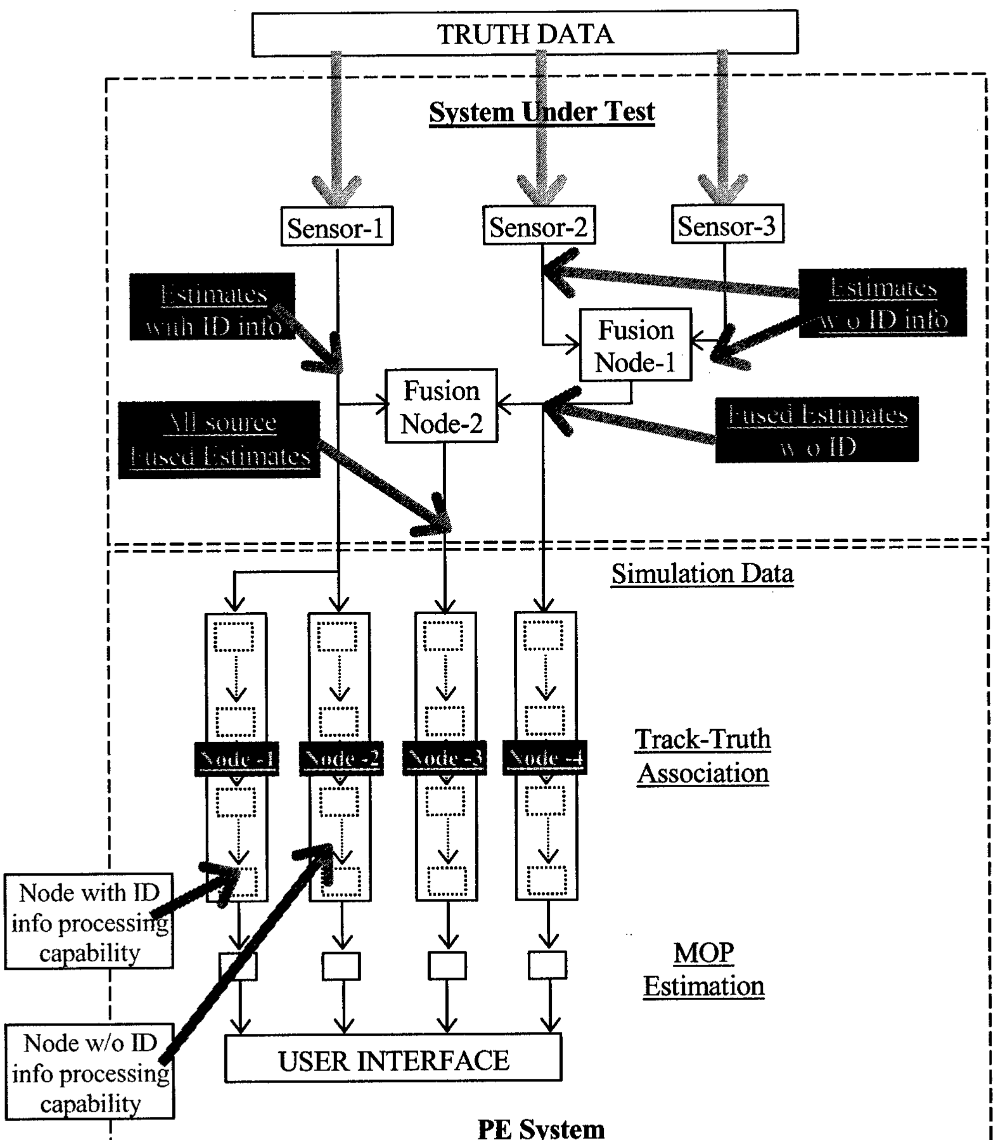


Figure 2.3 A complex PE Tree

As shown in Figure 2.3 the SUT consists of 3 sensors and 2 Fusion Nodes. Sensor-1 generates estimates with Identity information besides the usual kinematics data. However Sensor-2 and Sensor-3 provide kinematics estimation only. Fusion Node-1 fuses the kinematics data obtained from Sensor-2 and Sensor-3. This is followed by Fusion Node-2, which fuses the "fused" estimates provided by Fusion Node-1 with the estimates obtained from Sensor-1 (which includes identity information).

Now in this PE System the nodes, PE Node-1 and PE Node-2 process Sensor-1 estimates. Here PE Node-1 is designed to deal with "sensor estimates" having identity information besides the kinematics information whereas PE Node-2 ignores the identity information and process only the kinematics information. PE Node-3 processes the fused estimates obtained from Fusion Node-2 (which as mentioned above performs identity fusion besides kinematics data fusion). The last node i.e. PE Node-4 processes the sensor-sensor fusion estimates obtained from Fusion Node-1.

This PE System would be extremely useful in analyzing the role of the identity information obtained from Sensor-1 in overall SUT Fusion process output data quality. Since this PE System is designed to facilitate analysis and comparison of the various components of the SUT, it is critical that the MOP estimation process should bias free. Also the format of the MOPs and the frequency of MOP estimation should be commensurate amongst these nodes.

2.2.3 PE Node Optimization

At this third level the PE Tree Node processing design is optimized.

2.2.3.1 PE Node Requirements Analysis

Based on the Tree Design the requirements analysis in this phase provides further refinement for each PE node to include detailed descriptions of the node input data and form of output data expected from it. Sufficient description of the problem is provided to enable the PE node design development trade on performance vs cost/complexity.

2.2.3.2 PE Node Design Development

PE tree nodes are specialized by the type of input batching, and can be categorized according to combinations in a variety of dimensions. The design of a PE node within a PE tree involves selection among alternative techniques for preparing, associating and combining data received by the node.

The PE node should be able to carry out the following functions

1. Align the data
2. Generate Track – Truth association hypothesis
3. Score Track –Truth association hypothesis
4. Perform Track – Truth assignment

5. Compute PE metrics; both local and global metrics as per design specifications.

Thus the PE node structures are quite similar to the SUT Fusion Node.

Based on these requirements, the general PE node structure is as shown in Figure 2.4. The basic components of the PE node are discussed in the subsequent subsections.

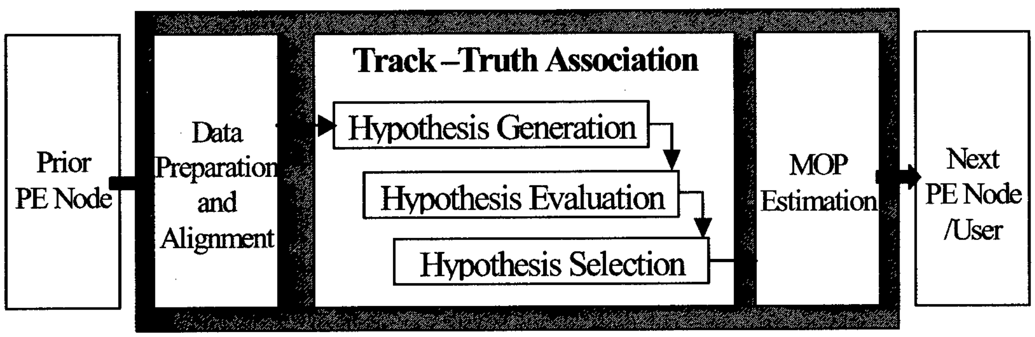


Figure 2.4 General PE Node Structure

2.2.3.2.1 Data preparation And Alignment

This stage involves processing and refinement of the data and estimated performance metrics for use by the subsequent nodes. Depending upon the simulation system and the means of data storage used in the system, the data processing steps may involve time and spatial alignment of data, metrics conversion, removing duplicate or redundant data, making structural changes to the way data is stored etc.

2.2.3.2.2 Track –Truth Association

As discussed previously the structure of the Track – Truth association problem is similar to the application-domain data association problem. Hence the software components employed for the Data association phase could be reemployed for the Track-Truth association phase. A key question here however, is whether using the same software components in track-truth association is justified from the point of view of assuring true objectivity in evaluation. One argument could be that this might cause bias in the performance metrics, whereas another perspective could be that reuse of software components would help understand the results produced by the fusion system under test. These issues will be addressed further in the concluding part of this research. The three components of the Track- Truth association phase are discussed below.

Hypothesis Generation (HG)

Hypothesis Generation is the step at which the range of what the analyst considers as feasible Track-Truth relationships are nominated. For example, under certain problem conditions it would be expected that the system-under-test would generate many Redundant Tracks due to high data clutter conditions. If so, the analyst would include such Track-Truth associations as feasible in the HG step. If the opposite were true then no such association hypotheses would be incorporated into the Track-Truth association process. Insightful decisions at this point are dependent on very good domain knowledge and good understanding of tracker operations and mathematics.

Given a set of computed tracks and a set of truth trajectories, track to truth hypotheses could be generated by associating every track with every truth and vice-versa. Associating each track with a non-existent "dummy" truth object generates a False Track hypothesis. Similarly, Missed Truth hypotheses are generated by associating each Truth object with a non-existent "dummy" track.

Having created the association hypotheses, the next step is to reduce the processing combinatorics involved in the overall association calculations by employing association gates. Generally, for track-truth association hypotheses, simple gating techniques are used. An example would be the use of a threshold on the Euclidian distance between Track and Truth locations as a gating criterion.

Hypothesis Evaluation (HE)

In this phase the feasible hypotheses from the HG phase are assigned scores or likelihood values that objectively reflect the "closeness" of the candidate Track to a given Truth trajectory that falls in the association gate for the Track. In Section 1.4 we saw that the scoring techniques that are generally employed for this purpose are quite simple and straightforward. However it is our conviction that this problem deserves a treatment at par with the original data association problem. Employing advanced scoring techniques, which are now finding their way into real operational Data Fusion-based tracking systems, would be quite simple within our PE framework. The selection of a scoring technique would be governed by the availability of truth data, computational facility, and time, especially when employing the PE software in real time. [18] provides a comprehensive discussion on scoring techniques.

Hypothesis Selection (HS)

Once the HG and HE processes have been completed, the overall association process has reached a point where the "most feasible" set of both Truth and Track (State estimation process) relationships or associations exist, and the question is to find the optimal set of Truth-Track pairs for computing performance metrics. In spite of a sound scoring methodology, ambiguities often exist in determining which of the feasible associations is "best" in some way. The usual optimality strategy is to find the hypotheses with lowest total cost assignment. (Depending on the scoring technique, the optimality objective may alternately be to maximize the cost). As mentioned in Section 1.1.2.1.3 this problem is called Assignment problem in the domain of combinatorial optimization.

The input to this process is a two dimensional matrix (or matrices) whose dimensions are Tracks and Truths, and whose contents reflect in all cases the corresponding Track-Truth score values. (The matrices may be greater than two dimensions depending on the Track-Truth Association strategy; for example, a third matrix dimension could be time, such that the associations are optimized over time as well as Track-Truth combinations at a given time; such problems however are much more difficult that two-dimensional problems.). It should be noted that the matrix would be a square one with equal number of rows and columns. The reason for this is quite obvious; for example, if we have n tracks and m truths, then during the HG phase m dummy tracks and n dummy truth are added to create Missed Truth and False Track hypotheses. This results in a (m+n) x (n+m) matrix. The advantage of having a square matrix is that many assignment algorithms work only on square matrices.

2.2.3.2.3 MOP Estimation

As mentioned previously, the State Estimation process in a PE context relates to the estimation of a measure used for evaluation. That measure, one of a set of MOP's, is an estimate in the sense that Track-Truth associations exhibit some degree of imperfection in the same fashion that Observation-Track assignments do, and also reflect an imperfect policy for computation and evaluation. That is, no single evaluation policy (such as a No Switch policy) will be totally correct or sufficient for an objective, complete evaluation of an association and tracking algorithm under test. However, once the Track-Truth pairs have been assigned, various Measures of Performance (MOPs) are able to be estimated. The definition of an MOP prescribes when and how to compute it. We have already seen a variety of MOPs in Section 1.5.

For each node, the PE Node functions (i.e., common referencing, data association, MOP estimation, MOP aggregation) are designed. The algorithmic characterizations for each of these three functions can then be determined. The detailed techniques or algorithms are not needed nor desired at this point. However, the characterization of the type of filtering, parsing, gating, scoring, searching, tracking, and identification in these fusion functions is accomplished.

The emphasis is on achieving balance within the nodes for these functions in their relative computational complexity and accuracy. It is at this point, for example, where a particular Track-Truth Association strategy is adopted. The detailed design and development (e.g., the actual equations) are not done until this node processing optimization balance is achieved on this third level.

2.2.4 PE Module Optimization

The final level determines the detailed design of the solution "patterns" for each sub function of each node in the fusion tree. There is a further flow down of the requirements and evaluation criteria for each of the sub functions. The design specifications generated at the four levels are actually implemented as software code at this level.

2.2.4.1 PE Module Requirements Analysis

In this phase the requirements for each function of every PE Node is refined with reference to the PE Node design specifications.

2.2.4.2 PE Module Design

In this phase solutions are developed for performing each function of every PE Node based on the requirement specifications. For some of the functions the solution can be in the form of simple mathematical computations and numerical conversions for example the MOP estimation process takes Track-Truth pairs as inputs and computes the Euclidean distance between the two as a measure of Kinematics Error. But for some other functions the solution approach is not that simple. The two functions whose solution requires complex set of algorithms are the HE and HS. In the Tracking community there exist several techniques for achieving the HE and HS tasks. A discussion on the HE techniques is out of scope of this thesis – [18] is a comprehensive reference for HE techniques.

As mentioned in Section 2.2.3.2.2, the HS problem is basically a form of the Assignment problem. The solution technique for HS plays an important role in PE system performance for two reasons – (1) This is the most complex (computationally) function of the PE system (2) This is where Track-Truth pairs are selected. A discussion on Assignment techniques follows in the next subsection.

1..1.1. .1 Assignment problem

The assignment problem is one in which the goal is to obtain an optimal way to assign N resources to M processes (M <>= N), such that a resource can be assigned to only a single process and vice versa. Each feasible assignment of resource to process has a cost (or value; we have called this the "score" above) associated with it and correspondingly the optimal assignment strategy would be to minimize the overall assignment cost (or maximize the overall assignment value).

An N x N assignment problem may be defined as follows

Minimize (or Maximize) $\sum_{i \in I} \sum_{j \in J} \mathrm{Cij}\ \mathrm{Xij}$

Subject to

$$\sum_{i \in I} \mathrm{Xij} = 1, j \in J$$

$$\sum_{j \in J} \mathrm{Xij} = 1, i \in I$$

$$\mathrm{Xij} = 0{,}1,\ i \in I\ j \in J$$

Assignment Problems and Computational Complexity

Given the "mission" goals (or whatever the top-level goals or purposes of the systen-under-test are), computation-time and solution optimality are the important aspects governing the selection of a particular assignment technique. In spite of the efficient HG and HE processing, the assignment matrix still may not be very be sparse, and so generally the only way to assure an optimal, minimum cost solution is to conduct an exhaustive search of all feasible assignment patterns and costs. Usually complexity grows as an inverse function of scarcity and direct function of matrix size. In addition, multi-dimensional assignment problems in their general formulation are NP-hard and have to be solved by sub-optimal techniques.

Classes of Assignment problems

In the applications of interest here, the assignment problems can be broadly partitioned into two classes, two-dimensional (2D) and those having more than two dimensions, henceforth referred to as N-dimensional (ND). The two dimensional problem has already been discussed. The N-dimensional problem consists of 2D data sets over N time points. But because of the interdependence between the 2D data sets it cannot be treated as N independent 2D problems.

The two dimensional problems are generally tractable whereas the N-dimensional problems are more complex and generally deemed NP hard. Poore [19] has shown a partitioning method for solving the N-dimensional HS problem.

Solution Techniques for the Assignment Problem

A comprehensive discussion on Solution Techniques for HS is out of the scope of this thesis, however in order to give the reader an essence of it, a brief discussion on some frequently cited solution procedures for assignment problems is provided here:

- *Relaxation Algorithms*

The relaxation methods relax the single assignment constraint of the problem, thereby allowing multiple assignments, essentially an easier problem. The relaxation algorithm gradually builds an optimal assignment by identifying the lowest cost paths from over assigned inputs to unassigned inputs. The Hungarian method of Kuhn [20] is an example of this. The Hungarian algorithm is potentially slow.

In 1957 James Munkres presented a modified form of the Hungarian algorithm called the Munkres Algorithm [25], which is computationally faster. It is one of the most cited algorithms in data fusion applications. The Munkres algorithm is for what is often called a balanced problem (i.e square matrix). Kaufmann [26] used dummy entries for squaring off rectangular matrices to solve them using the Munkres algorithm. Bourgeois and Lassalle's modification [27] to the Munkres algorithm for solving non-square matrices is faster than Kaufmann's method.

- *Successive Shortest Path (SSP)*

The successive shortest path algorithm maintains dual feasibility of the solution at every step and strives to attain primal feasibility. At each step SSP selects an unassigned input and an unsatisfied estimator and assigns the input to the estimator with the lowest assignment cost. The SSP is similar to primal dual algorithm but instead of assigning multiple inputs it assigns one input at a given step.

- *Auction Algorithm*

Bertsekas[28] devised the auction algorithm for solving assignment problems. As the name suggests this method is based on the "auctioning process". The auction algorithm is also primal-dual type solutions. In this algorithm the estimators / processes act as bidders bidding on the input data. The estimators/ processes bid on the inputs using bid amounts based on the utilities or costs associated with each feasible assignment. At each iteration an unassigned estimator/ process bids on an input that has the highest marginal utility for that estimator.

The auction algorithm is potentially slow. To overcome this Bertsakas [28] devised a forward-reverse auction-based approach, which alternately has estimators bidding on inputs, and/or inputs bidding on estimators to caused faster matching.

- *Shortest Augmenting Path Algorithm.*

This is an extension of SSP conceptualized by Tomizawa [21]. He augments partial assignments into a complete assignment solution by primal steps in each of which one shortest augmenting path is determined. Jonker and Volgenant [22] developed a variant of this approach, a shortest augmenting path algorithm that exploits the augmenting cycle property and achieves good computational performance. This is one of the fastest algorithms for HS.

- *Branch and Bound Techniques*

Since we are dealing with problems that have finite solution boundaries, it is natural to consider some type of enumeration procedure for finding an optimal solution. In many cases however, the number of feasible solutions can still be very large since the problems tend to be of a factorial type in the number of variables and the number of values the variables can take on. However, if we could define an enumeration strategy, which does not explicitly enumerate all feasible solutions but implicitly eliminates a large group of solutions without evaluating their cost functions, a practical approach might be produced. Such implicit enumeration techniques include the Dynamic programming approach and Branch and Bound (B&B) techniques. It should be noted that these methods are usually termed, as "strategies" because the specifics of implementation are highly problem dependent, and the methods are not explicit algorithms in the usual sense.

The basic idea behind B&B is the following. Suppose that an objective/cost function is to be minimized. Assume that an upper bound on the optimal values of this function is available. In B&B approach, the first step is to partition the set of all feasible solutions into several subsets, and for each one to determine the lower bound on the value of objective function for all solutions within the subset. Those subsets with lower bound greater than the upper bound are then excluded from further consideration (the excluded subsets are said to be fathomed). Of the remaining subsets, the one - say with the lowest lower bound is then further partitioned into subsets and the reduction process continues. The branch step is one in which the decision of which subset to expand into further subsets is made. Two most popular branching rules for selecting the subset are best-bound rule and the newest bound rule. The best-bound rule selects the subset having the most favorable bound (e.g. smallest lower bound for minimization problems). The newest bound rule selects the most recently created subset. The bound step is one in which the lower bound for each subset is calculated.

The B&B method is a potentially useful method (computationally tractable) for solving ND assignment problems. Although the solutions have exponential time behavior, it is quite appealing in ND assignment problems that are generally classified as NP-hard. There are a few citations of application of B&B methods for multiple target tracking problems – for example [23]. An important point of consideration while employing B&B methods is selecting a stopping point for which the solution is good enough.

2.3. Summary

The four phases of the PE design framework are summarized as the following:

1. <u>PE System Role Design</u>: analysis of system requirements to determine the relationship of a proposed data fusion system with respect to other systems with which it interfaces.

2. <u>PE Tree Design</u>: defining how the data is batched to partition the Track-Truth Association problem.

3. PE Node Design: defining the data and control flow within the nodes of a selected fusion tree.

4. PE Module Design: defining processing methods for the functions to be performed within each fusion node.

The topmost level i.e. the PE System Role Design as said earlier is a very meta-level view, mostly focused on designing a black box system that seamlessly integrates with the concerned environment. The lowest level i.e. the PE Module Design mainly involves designing algorithms and generating software in accordance with the Node Design and Tree Design. In other words the issues and problems encountered at these two levels are more or less commonplace (for the Tracking System Designer); thus rendering the role of these levels inconsequential in the development of the nascent PE technology.

Of the four levels of PE System Design framework the Tree Design and the Node Design are very critical since the core aspects of the PE problem are dealt with at these two levels. For the advancement of the PE technology it is crucial that the Tracking community gets more insight into the issues and problems encountered at these two levels.

Chapter 3

A Case Study Implementation of the Performance Evaluation (PE) System Design Framework

3.1 Introduction

In Chapter-2 we presented the PE System Design Framework. In this chapter we present a Case Study implementation of this Framework. The main objectives for this implementation were

- To demonstrate a proof of concept for the Framework
- To demonstrate the need to address the Track-Truth Association problem in a single but representative case.
- To understand the interdependencies between the Track-Truth Association strategies and the MOPs.
- To identify critical issues/roadblocks in the development of a PE System.

In the following subsections we shall discuss a phase-by-phase implementation of the PE System Design Framework.

3.2 PE System Role Design

The role for the PE system in this Case Study is to estimate the performance of a typical or generic tracking SUT in a Case Study that involves some of the challenges to PE tree design. No "Mission" context for the use of the particular SUT is specified as the basic role for PE here is to serve as a generic PE process model for a relatively simple but demonstrative example. This is a constraint to the overall role for the PE process since a mission context would provide a basis for effectiveness evaluation of the SUT, but the purpose of the Case Study is to raise some general issues in PE tree design, and to show some of the design tradeoffs involved.

3.2.1 PE System Requirements Analysis

For our purpose this phase involves a thorough understanding of the SUT that leads to systematic evolution of the PE System objectives. The SUT description is presented in Section 3.2.1.1, which is followed by discussion on PE System objectives in Section 3.2.1.2. The basic PE requirement here is to enable an evaluation process for a case where the SUT association ambiguity extends over a range of values and thus a range of complexity in defining an equitable evaluation approach.

3.2.1.1 Description of the SUT

Prior to understanding the requirements of the PE system it is important to understand the "Test Object" that the PE system is supposed to evaluate. The SUT in this case is built using Level-1 Data Fusion simulation tool CASE_ATTI. The characterization of the SUT is as follows.

(a) Objective of the SUT

Since we are considering the SUT out of context of the missions, which it might support, the only high-level objective of the SUT is to Track the multiple targets in the scenario accurately, and continuously.

(b) Description of the SUT operational scenarios

As said earlier, one of the main objectives of this case study is to study the interdependencies between the Track-Truth Association strategies and the MOPs. In order to bring out the characteristics of the Track-Truth Association problem the operational scenarios were designed to be highly dynamic and consisting of multiple, maneuvering objects in trajectory configurations that lead to potential association problems. The main characteristics of the operational scenarios are as follows:

- Multiple objects (# of objects >1).
- Objects displaying kinematics variation

- Objects performing high speed maneuvers (even cross over maneuvers)
- Object spacing varying drastically over the scenario (sometimes closely spaced and at times very much apart)

Two sets of scenarios were designed with the above requirements in mind. The first scenario set contained only one simple scenario and the other scenario set contained three scenarios. These scenario sets are discussed below

Scenario Set 1: Single Feigned Crossing Target Scenario

The driving factor behind this scenario was to demonstrate how a simple, one time switch in the SUT data association process can lead to highly disparate evaluation of the SUT under different PE Track-to-Truth switching methods. This scenario is shown in Figure 3.1. As shown in the figure, the scenario begins with two Targets far apart from each other. As the scenario proceeds the Targets come very close to each other (approximately 200 meters apart) and then turn around and diverge. This maneuver forces data association errors in the SUT, and as a result the SUT output shows two Targets traveling in straight lines and crossing each other.

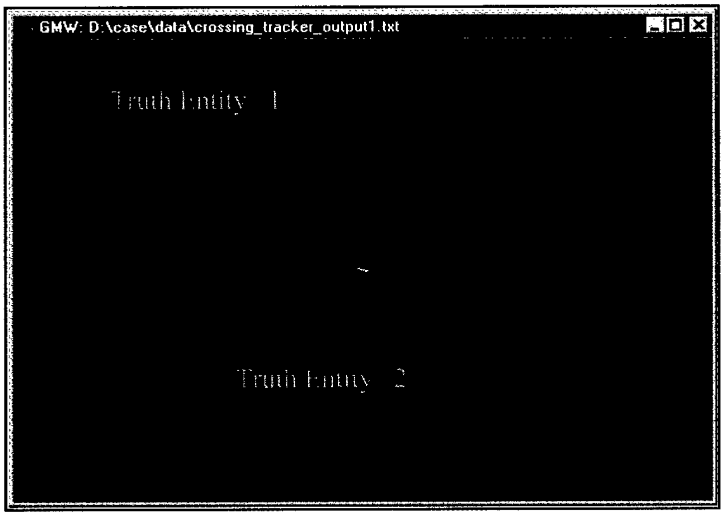


Figure 3.1 "Feigned" Crossing Target Scenario

Scenario Set 2: Maneuvering Targets with Intertarget Spacing Varying over Scenarios

Blackman [30] has shown that Target spacing is a scenario parameter that can create ambiguities in data association; simply put "the closer the Targets are, greater is the confusion in data association", although the degree of induced ambiguity is itself a function of yet other factors, such as sensor resolution. In section 1.2 we presented an elaborate discussion as to how the data association ambiguities give rise to the key issue in the evaluation process i.e. the Track-Truth Association problem. A set of 3 scenarios was designed to demonstrate how the Track-Truth Association process inherits the data association ambiguities created due to inter Target Spacing. As shown in Figure 3.2, in these scenarios four sets of Targets perform high-speed maneuvers. The only difference in the three scenarios was the variations in inter Target spacing. The first scenario had

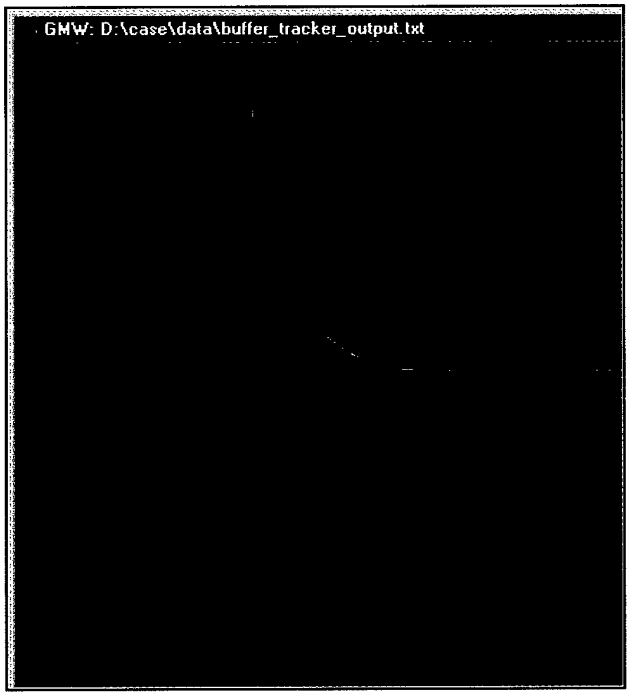


Figure 3.2 Generic Trajectories for the Maneuvering Targets

Targets flying very close to each other; as a consequence the data association ambiguities were high. In the second scenario the Targets were some what separated creating infrequent association ambiguities. The third scenario had the Targets traveling widely apart from each other resulting in no association ambiguities except for the instant when the Targets crossed each other (this crossing occurs in the middle of the turns for each target pair).

While designing any scenario an important consideration was that it should be complex enough to capture the effects of a *real operational scenario* without overloading the hardware support available for the PE System.

(c) SUT output data description.

The output data from the SUT was available in two formats – (1) Large text files containing unlabelled numeric values – (2) Data distributed into several tables of an Oracle database.

The summary of SUT output data characteristics is as follows

- Object ID – for both Truth and the Track
- Kinematics data (position, velocity and acceleration) – for both Truth and the Track
- Time-stamp – for both Truth and the Track
- Identity information (for identity tracking). – only for Truth. The version of SUT used in this case study did not possess Identity tracking capability; hence there was no identity information in the Tracking data.
- Tracker covariance matrix
- New data arrival rate <2 sec.

3.2.1.2 PE System Objective

One of the main goals of this case study is to study the effects of different Track-Truth Association strategies on MOP's over a range of association-complexities, in order to develop some initial empirically-based knowledge about the sensitivities of Track-Truth association strategies to SUT association complexity and to problem-space (ie target spacing) factors. Given a Track-Truth Association strategy the objective of the PE System is to evaluate the overall Tracking performance of the SUT and present it to the user. Following three categories of performance were nominated for estimation.

- Kinematics Accuracy

  This criterion quantifies the accuracy of the Kinematics estimates of the system output.

- Association Performance

  This deals with the ability of the system to correctly associate the Track with their sources.

- Detection Performance

  This criterion evaluates the reliability of system in Target identification.

The reason behind nominating these three categories is that they broadly capture the essence of a Tracking system. Also there is no inverse relation amongst these measures i.e. tweaking the SUT for achieving better detection performance does occur at the cost of other two measures. Another important characteristic of these three measures is that the Track-Truth Association directly affects them.

3.2.2 PE System Role Design

For the purpose of this case study the interaction of the PE system is only with the SUT and the user. The interaction with the SUT is only in form of "pulling" the data from it and vice-versa is not true. Similarly the interaction with the user is also a one-way process in form of the PE System sending output data to the user. Thus the PE system should have following characteristics

- It should be able to connect to SUT output database and pull data from it. As we saw earlier the SUT has two formats for the output data. We choose to use the output in database format since "parsing" the output in text files is a complicated business. Also the database in question is Oracle database, which provides excellent data manipulation functionalities, which will come handy in data preparation and hypothesis generation stages.
- It should store the intermediate computations for purpose of later reference/validation.
- In this case study the PE System output has to be presented only to the user i.e. it does not need to be fed to the SUT as a feedback or to any other component of the MTTS development environment. The output to the user should be in two forms- (1) Detailed information should be presented in form of database tables and (2) Aggregate picture should be provided in form of view graphs.

3.3 PE Tree Optimization

In this section we shall discuss the design of composite tree enabling all switching strategies at once.

3.3.1 PE Tree Requirement Analysis

One of the aims of this case study is to understand the characteristics of various Track-Truth Association strategies. Keeping this in mind it was decided that the PE System should be able to implement following three strategies

- The Switching Strategy
- The No Switch Strategy
- Restricted Switch Strategy

Thus the PE Tree should consist of three different "recursive" PE Nodes, one for each switching strategy. In the following phase we shall see how these three Nodes are integrated to form the PE Tree.

3.3.2 PE Tree Design

A simple method of batching would be to create a new data batch every time the tracker reports an update on any track. The new data batch would consist of incoming tracker data (which may not contain information about all the existing tracks) and truth information for <u>all the targets</u>.

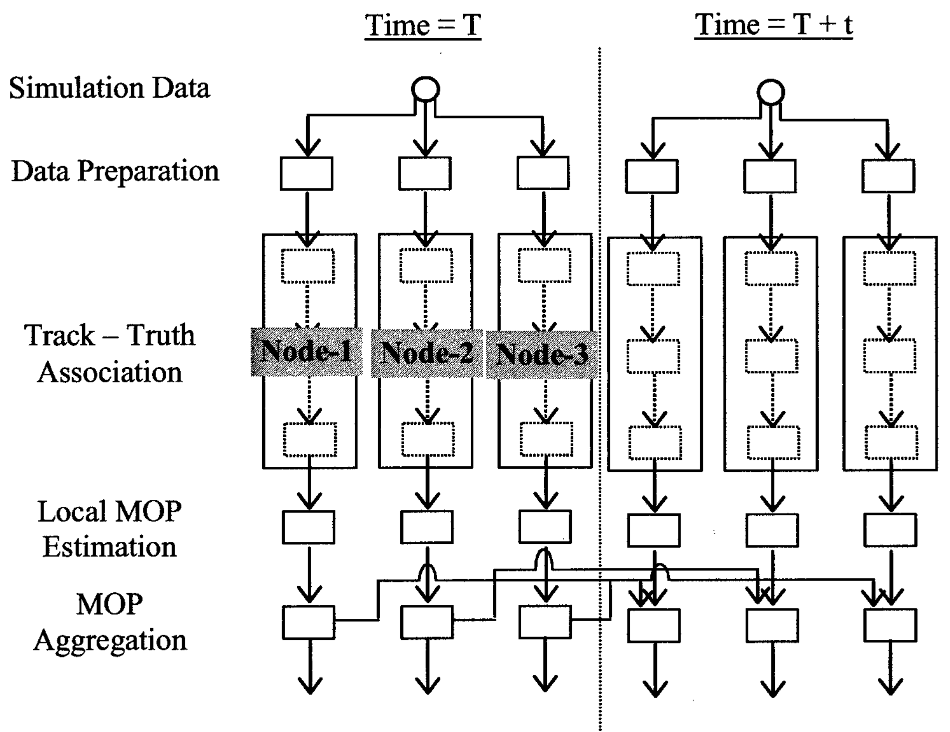


Figure 3.3 Case Study PE Tree Structure

Since the PE System needs to implement different Track-Truth Association strategies and facilitate their comparison, the PE Tree should consist of three independent PE Nodes. Each PE Node would be fed with the same data set but the Track-Truth Association process would be designed in accordance with the Track-Truth Association strategy allocated to that particular node. However the MOP computation process should be similar for each of the three PE Nodes so that the comparison between the various nodes is justified. Based on these specifications the time-batched recursive PE Tree structure is as shown in Figure 3.3. As shown in Figure 3.3 the PE Nodes are recursively instantiated for each batch.

3.4 PE Node Design

3.4.1 PE Node Requirements Analysis

For the PE Tree shown in Figure 3.3 three different PE Nodes need to be designed. These nodes should be able to support recursion since they will be invoked recursively for each batch of data. Depending on the point design the nodes may be required to have a "memory" which stores information during the recursive calls. Now lets see the individual requirements for each of the three PE Nodes.

(a) PE Node-1

This node has to implement the Switching Strategy. Enforcing the Switching Strategy requires that the Track-Truth Association process for any given data batch be independent of the processing of the adjacent batches. Thus the Track-Truth Association processing of PE Node-1 should be memory less i.e. to say that when the PE Node-1 is called during a recursion (or instantiated) it does not store and carry ahead any piece of information for its next recursion (or instantiation).

(b) PE Node-2

This node has to implement the No Switch Strategy. The No Switch Strategy requires solving the Track-Truth association problem for the whole scenario all at once. This calls for cumulating the association hypothesis scores for all the batches. In other words the PE Node-2 should be able to store the association hypothesis over its various recursions.

(c) PE Node-3

This node has to implement the Restricted Switch Strategy. We have seen several variations of this strategy in Chapter-1. One such strategy uses a moving window based average of hypothesis scores for Track-Truth Association. This variation of the Restricted Switch Strategy was adopted for the case study implementation. The following discussion explains the motivation behind adopting this strategy

Often there are isolated jumps in the Tracker output in which suddenly for one update the Track estimate is way off the real picture, but the very next update returns a normal estimate. When using the Switching strategy such jumps get reported as a Track Switch or a Missed Truth rather than reporting it as a jump in Kinematic Error. Hence to minimize the effects of such isolated jumps a moving window average approach is used. According to this strategy the hypothesis score fed to the HS process is an average of hypothesis scores over that particular window and a given number of its predecessor batches.

3.4.2 PE Node Design Development

3.4.2.1 Data Preparation and Alignment

This function was directed at establishing an efficient approach to handling and processing of the SUT output data. The simulation environment employed in this case study stores data in Oracle tables. One of the main processing steps is to structurally align the data since the data is distributed in several tables with structural inconsistencies among other disparities. An example of structural inconsistency – the information about ground Truth data is stored in a single table which contained the true Target identity and its kinematics information with reference to time, while the output information is spread across different tables some of which are stored with reference to time while others are referencing some other entity.

3.4.2.2 Hypothesis Generation

Three types of association hypothesis were nominated.

- Track-Truth Hypothesis in which output Tracks would be paired with Truth entities
- False Track Hypothesis in which output Tracks would be paired with non-existent "dummy" Truth entities.
- Missed Truth Hypothesis in which non-existent "dummy" Tracks would be paired with Truth entities

In this case study for the sake of simplicity the issues of "redundant/duplicate" tracks and spurious tracks were not treated separately; the False Track Hypothesis definition would encompass redundant and spurious tracks. The scenarios designed for this Case Study had no abrupt birth or death of a Truth entity i.e. all the Targets existed from the start of the scenario run till the end of the scenario. So New Target pop-up hypotheses or Target Drop Hypotheses were not required.

Euclidean distance between the Track and the Truth would be used as a gating criterion. The Track-Truth Hypothesis that fails the gating process would not be considered for the subsequent HE and HS processing.

3.4.2.3 Hypothesis Evaluation

For the three PE Node designs, the Association Hypothesis scoring technique would be the same however the method in which the hypothesis scores are aggregated/not aggregated over batches would differ.

(a) PE Node-1: The Switching Strategy Node

For the node implementing the Switching Strategy every batch is independent of the other batches. So the hypothesis scores computed for a given batch are used during HS process of that batch only. Thus the structural design of this node is same as the general node structure that was shown in Figure2.x and discussed in Section 2.xyz.

(b) PE Node-2: The No Switch Strategy Node

This node stores the hypothesis scores for each batch till end of the scenario. At the end of the scenario these set of hypothesis scores are to be used to solve the Track-Truth Association problem using some assignment technique. Now one can come up with several ways of solving the assignment problem for this large set of association hypothesis matrices; however even the near optimal solutions would be computationally complex. A simple way to solve this problem would be to compute the average hypothesis score, averaged over all the hypothesis score matrices. The assignment problem then will be reduce to just a single assignment matrix which can be solved easily. The design for such a PE node is shown in Figure 3.4.

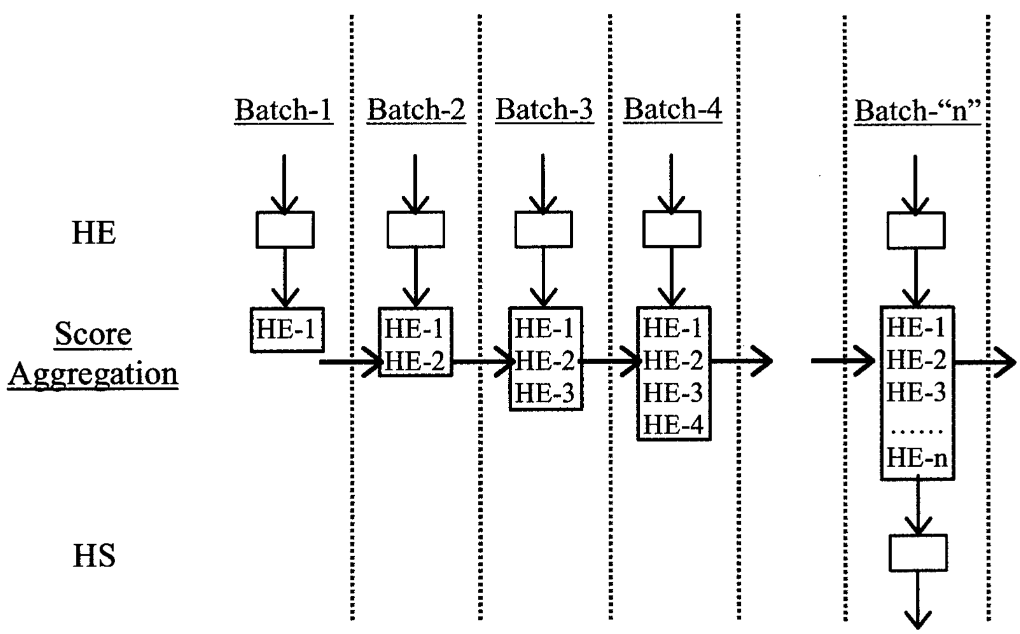


Figure 3.4 The No Switch Strategy Node Design

As shown in Figure 3.4 this PE node contains an additional component that is responsible for aggregating the HE results. At the end of the scenario this component feeds the aggregate scores to the HS component, which then processes the Track-Truth Association problem.

(c) PE Node-3: The Restricted Switch Strategy Node

The design for this node somewhat resembles the design used for PE Node-2. Figure 3.5 shows how the data flows through the various components of this node over several iterations/instances. The example shown in this figure has window size of "3" – said in other words for each instance of this node the Score Aggregation component keeps stored at most three batches of hypothesis scores. The average of these scores is fed to the HS component.

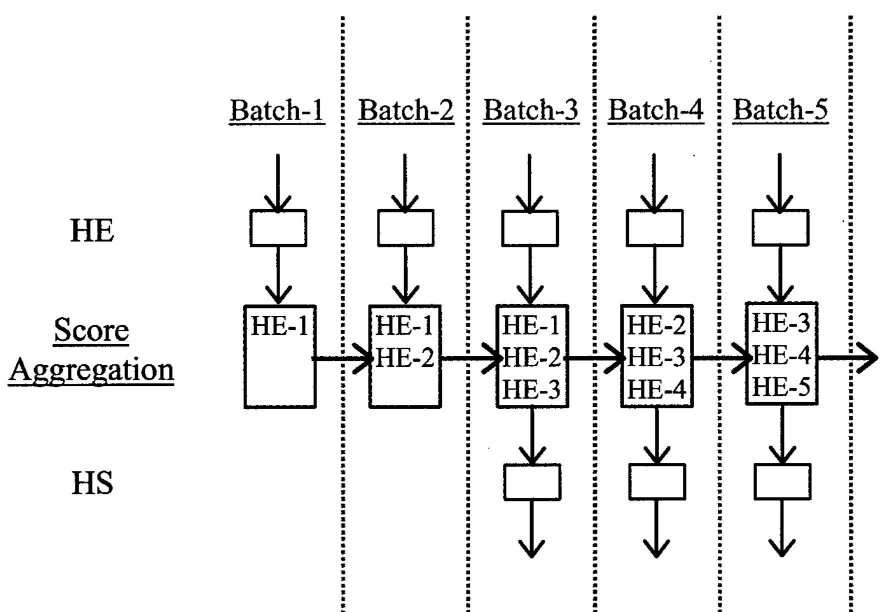


Figure 3.5 The Restricted Switch Strategy Node Design

The window size should be just big enough to smooth out the isolated jumps. However no particular criterion has been developed for nomination of an optimal window size.

3.4.2.4 Hypothesis Selection

The input to HS process for all the three PE Nodes is a square, 2-D matrix. Since we are restricting association of a Track to only 1 Truth and vice versa, this becomes a simple assignment problem. This makes the task for the selection of assignment technique very simple. In fact all the three PE Nodes can have exactly the same design for the HS process.

Kuhn's Hungarian Method was nominated for selecting the best hypotheses. Harold W. Kuhn devised this method in 1955[20] to solve the assignment problem. His algorithm is based on work by Hungarian mathematicians Konig and Egervary [24]. In honor of them Kuhn called his method the Hungarian Method. Section 3.2.3.1 discusses the optimality of the Hungarian Method and Section 3.2.3.2 gives the actual procedure used to implement the Hungarian Method.

Optimality of the Hungarian Method

Consider the following Assignment Problem.

$$\text{Minimize} \quad \sum_{i\epsilon I} \sum_{j\epsilon J} \text{Cij Xij} \qquad \text{-(1)}$$

Subject to

$$\sum_{i\epsilon I} \text{Xij} = 1, j\epsilon J \qquad \text{-(2)}$$

$$\sum_{j\epsilon J} \text{Xij} = 1, i\epsilon I \qquad \text{-(3)}$$

$$\text{Xij} = 0,1, \; i\epsilon I \; j\epsilon J \qquad \text{-(4)}$$

Now introduce a variable *Ui* for each row *'i'* where *Ui* = min Cij for that row. Introduce a variable Vj for each column 'j' where Vj = min (Cij – Ui) for that column.

Define new cost elements as,

$$\text{Wij} = \text{Cij} - (\text{Ui} + \text{Vj}) \text{ for each i,j and Wij} > 0 \qquad \text{-(6)}$$

$$\sum_{i,j} \text{Cij Xij} = \sum_{i,j} \text{Ui Xij} + \sum_{i,j} \text{Vj Xij} + \sum_{i,j} \text{Wij Xij} \qquad \text{-(7)}$$

Since Wij and Xij are non negative and (7) can be reduced as

$$\sum_{i,j} \text{Cij Xij} \geq \sum_{i,j} \text{Ui Xij} + \sum_{i,j} \text{Vj Xij} \qquad \text{-(8)}$$

Also due to (2) and (3) we can further reduce (8) as

$$\sum_{i,j} \text{Cij Xij} \geq \sum_{i} \text{Ui} + \sum_{,j} \text{Vj} \qquad \text{-(9)}$$

We can now formulate the dual of the assignment problem as follows.

$$\text{Maximize} \quad \sum_i Ui + \sum_j Vj \qquad -(11)$$

$$\text{Subject to} \quad Ui + Vj \leq Cij, \text{ for all } i,j \qquad -(12)$$

The inequality in (8) tells us that feasible values of the objective variable for the dual problem are always less than or equal to feasible values of the objective variable for the assignment problem. Thus if we can find feasible values for these two objective variables that are equal, each will be optimal for its problem. From (7) we can say that these two objective variables are equal if and only if,

$$\sum_{i,j} Wij\ Xij = 0$$

Since Wij and Xij are nonnegative it follows that Wij should be zero for non zero values of Xij. Thus if we can find a reduced cost matrix with 'n' independent zeros we have a feasible solution for the dual as well as the assignment problem. This forms the basis of the Hungarian Method.

3.4.2.5 MOP Computation

(a)Kinematics Accuracy

Two parameters were selected to describe the kinematics accuracy namely the position estimation error and the standard deviation of the position error.

Each of these measures is estimated at two distinct levels. One is the local estimation in which these measures are computed for each individual Truth Entity for each time point. The second level aggregates these measures over all the Truth Entities but still computed independently for each time point.

(b)Detection Performance

The Detection Process Performance is estimated in form of probability of False Track and probability of Missed Truth.

Simply put the probability of False Track is the percentage of Tracks that have been declared as being "False" (i.e. they did not represent any Truth entity). Thus the *Probability of False Track* is computed as the number of Tracks not associated with any Truth. This measure is cumulated over all the time points.

The probability of Missed Track is the percentage of Truth Entities that have been not been represented by any valid Track. This measure is also cumulated over all the time points.

(c) Association Performance

The Association Performance is described in terms of Track Purity and Track Switches.

Track Purity gives an estimate of consistency with which a given Target was represented by the same Track. This measure is cumulated over all targets and all time points.

Track Switches is a measure of Track unfaithfulness in consistently representing the same Target. Said in other words it captures the frequency with which the Track switches over from representing one Target to representing another Target. This measure is cumulated over all targets and all time points.

3.5 PE Module Optimization

The final level determines the detailed design of the solution "patterns" for each sub function of each node in the fusion tree.

3.5.1 PE Module Requirement Analysis

The requirements for the PE Module are basically the design specifications from the previous level.

3.5.2 PE Module Design

In this phase solutions are developed for performing each function of every PE Node based on the requirement specifications.

3.5.2.1 Hypothesis Evaluation

For evaluating the Track –Truth association hypothesis we develop a scoring scheme based on the Max a Priori Scoring. The Max a Priori Deterministic data association technique is a popular method for correlating Sensor measurements with Tracks during the Data Association stage. At any one point, the overall MAP hypothesis score is the product of three MAP individual scores, which are explained subsequently.

As mentioned earlier in this Case Study we are concerned about the following three hypotheses

- Track-Truth association hypotheses
- False Track Hypothesis
- Missed Truth Hypothesis

In the following subsections we present the MAP scoring along with its adaptation for our purpose for the above-mentioned hypothesis.

Kinematics Scoring for Association Hypothesis

The association hypothesis kinematic scoring for a new incoming sensor mesurement, y(S) to an existing Track, y(T) assumes a multivariate Gaussian distribution [ellipsoid], with a central Track covariance P which models the error in the Track location due to possible motion. Then the kinematics score for Measurement to Track association hypothesis 'h'is computed as follows:

$$f(h) = \{1/ (2\pi)^{d/2}\} \ \{|V|^{1/2}\} \exp[-1/2\{I^T V^{-1} I\}] \qquad (3.1)$$

where

- y(S) are the sensor measurment Gaussian kinematics with covariance R,
- y(T) are the Track Gaussian kinematics with covariance P,
- h is the hypothesis that the Sensor measurement and Track are associated,
- d is the dimension of the Gaussian kinematics state,
- $|V|$ is the determinant of the innovations covariance, $V=[\ \phi P \phi^T + Q] + R$,
- ϕ is state transition matrix, Q is the noise covariance, and the measurement matrix, h, is the identity,
- I is the innovations vector, $I = y(S) - y(T)$.

When all the covariances have a constant dimension the first term becomes constant through out and hence it can be dropped. Thus we have a simplified Kinematics score equation.

$$f(h) = \{\,|V|^{1/2}\,\}\ \exp[-1/2\{I^T V^{-1} I\}] \tag{3.2}$$

Now we are concerned with associating Truth with Tracks. Here instead of y(S) i.e. the Sensor measurement we have the Truth data. As a result we do not have the sensor noise covariance "Q" and the sensor measurement kinematics covariance "R". The state transition matrix "ϕ" can also be ignored since the "lag" between time of Track update and time of measurement is negligible. So the kinematic score equation for Track to Truth association hypothesis 'h' can be described as follows.

$$f(h) = \{\,|V|^{1/2}\,\}\ \exp[-1/2\{I^T V^{-1} I\}] \tag{3.3}$$

where

- y(S) are the Truth Data,
- y(T) are the Track Gaussian kinematics with covariance P,
- h is the hypothesis that the report and Track are associated,
- $|V|$ is the determinant of the innovations covariance, $V=P$
- I is the innovations vector, $I = \mathrm{y}(S) - \mathrm{y}(T)$.

Kinematics Scoring for non-Association Hypothesis involving Track data

The approach to computing kinematics score for hypothesis involving only the Track data and not the Sensor measurements is somewhat different from Equation 3.3 since for these hypothesis y(S) does not exist. Here the chi-square statistic (i.e., $\{I^T V^{1} I\}$) is replaced with its mean, μ. Namely,

- $\mu = .455$ for 1 degree of freedom (DOF) (e.g., bearings-only)
- $\mu = 1.39$ for 2 DOF (e.g., x and y)
- $\mu = 2.37$ for 3 DOF (e.g., Cartesian (x, y, z))
- $\mu = 3.36$ for 4 DOF (e.g., 2 dimensions with rates)
- $\mu = 4.35$ for 5 DOF
- $\mu = 5.35$ for 6 DOF (e.g., Cartesian (x, y, z) with rates)

Thus the kinematics scoring equation for non-Association Hypothesis involving Track data such as False Track hypothesis is as follows,

$$f(h) = |V|^{1/2}\ \mu \qquad (3.4)$$

This equation can be adopted as it is for our purpose.

Kinematics Scoring for non-Association Hypothesis involving Truth data

For non association hypothesis involving only the Truth Data the terms y(*T*) and *V* are not available in Equation 3.3. The only available data in this regard is the Truth Kinematics information, which does not provide any insight into non-Association hypothesis such as Missed Truth hypothesis etc. Hence we can ignore the kinematics score term for such a hypothesis.

Parametric/Attribute Association Scoring

Often tracking algorithms incorporate object classification techniques for more accurate data association. Object classification is based on the object attributes measured by the various sensors. For example a stealth bomber would have a different range of speed, maneuverability, thermal emissions etc as compared to an anti aircraft missile. The parametric scoring is computed as the product of commensurate attributes and non-commensurate attributes.

However the tracking scenario developed for our case-study implementation does not incorporate object classification techniques for data association. Hence we ignore this second term in the MAP scoring equation.

A Priori Association Hypothesis Scoring

This part of the hypothesis score nominates apriori probabilities for the given hypothesis. The apriori probability is computed using the knowledge about the SUT (i.e. the sensors and the tracking algorithm characteristic) and the scenario characteristic. Some of these characteristics are given below:

- Probability of detection and false alarm statistics
- Object birth and death statistics
- Sensor scan rate
- Source field-of-view, operating mode, and conditions
- A priori scene descriptors and probability of redetection

For computing the apriori probabilities for the above three hypotheses, the detection and false alarm statistics were deemed to be sufficient. For this purpose we define the following probabilities

- Pd(S)-The probability that the given Truth is represented by a valid Track. New Truth arrival statistics and sensor coverage statistics are used to determine the apriori value of this probability. However when available the value of this term could be updated dynamically with the help of estimates obtained from the PE System itself.

- Pfa(T) – The probability that the given Track is a false Track. Track death statistics and sensor false alarm statistics are used to determine the apriori value of this probability. However when available the value of this term could be updated dynamically with the help of estimates obtained from the PE System itself.

Now that we have defined Pd(S) and Pfa(T) let us see how the apriori probabilities are computed for the 3 association hypotheses.

- <u>Track-Truth association hypotheses:</u> This hypothesis requires computing the probability that the given Track was caused by the given Truth. For this the given Track needs to be a valid Track which is computed as 1 minus the Pfa(T). This probability multiplied by probability that the given Track is represented by some valid Track Pd(S) gives us the apriori probability for the association hypothesis.

  Thus we have,

  P(association) = [1- Pfa(T)] x Pd(S) (3.5)

- <u>False Track Hypothesis</u>: For this one needs to compute the probability that the given Track is invalid i.e. it does not represent any Truth Entity. Simply put this is same as Pfa(T). Thus

  P(False Track) = Pfa(T) (3.6)

- Missed Truth Hypothesis: This hypothesis requires computing the probability that the given Truth is not represented by a valid Track which is 1 minus Pd(S). Thus

$$P(\text{Missed Truth}) = 1 - Pd(S) \qquad (3.7)$$

The Hypothesis Scoring Equations

3. Track – Truth Association Hypothesis

This consists of two parts the kinematics score (Equation 3.3) and the apriori score (Equation 3.5), which add up to give the following equation.

$$f(h) = \{|V|^{1/2}\} \exp[-1/2\{I^T V^{-1} I\}] [1 - Pfa(T)] Pd(S)$$

2. False Track Score

This consists of two parts the kinematics score (Equation 3.4) and the apriori score (Equation 3.6), which add up to give the following equation.

$$f(h) = \{|V|^{1/2}\} \mu \; Pfa(T)$$

3. Missed Truth Score

As we saw earlier the kinematics score for this hypothesis in not available. So the scoring equation for this hypothesis has only the apriori score (Equation 3.7), which is as follows.

$$f(h) = \quad 1 - Pd(S)$$

Looking at the SUT characteristics and the scenarios developed for the Case Study following values were nominated for the constants in the above three hypothesis scoring equations

- $Pd(S)=0.95$
- $Pfa(T)=0.03$
- $\mu= 4$

3.5.2.2 Hypothesis Selection

The Hungarian Algorithm was nominated for the HS phase for all the 3 Node designs. The Hungarian Method procedure is explained as follows in Table 3.1.

Step #	Action
1	If the minimum element in row 'i' is not 0, then subtract this minimum element from each element in row i.
2	If the minimum element in column 'j' is not 0, then subtract this minimum element from each element in column 'j'.

3	Examine rows successively, beginning with row 1, from a row with exactly one unmarked zero. If at least one exists, mark this zero with a star '*' sign to denote assignment. Mark 'X' on the other zeroes in the same column so additional assignments will not be made to that column. Repeat the process until each row has no unmarked zeros or at least two unmarked zeroes.
4	Examine columns successively for single, unmarked zeroes and them with a star '*' sign to denote assignment. Mark 'X' on the other zeroes in the same row so additional assignments will not be made to that row. Repeat the process until each column has no unmarked zeros or at least two unmarked zeroes.
5 (a) (b) (c)	Repeat steps 3 and 4 until one of the following occurs ▪ Every row has an assignment '*'. ▪ There are at least two unmarked zeroes in each row and each column. ▪ There are no zeroes left unmarked and a complete assignment has not been made.
6	If 5(a) occurs, then the assignment is complete and it is an optimal assignment. If 5(b) occurs, arbitrarily make an assignment '*' to one of the zeroes and mark 'X' on rest of the unmarked zeroes in the same row and column, and then proceed to step3. If 5(c) occurs, go to step 7.
7	Check (√) all rows for which an assignment '*' has not been made.
8	Check (√) columns not already checked that have a zero in the checked rows.
9	Check (√) all rows not already checked that have assignments made in the checked columns.

10	Repeat steps 8 and 9 until the chain of checking ends.
11	Draw lines through all unchecked rows and through all checked columns. This will give the minimal cover.
12	Examine the elements that do not have at least one line through them. Select the smallest of these and subtract it from every element in each row that contains at least one uncovered element. Add the same element to every element in each column that has a vertical line through it. Return to step3.

Table 3.1 The Hungarian Algorithm Procedure

3.5.2.3 MOP Estimation

The definition of the MOPs nominated at the Node design level is as follows:

Position Error

Position Error for Truth Entity 'i', for the given time point is computed as the Euclidean distance between the true location of the Truth Entity at that time point and the estimated location of the Track (which is representing this Truth Entity at that time point). This metric is useful in analyzing how the SUT performance with reference to each Truth Entity.

If (x1,y1) is the true location of Truth Entity 'i' at time 't' and (x2,y2) is the estimate location of Track 'j' (which is representing 'i' at time 't') the Position Error is computed as follows

$$P_{error}(i) = \{(x1 - x2)^2 + (y1-y2)^2\}^{1/2}$$

If at any time point a given Truth Entity is declared as "Missed Truth" then this metric is not computed for that particular Truth Entity for that time point.

Average Position Error

Average Position Error for a given time point is the Average of Position Errors for all the Truth Entities that existed at the time point. This is computed as

$$P_{error\_avg}(t) = 1/n \sum_{i=1}^{n} P_{error}(i)$$

This metric gives overall perspective as to how the SUT fared in its task of estimating location of the Truth Entities in the Field of View.

Track File Probability of detection (Pd)

Track File Probability of Detection for a given time point is defined as the number of Truth Entities represented by valid Tracks divided by the total number of Truth Entities existing at that time point.

$$Pd = \frac{\text{\# Truth Entities represented by valid Tracks at time "t"}}{\text{Total \# of Truth entities at time "t"}}$$

Probability of False Track (Pf)

Probability of False Track for a given time point is computed as the number of Tracks not associated with any Truth for the give time point divided by the number of Tracks at that time point.

$$Pf = \frac{\text{\# of unassociated Track at time “t”}}{\text{Total \# of Truth entities cumulated at time “t”}}$$

Track Purity (Tp)

Track purity for Truth Entity ‘i’ is computed as the ratio of number of associations of Truth Entity ‘i’ with Track ‘j’ to the total number of associations for Truth Entity ‘i’ with any valid Track (where ‘j’ has been associated with ‘i’ more than any other Track). This is the only metric in this Case Study, which is computed over the complete scenario.

$$Tp(i,j) = \frac{\text{\# of associations for Truth “i” with a valid Track ‘j” over the whole scenario}}{\text{Total \# of associations for Truth “i” with any valid Track over the whole scenario}}$$

$$Tp(i) = \max Tp(i,j) \ \forall \text{ “j”}$$

Track Switches (Ts)

Track Switches for a given time *'t'* is computed as the number of Truth Entities assigned to a different Track as compared to their assignment at time *'t-1'* divided by the total number of Truth Entities existing at time point 't'. So for example if Truth Entity 'i' was being represented by Track 'j' at time 't' and at time 't+1' Track 'k' represents Truth Entity 'i', then it is counted as a "switch". However missed Truth is not counted as a switch.

$$Ts = \frac{\text{\# of Switches at time 't'}}{\text{Total \# of Truth entities at time "t"}}$$

Chapter 4

Case Study Results and Analysis

4.1 Introduction

As discussed previously two set of scenarios were implemented in this Case Study. In the following sections we described the results of these scenario sets.

4.2 Feigned Crossing Target Scenario

The Feigned Crossing Target Scenario was designed to demonstrate effects of a simple one time switching on the PE process. Figure 4.1 displays the trajectory of two targets, which come very close to each other and then move apart. The red oval shown in the Figure 4.1 marks the zone in which there is high probability of association complexities. The targets were flying at a constant velocity of 500m/sec. The scenario consisted of a single ground based platform, which had 2 sensors (Radars) mounted on it. The details of the sensors are presented in Table 4.1. The Fusion system details are given in Table 4.2.

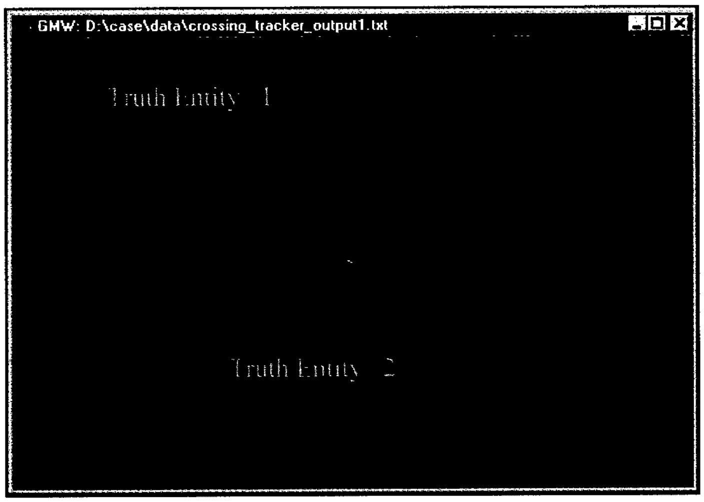


Figure 4.1 "Feigned" Crossing Target Scenario

		Scan RPM	Probability of detection	False Alarms
Platform 1	Sensor 1	60	0.9	No
(Stationary)	Sensor 2	40	0.95	No

Table 4.1 Sensor Details

Filter	IMM Filter
Gating	Ellipsoidal Gating
Gating criteria	Probabilistic
Assignment algorithm	JVC
Fusion Architecture	Contact Fusion

Table 4.2 Tracker Details

The tracker output for this scenario is shown in Figure 4.2. As shown in the figure initially measurements generated by Truth-1 were used to update Track – 'A' and measurements generated by Truth-2 were used to update Track – 'B'. However when the two Truth Entities come very close to each other, the Tracker got confused, dropping Tracks 'A' and 'B', and initiating new tracks for a next few updates. After that the Tracker reinitialized Track –'B' but failed to reinitialize Track – 'A'. However now Track 'B' was updated from the measurements generated by Truth-1. Thus a "switch" occurred during Data Association. A new Track, Track –'C' was initiated and it got updated from the measurements generated by Truth-2 till the end of the scenario.

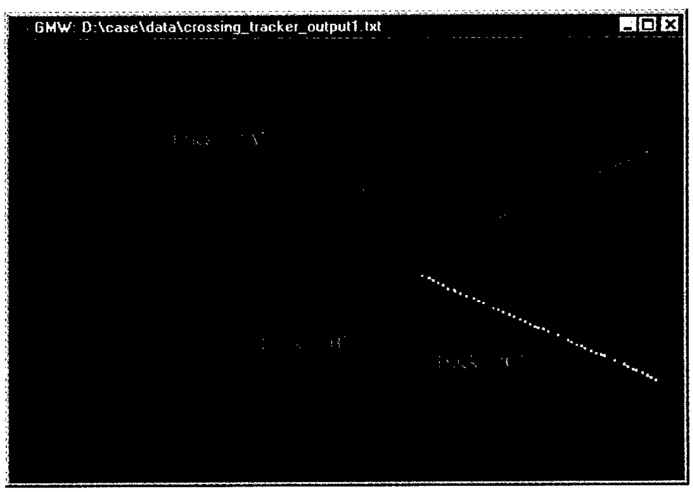


Figure 4.2 "Feigned" Crossing Target Scenario Tracker Output

The performance of the SUT under this scenario was tested using the Case Study PE system implementation. The results and analysis of this testing is presented in the following subsections.

4.2.1 Position Error Estimation

The Position Error Estimation plots for Truth Entity-1 are shown in Figure 4.3 and those for Truth Entity-2 are shown in Figure 4.4. Each of these plots shows Position Error Estimates for the Switching Node and the No-Switch Node. In these plots Position Error is shown to be zero for those time points for which the Truth Entity has been declared as Missed.

As seen in Figure 4.3 for the first half of the scenario, both the Switching Node and the No-Switch Node report same values for Position Error estimates most of the time. But after time point "50 seconds", the No-Switch Node reports a value of "0" for Position Error estimates – implying that it could not associate the Truth Entity-1 with any valid Tracks. Whereas the Switching Node continued to churn estimates for Position Error with some intermittent hiccups during the "confusion zone".

Similar behavior is echoed in the Position Error plot for Truth Entity –2. Here the No-Switch Node could not associate the Truth Entity-2 with any valid Tracks for the first half of the scenario.

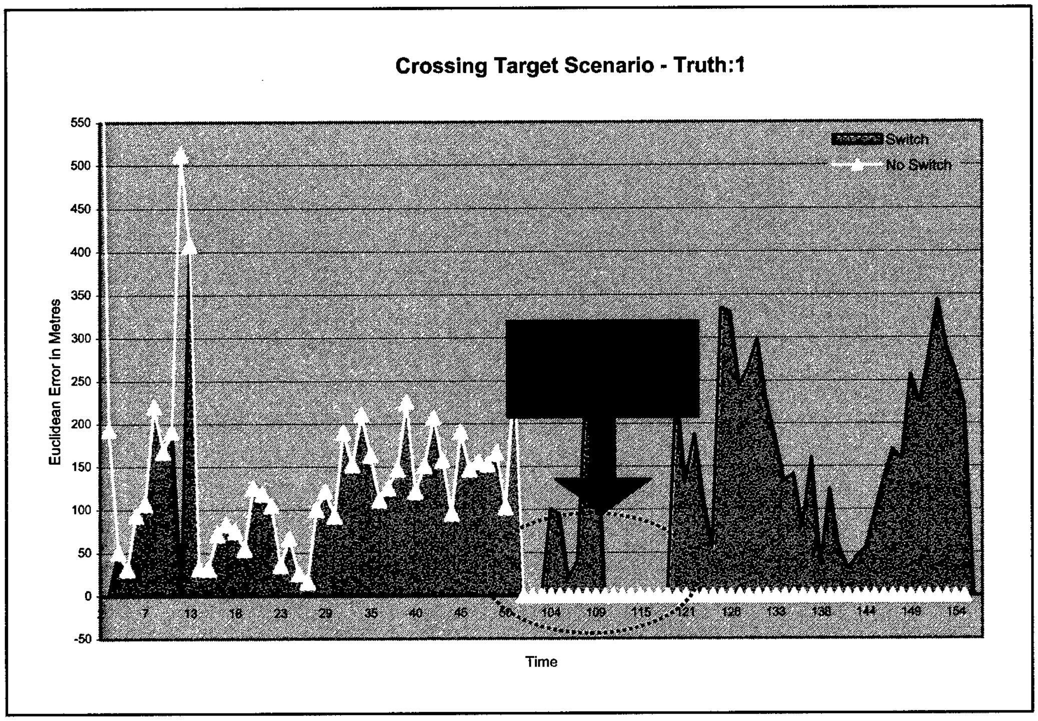


Figure 4.3 Position Error Estimates for Truth Entity-1

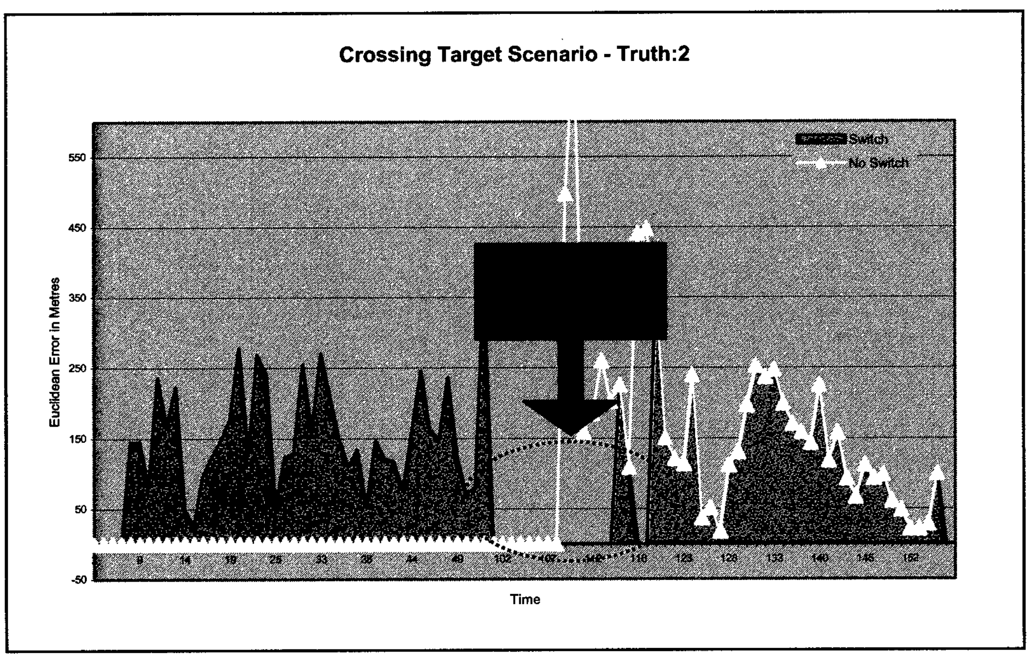


Figure 4.4 Position Error Estimates for Truth Entity-2

The Probability of Detection plot for the two Nodes is shown in Figure 4.5. As shown in the figure, the PD for the Switching Node has value of 1.0 most of the time while the PD for the No-Switch Node has value of 0.5. This perfectly matches with our above analysis.

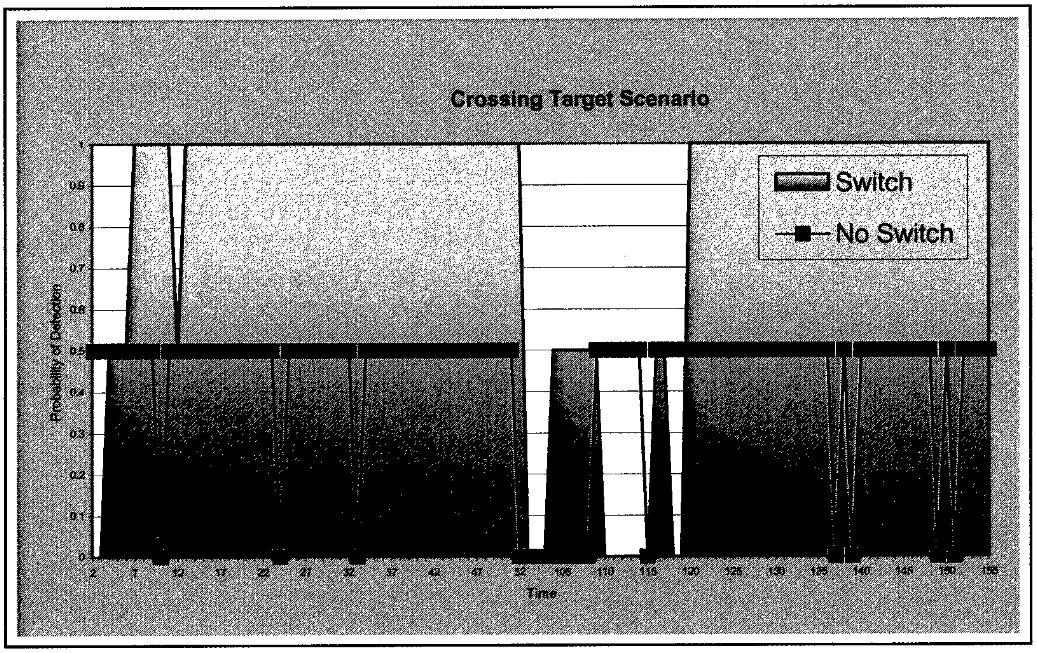


Figure 4.5 Probability of Detection

Now let us see what was the exact outcome of Track-Truth Association process for the two PE Nodes. Figure 4.5 displays the Track-Truth Association result for Truth Entity-1 by the No-Switch Node As one can see here the Truth Entity –1 was represented by Track –A. However since the lifespan of Track –A was only the first half of the scenario, Truth Entity –1 was declared as Missed Truth for the second half of the scenario (since the No-Switch policy does not allow representation of a Truth Entity by more than one Track throughout the life span of the Truth entity).

Now Figure 4.6 displays the Track-Truth Association result for Truth Entity-1 by the Switching Node. Here just like the No-Switch Node, Track –A represented the Truth Entity-1 for the first half of the scenario. During the second half of the scenario Track – C represented the Truth Entity-1. In the "confusion zone" where the two Tracks A and B did not exist, the Switching Node represented Truth Entity –1 by the residual Tracks.

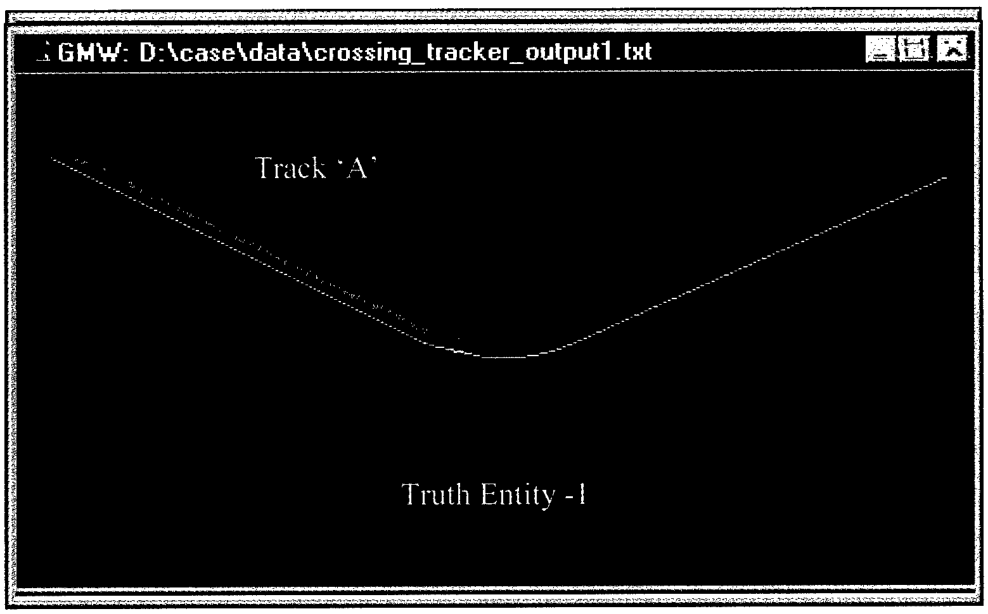


Figure 4.5 Truth Entity-1 representation by No-Switch Node

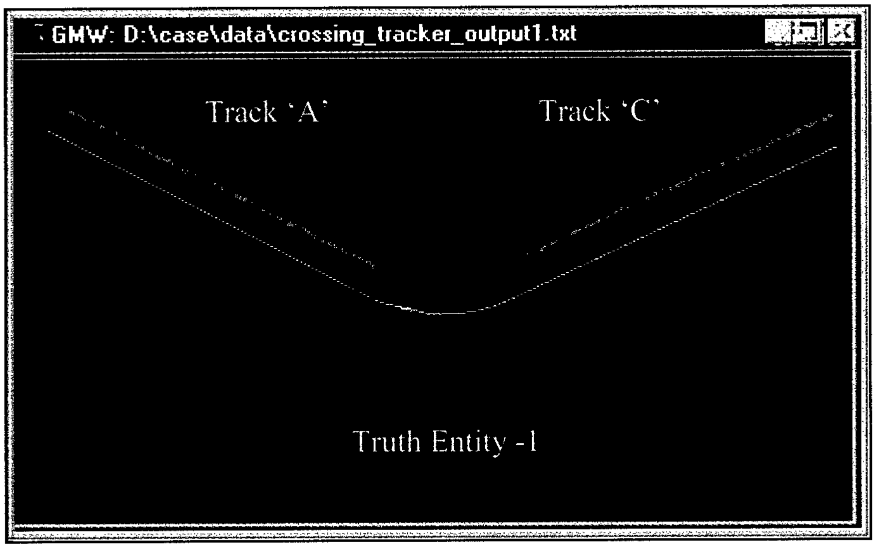


Figure 4.6 Truth Entity-1 representation by Switch Node

Thus the outcome of the Track-Truth Association process for the two switching policies is as expected. So going by the results of the No-Switch Node one would declare the performance of the SUT as bad whereas judging by the results of the Switching Node one could call the SUT performance as consistent and fair.

4.3 Maneuvering Targets Scenarios

A second set of scenarios consisting of three scenarios was created. The main difference across this set of scenarios was the inter target spacing. Accordingly the three scenarios have been named as the Low Spacing scenario in which the targets were quite close to each other, Medium Spacing scenario in which the targets were somewhat close and the High Spacing scenario in which the targets were quite apart. The driving factor behind this set of scenarios was to study the effect of the relationship between inter-target spacing and the Track-Truth Association policy on the estimation process. This scenario set would also be used to demonstrate the use of the estimates obtained from the PE System for analysis.

Figure 4.7 displays the basic set of these scenarios. As shown in the figure the scenario consists of four sets of Targets performing high-speed maneuvers. Each scenario was partitioned into 3 zones. Zones 1 and 3 are the ones in which the Targets are moving in a straight line and at a large distance from the sensors. Zone –2 is the area in which the Targets are maneuvering but close to the sensors.

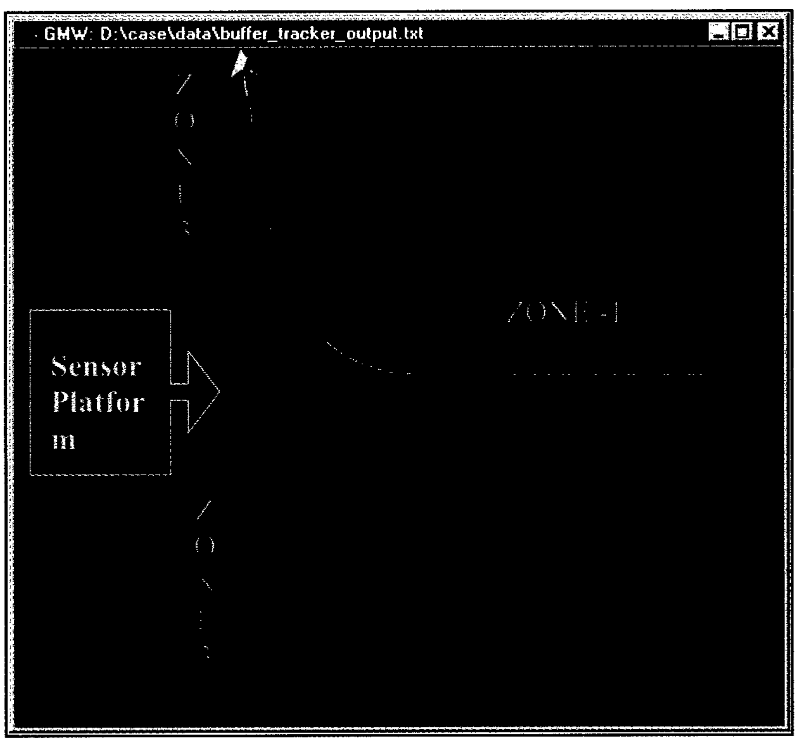


Figure 4.7 Generic Trajectories for the Maneuvering Targets Scenarios

In table 4.3 we present the Average Position Error and Standard Deviation of Position Error for Truth Entity-1 for each of the scenarios as estimated by different PE Nodes.

Scenario	Mean Position Error			Standard Deviation of Position Error		
	Switch	Window	No Switch	Switch	Window	No Switch
Low Spacing	229.58	251.16	287.39	160.12	164.55	184.87
Medium Spacing	219.37	224.24	264.40	144.70	146.28	176.70
Large Spacing	212.58	230.83	231.25	155.41	156.82	162.87

Table 4.3 Error trends across the maneuvering targets scenarios

Following trends appear from Table 4.3

- In all the three scenarios the Switching Node gives the lowest estimate of Mean Position Error and the Standard Deviation for the Position Error is also the lowest. This is obvious since the Switching Node always optimizes locally (i.e. for given time instance) unlike the other two Nodes.
- The Window Node (or the Restricted Switching Node) generates estimates better than the No Switch Node but worse than Switching Node. This is because the Window Node attempts to reduce switching and in doing so it does not always achieve local optimum.
- The estimates from the 3 PE Nodes appear to be converging across the three scenarios. In other words the difference between the estimates from the three Nodes is very high in the Low spacing scenario, the difference lowers in the Medium Spacing scenario and in the Large Spacing scenario the difference is the lowest. This because of decrease in Association ambiguities with increase in

Inter-Target spacing. This can also be concluded from the Table 4.4 that shows the Switching trends across the three scenarios. One can see that the number of switches reduces for both the Switch and the Window Node. In short the Track-Truth Association result from the Switch Node and the Window Node tends to be the same as that from the No Switch Node with reduction in Association Ambiguities.

Scenario	Number of Switches per scenario run	
	Switch	Window
Low Spacing	122	85
Medium Spacing	72	52
Large Spacing	18	13

Table 4.4 Switching trends across the maneuvering targets scenarios

Now let us compare the performance of the three PE Nodes in each of the three zones across the scenario set. Tables 4.5, 4.6, and 4.7 display the Average Position Error for each of the three zones.

Scenario	Average Position Error		
	Switch	Window	No Switch
Low Spacing	123.23	162.49	180.92
Medium Spacing	119.1	152.70	177.23
Large Spacing	118.85	131.02	151.22

Table 4.5 Error trends across the maneuvering targets scenarios – Zone 1

Scenario	Average Position Error		
	Switch	Window	No Switch
Low Spacing	67.17	77.76	83.06
Medium Spacing	63.64	66.703	80.22
Large Spacing	70.92	94.60	116.04

Table 4.6 Error trends across the maneuvering targets scenarios – Zone 2

Scenario	Average Position Error		
	Switch	Window	No Switch
Low Spacing	329.06	333.27	444.93
Medium Spacing	281.67	299.206	322.30
Large Spacing	312.33	318.25	320.25

Table 4.7 Error trends across the maneuvering targets scenarios – Zone 3

Looking at Table 4.5 and 4.6 one can find no deviation from the above conclusions. However the results in Table 4.7 appear to be somewhat deviating. If one observes only the Switch and the No Switch column then one can easily say that the above results still hold good. It is only the Window Node data that appears to be "hazy". In the low spacing zone the Window Node estimates are quite close to the Switch Node estimates. While in the medium spacing zone the Window Node estimates are distant from the Switch Node estimates and in the large spacing zone the Window Node estimates are actually close to the from the No Switch Node estimates –why so? The reason behind this behavior of the Window Node is that it tries to minimize the effect of random jumps in the data sets. But if there are no such jumps in the data set then the results of Window Node tend to be same as that of the Switching Node.

4.4 SUT Performance Analysis

Now let us see how one would go about analyzing the performance of the SUT using this PE System. Figures 4.7 - 4.13 display the plot of Position Error for Truth Entity-1 for each of the scenarios for each of the PE Nodes.

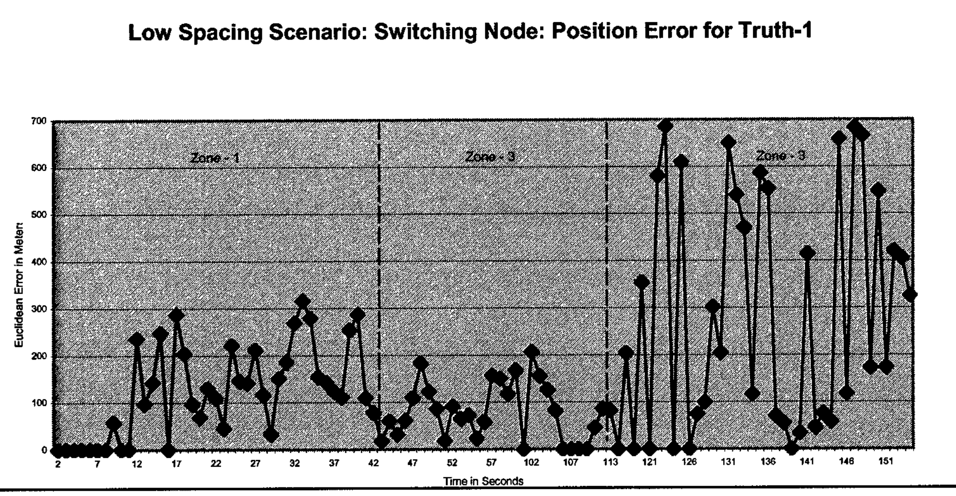


Figure 4.8 Low Spacing Scenario Switching Node Results

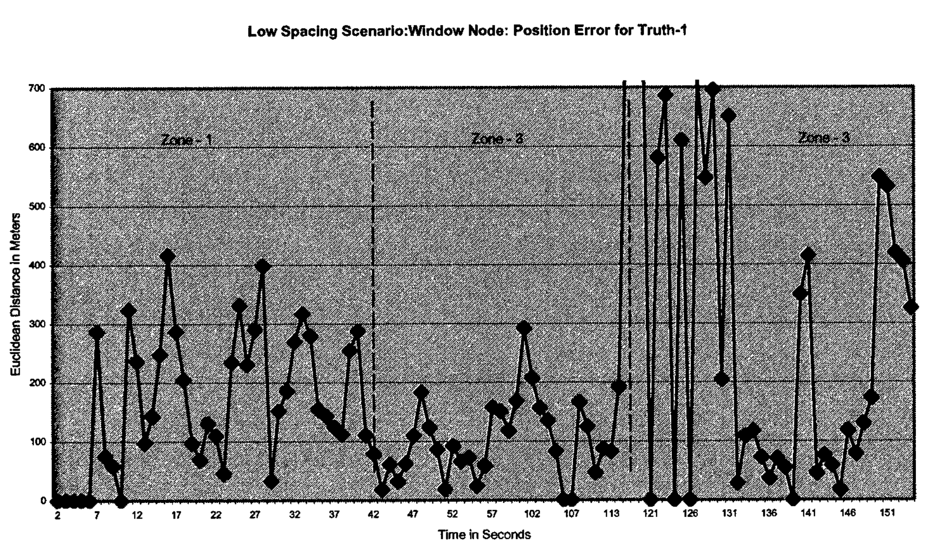


Figure 4.9 Low Spacing Scenario Window Node Results

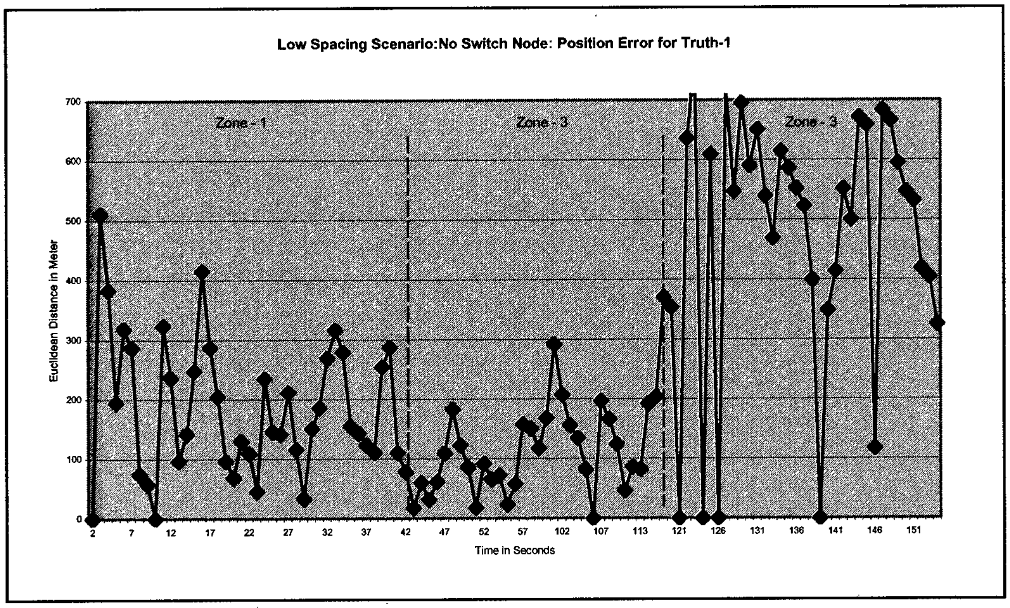


Figure 4.10 Low Spacing Scenario No Switch Node Results

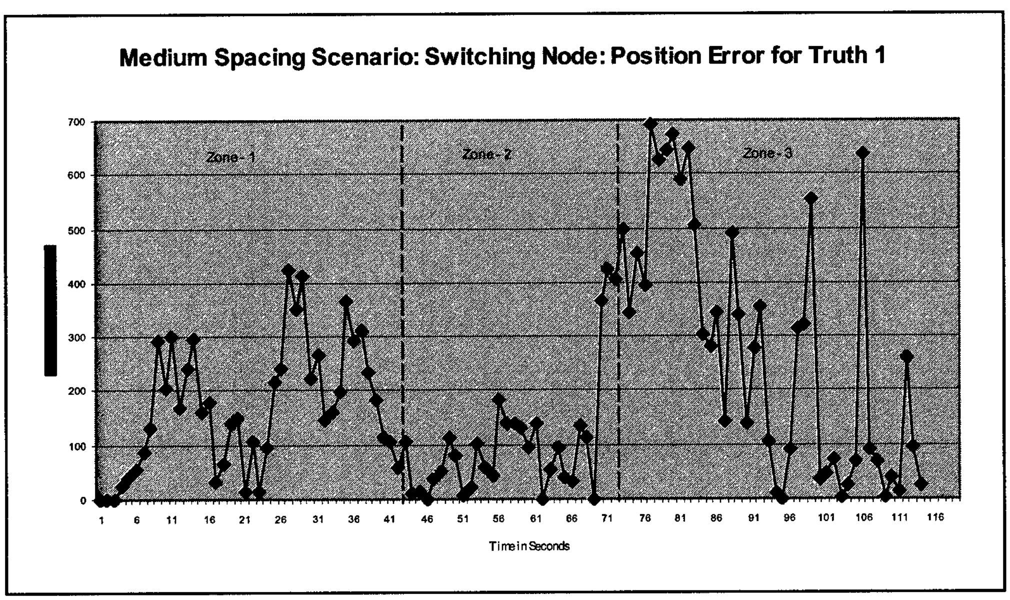


Figure 4.11 Medium Spacing Scenario - Switching Node Results

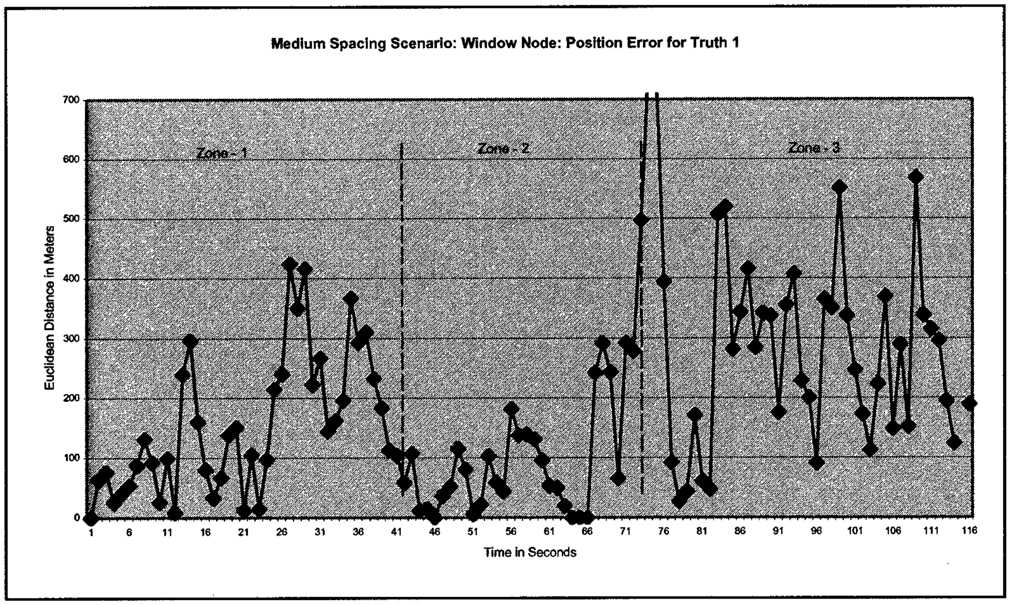


Figure 4.12 Medium Spacing Scenario - Window Node Results

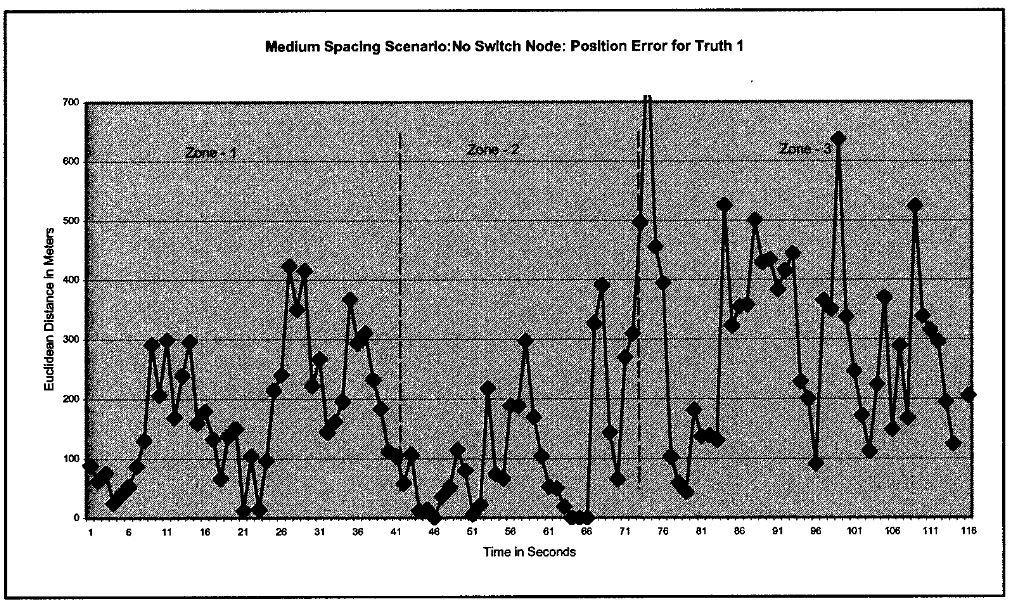


Figure 4.13 Medium Spacing Scenario - Switching Node Results

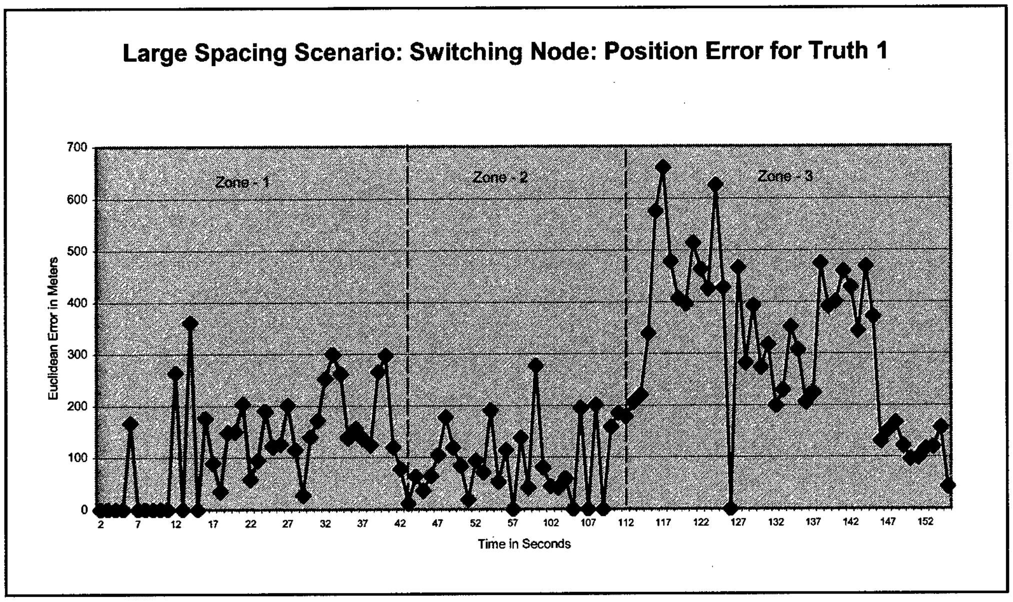


Figure 4.14 Large Spacing Scenario - Switching Node Results

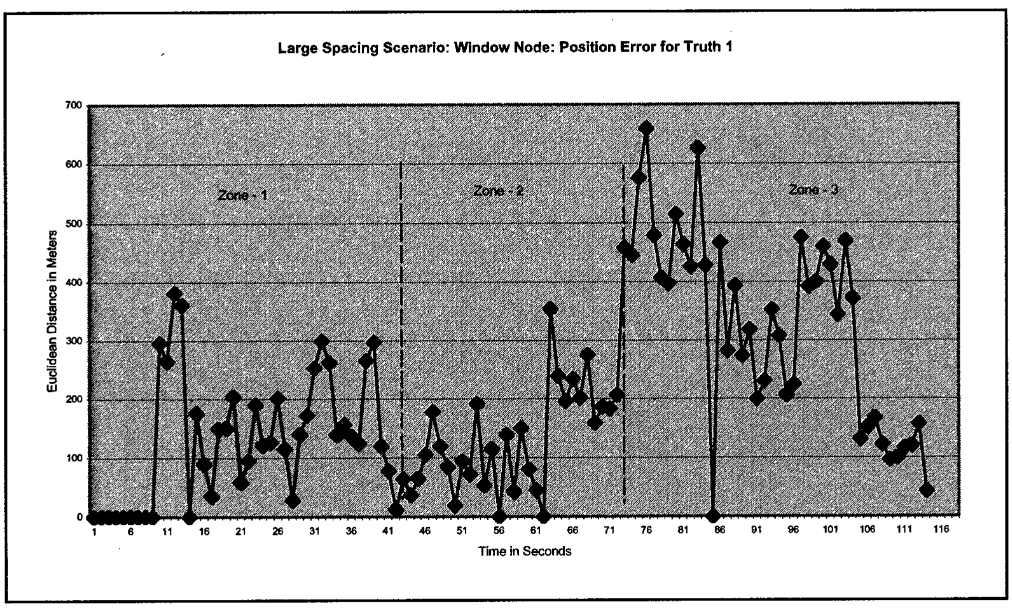


Figure 4.15 Large Spacing Scenario - Window Node Results

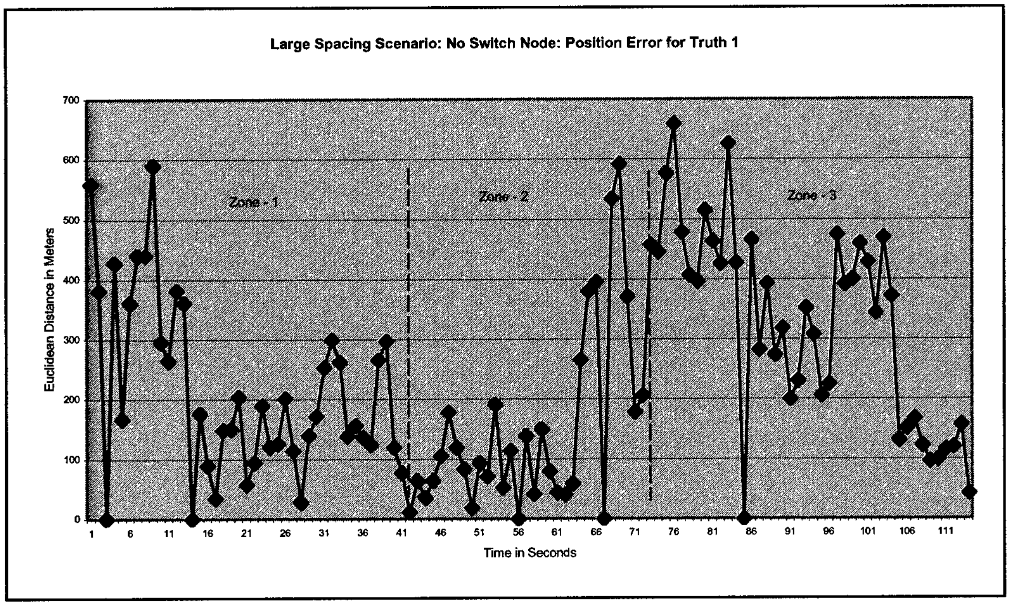


Figure 4.16 Large Spacing Scenario – No Switch Node Results

We choose to display the results for the individual target rather than the average over all the targets because that would hide away some of the trends. Looking at the above plots we observe that the error is generally very high in zone-3 while it is the lowest in zone-2. This is same as the trends observed in the Tables 4.5,4.6 and 4.7. Low estimation errors in zone-2 make perfect sense since this zone is very close to the sensor platform. However the positioning of zone-1 and zone-3 with reference to the sensor platform is very much similar. Then the question arises – why such a huge disparity in the estimation errors in the two zones? This question is further aggravated by the fact that in zone-3 the average distance between the targets was more than that in zone-1. This means that in zone –3 the main source of errors should be the sensing process and not the association process. Keeping this in mind we tried to tweak the SUT to find the reason behind this behavior. After several different simulations and analysis we still were unable to fix that error.

However this was a good example of how to use the PE System.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

Hitherto not much work has been done in the field of Performance Evaluation of Multi Target Tracking Systems. In this thesis we presented a Frame Work for Performance Evaluation System Design. As a part of this Frame Work we also discussed the fundamental issues encountered during the evaluation process and the solution approaches to these issues. This was followed by a Case Study implementation of the Frame Work. Through this Case Study we successfully demonstrated the following

- A proof on concept for the Framework.
- Use of advance scoring techniques (for PE), which are now finding their way into real operational Data Fusion-based tracking.
- Through the "Feigned Crossing Target Scenario" we presented a simple proof for the argument that "the No-Switch policy is not necessarily the best way to judge a Tracking System"

- Solution approaches to the Track-Truth Association problem.
- We observed that inter target spacing creates ambiguities in the Data Association process which are subsequently inherited by the Track-Truth Association process.
- The Switching policy and the Window policy yield better estimates of performance metrics when the inter target spacing is low. As the inter target spacing increases the estimates of the Switching policy and the Window converge towards the estimates of the No-Switch policy. These results are in full agreement with our expectations.

In short the Performance Evaluation System Design Frame Work provides excellent guidelines for evaluating, comparing and analyzing Tracking Systems and their components. Through this Frame Work we have laid the foundation for formalizing the field of Performance Evaluation in the Tracking community.

5.2 Recommendations

This thesis has given us insight into the Performance Evaluation problem and some of the solution approaches. However much work needs to be done before this field matures. The future work should concentrate on the areas:

- Understanding the interdependencies between the Track-Truth Association policy and the sophistication of the MTTS is the key to achieving confidence in the evaluation process.

- Impact of PE Tree Design on the Track-Truth Association and subsequent process of metric estimation needs to be studied.

- In the current Case Study the hypothesis score computation process used basic kinematics information only. Use of Attribute information and Identity information for hypothesis scoring needs to be explored.

Bibliography:

1. Maybeck, Peter S., "Stochastic Models, Estimation, and Control, Vol.1", 1979, Academic Press, Inc.
2. Gelb, A., "Applied Optimal Estimation", 1979, MIT Press.
3. Bowman, C. L., and Morefield, C. L., "Multi Sensor Fusion of Target Attributes and Kinematics Reports", 3rd ONR/MIT conference, June 1980.
4. Llinas, J., "Perspectives on the Evaluation of Fusion-Based Distributed Tracking System", November 2000.
5. Zheng, H., Farooq, M., and Main, R., "Mulitarget Tracking Algorithm Performance Evaluation", SPIE Aerosense Conference, Sensor Fusion, 3719, 300-310, 1999.
6. Rpy, J., and Bosse, E., "Definition of a Performance Evaluation Methodology for Sensor Data Fusion Systems", DREV Report, September 1995.
7. Desbois, M., "Methods of Evaluating Multi-Sensor Tracking Performance", Fusion 1998.
8. Sea, R. G., "An efficient suboptimal decision procedure for associating sensor data with stored tracks in real time surveillance system", Proceedings IEEE conference on Decision and Control, Miami, December 1971.
9. Singer, A. and Sea, R. G., "New results in optimizing surveillance system tracking and data correlation performance in dense multi target environments", IEEE transactions on Automatic Control, Vol. AC-18 No.6, December 1973.
10. Blackman, S. and Popoli, R., "Design and Analysis of Modern Tracking", ArtechHouse, 2000.

11. Woff, T., and Fixsen, D., "Standards, Benchmarks, and Monte Carlo: Evaluating Multi-Sensor Fusion Systems", Technical Report.

12. Drummond, O. E., and Fridling, B. E. , "Ambiguities in Evaluating Performance of Multiple Target Tracking Algorithm," Signal and Data Processing of Small Targets 1992, Proc. SPIE, Vol. 1698.

13. Drummond, O. E., and Fridling, B. E. , "Performance Evaluation for Multiple Target Tracking Algorithm," Signal and Data Processing of Small Targets 1991, Proc. SPIE, Vol. 1481.

14. Drummond, O. E., "Methodlogies for Performance Evaluation of Multiple Target Multi Sensor Tracking," Signal and Data Processing of Small Targets 1999, Proc. SPIE, Vol. 3908.

15. Drummond, O. E., and Rothrock, R. L., "Performance Metrics for Multiple Target Multi Sensor Tracking ," Signal and Data Processing of Small Targets 2000, Proc. SPIE, Vol. 4048.

16. Kovacich, M., and Chong, C., "Definition of Track Purity", Proceedings of the SDI Panels on Tracking, Issue No. 3/1989, July 1989.

17. Llinas, J., "Assessing the Performance of MultiSensor Fusion Systems", SPIE, Vol. 1611, 1991.

18. Bowman, C., Llinas, J., Rump, C., Hall, D. and Savas, S, "Research in Correlation Science", Report No. CMIF-5-98, October 1998.

19. Poore, A., "Multi Dimensional Assignment formulation of Data Assocaition Problems Arising from Multiple Target Multi Sensor Tracking", Computation Optimization and Applications, 1994.

20. Kuhn, HW., "The Hungarian Method for Assignment problem". Naval Res Log Quart,2,1995

21. Tomizawa, N. "On Some techniques useful for the solution of transportation problems" Network Flows 1, 173-194 1971.

22. Jonker, R. and Volgenant, A., " A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems", computing,38,1987.

23. Chen, C., "A Branch and Bound Algorithm for Multi-Target Tracking and its parallel Implementation", Proc American Control Conf , Atlanta, June 1988.

24. Hopcroft, E., and Karp, R., "A n^{5/2} algorithm for maximum matchings in bipartite graphs", SIAM Journal of Computing, 2(4):225--231, December 1973.

25. Munkres J., "Algorithms for Assignment and Transportation Problems", J SIAM 5, 1957.

26. Kaufmann, A., "Introduction a la combinatiruque en Vue des ses Applications", Dunod Pubs, Paris, 1968.

27. Bourgeois, F., and Lassalle, J., "An Extension of the Munkres algorithm for the Assignment problem to Rectangular Matrices", Comm. ACM, 14, 12, Dec 1971.

28. Bertsekas, D. P., "A Distributed Algorithm for the Assignment Problem", MIT report, 1979.

29. Castanon, D. A., "Reverse Auction Algorithms for Assignment Problems," in Algorithms for Network Flows and Matching, 1993.

30. Blackman, S., "Multiple-Target Tracking with Radar Applications", Artech House, 1986.