A VARIABLE STRUCTURE - AUTONOMOUS - INTERACTING MULTIPLE MODEL GROUND TARGET TRACKING ALGORITHM IN DENSE CLUTTER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

GÖKÇEN ALAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2013

Approval of the thesis:

A VARIABLE STRUCTURE - AUTONOMOUS - INTERACTING MULTIPLE MODEL GROUND TARGET TRACKING ALGORITHM IN DENSE CLUTTER

submitted by **GÖKÇEN ALAT** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. Kemal Leblebicioğlu
Supervisor, **Electrical and Electronics Engineering Department, METU** ________________

Examining Committee Members:

Prof. Dr. Mustafa Kuzuoğlu
Electrical and Electronics Engineering Department, METU ________________

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU ________________

Prof. Dr. Orhan Arıkan
Electrical and Electronics Engineering Department, Bilkent University ________________

Prof. Dr. Buyurman Baykal
Electrical and Electronics Engineering Department, METU ________________

Assist. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Department, METU ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: GÖKÇEN ALAT

Signature :

ABSTRACT

A VARIABLE STRUCTURE - AUTONOMOUS - INTERACTING MULTIPLE MODEL GROUND TARGET TRACKING ALGORITHM IN DENSE CLUTTER

Alat, Gökçen

Ph.D., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Kemal Leblebicioğlu

January 2013, 115 pages

Tracking of a single ground target using GMTI radar detections is considered. A Variable Structure-Autonomous- Interactive Multiple Model (VS-A-IMM) structure is developed to address challenges of ground target tracking, while maintaining an acceptable level computational complexity at the same time. The following approach is used in this thesis: Use simple tracker structures; incorporate a priori information such as topographic constraints, road maps as much as possible; use enhanced gating techniques to minimize the effect of clutter; develop methods against stop-move motion and hide motion of the target; tackle on-road/off-road transitions and junction crossings; establish measures against non-detections caused by environment. The tracker structure is derived using a composite state estimation set-up that incorporate multi models and MAP and MMSE estimations. The root mean square position and velocity error performances of the VS-A-IMM algorithm are compared with respect to the baseline IMM and the VS-IMM methods found in the literature. It is observed that the newly developed VS-A-IMM algorithm performs better than the baseline methods in realistic conditions such as on-road/off-road transitions, tunnels, stops, junction crossings, non-detections.

Keywords: Ground Target Tracking, Hide Model, Variable Structure Autonomous Interactive Multiple Model, IPDAF

ÖZ

YOĞUN PARAZİT ORTAMINDA YER HEDEF TAKİBİ İÇİN DEĞİŞKEN YAPILI - OTONOM - ETKİLEŞİMLİ ÇOKLU MODEL TEMELLİ ALGORİTMA GELİŞTİRİLMESİ

Alat, Gökçen

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu

Ocak 2013 , 115 sayfa

Bu tezde GMTI radar tespitlerini kullanarak yer hedefleri takibi gerçekleştirme problemi incelenmiştir. Değişken yapılı bir otonom etkileşimli çoklu modelleme yapısı geliştirilerek yer hedef takibinde ortaya çıkan sorunlara çözümler getirilmiştir. Aynı zamanda hesaplama maliyetinin de düşük olması sağlanmıştır. Tezde karmaşık olmayan ve etkin algoritmalar geliştirmek için şu yöntemler kullanılmıştır: i) Basit takip yapıları, ii) Topografik kısıtlar, yol haritası gibi önbilgilerin kullanılması, iii) Parazit etkisini azaltmak için iyileştirilmiş kapı açma yolları, iv) Durma-kalkma hareketi ve hedefin saklı olması durumu için yeni modeller geliştirme, v) arazi-yol geçişleri, kavşak geçişleri için iyileştirme, vi) hedefin ya da çevre koşullarının hedef tespitini engellemesi durumları için karşı tedbir geliştirme. Geliştirilen algoritma kompozit bir durum kestirimi matematiksel yapısından hareketle geliştirilmiştir. Kestirim sırasında MAP ve MMSE yöntemleri kullanılmıştır. Geliştirilen algoritmanın pozisyon ve hız kestirim hataları ölçülerek referans algoritmalarınkiyle karşılaştırılarak performans ölçümleri yapılmıştır. Geliştirilen yöntemin performansının referans yöntemlerin performansından daha iyi olduğu gözlemlenmiştir.

Anahtar Kelimeler: Yer Hedeflerini İzleme, Saklı Hedef Modeli, Değişken Yapılı Otonom Etkileşimli Çoklu Modelleme, IPDAF

To my family

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Prof. Dr. Kemal Leblebicioğlu for his patience, for his technical and spiritual guidance, and for providing me with an excellent atmosphere for doing research.

I would like to thank Prof. Dr. Orhan Arıkan for his substantial contribution to the vision of this thesis and for his encouragement.

I would like to express my deepest gratitude to Prof. Dr. Mustafa Kuzuoğlu for his invaluable comments, mentoring and support.

I would like to thank Prof. Dr. Buyurman Baykal for his support at the initial stages of my PhD study.

Last but not least, I would also like to thank my parents, sisters and elder brother for their endless love and support that also made this study possible.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

F	State transition matrix
F^h	State transition matrix for hidden model
F^{stop}	State transition matrix for stop model
F^c	State transition matrix for coordinated turn model
G	Process noise matrix
G^h	Process noise matrix for hidden model
G^{stop}	Process noise matrix for stop model
G^c	Process noise matrix for coordinated turn model
$g_{j\ell}$	Gaussian likelihood function associated with the assignment of observation ℓ to target j
H	Observation matrix
L	Number of measurements associated to tracks
w	Process noise vector
v	Measurement noise vector
R	Measurement error covariance matrix
Q	Process error covariance matrix
k	Scan number or time step
$\hat{x}_{k\|k}$	State estimate (vector) at scan k
$P_{k\|k}$	State error covariance at scan k
DP	Uncertainty in the data association
z_k^{ℓ}	Observation ℓ at scan k
$z_k^{\ell j}$	Observation ℓ for target j at scan k
z_k	A set of measurement at scan k
z^k	A sequence of all measurement sets from scan 1 to scan k
μ_k^j	Model probability for model j at scan k
$\mu_{k\|k-1}^i$	Predicted mode probability for model i at scan k
μ_k^{ij}	Mixing probability from model i to model j at scan k
P_d	Probability of detection
P_g	The probability that a correct target return will fall within the track gate

S_k	Innovation covariance at scan k
W_k	Kalman gain at scan k
$\tilde{z}_k$	Innovation at scan k
$\hat{z}_{k\|k}$	Predicted measurement at scan k
M	Number of models predetermined in the model set
M_d	Dimension of measurement vector
M_k	Number of models in effective
M_{offroad}	Number of models fixed for offroad branch
M_{road}	Number of models predetermined for road branch
N	Number of all measurements
N_k	Number of validated measurements
$\hat{N}_k$	Expected number of false alarms
Λ	Likelihood function
β	Conditional probability
c_{n_z}	Volume of a n_z dimensional unit hypersphere
σ	Standard deviation
σ_R	Standard deviation for range
σ_{Az}	Standard deviation for azimuth
σ_{stop}	Standard deviation for stop model
σ_x^2	Variance in x direction
σ_y^2	Variance in y direction
π_{ij}	Mode transition probability matrix
T	Sampling interval
T_s	Pulse repetition time
ψ	Track existence probability
$\psi_{k\|k}^o$	Track existence probability for observable target at scan k
$\psi_{k\|k}^n$	Track existence probability for non-observable target at scan k
V_{blind}	Blind velocity
V_k	Volume of the gate
$\Gamma(.)$	Gamma function
α	Probability of false track confirmation
β	Probability of true track deletion

T_1	Lower threshold
T_2	Upper threshold
n_s	Orthogonal vector
$P_{\perp}$	Orthogonal projection of state vector
P_{H_m}	Normalized probability of hypothesis m
$\acute{P}_{H_m}$	Hypothesis probability
λ	Clutter density
λ_{total}	Joint clutter density
γ	Gate threshold
λ_t	Wavelength of the transmitted pulse
$N\{\mu, P\}$	Gaussian probability density function with mean μ and covariance P
ξ_k^o	The event that the none of the observations are relevant to target
ξ_k^ℓ	The event that the observation ℓ is originated from the true target
ω	Magnitude of the velocity in coordinated turn model
ϕ	Angular tun rate in coordinated turn model
$E\{.\}$	Expected value
$(.)^T, [.]^T$	Transpose
$< ., . >$	Inner product
$P\{.\}$	Probability operator

ACRONYMS

AMM	Autonomous Multiple Model
EKF	Extended Kalman Filter
ESM	Electronic Support Measures
DTED	Digital Terrain Elevation Data
GBP1	First Order Generalized Pseudo Bayesian
GBP2	Second Order Generalized Pseudo Bayesian
GMTI	Ground Moving Target Indicator
IMM	Interacting Multiple Model
IPDA	Integrated Probabilistic Data Association
IPDAF	Integrated Probabilistic Data Association Filter
ISR	Intelligence Surveillance Reconnaissance
JIPDAF	Joint Integrated Probabilistic Data Association
JMAP	Joint Maximum A Posteriori
JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filter
KF	Kalman Filter
LLR	Log Likelihood Ratio
MAP	Maximum A Posteriori
MDV	Minimum Detected Velocity
MHT	Multiple Hypothesis Tracker
MMSE	Minimum Mean Square Error
NN	Nearest Neighbourhood
PDA	Probabilistic Data Association
PDAF	Probabilistic Data Association Filter
PF	Particle Filter
PRF	Pulse Repetition Frequency
RV	Random Variable
RMSE	Root Mean Square Error
SPRT	Sequential Probability Ratio Test

TCF	Topographic Coordinate Frame
UKF	Unscented Kalman Filter
VSIMM	Variable Structure Interacting Multiple Model
VS-A-IMM	Variable Structure - Autonomous - Interacting Multiple Model
WGS84	World Geodetic System 1984
pdf	Probability Density Function

CHAPTER 1

INTRODUCTION

Target tracking has been a challenging problem since the invention of radar systems in world war II. First radar systems were developed for detecting airborne targets [57]. Hence, target tracking literature evolved to address tracking of maneuvering airplanes and helicopters. Another genre of radars evolved in 1970s to detect ground targets in near real-time as a part of cold war intelligence, surveillance and reconnaissance requirements. Such radars are called Ground Moving Target Indicator (GMTI) radars and they have opened up a new research line regarding tracking of ground target in dense clutter environments [4, 22, 23, 24, 49].

The operating logic of GMTI radars is that they return the position of a ground moving target. They do not detect the velocity of the target and they also cannot detect targets moving slowly below a certain velocity referred to as "minimum detectable velocity" (MDV) [11, 12, 64, 65]. GMTI radars detect the position of targets by sending low PRF pulses and then examining the returns [12]. The returned signal contains many false detections in addition to the true target detection, which might sometimes also be missed, referred to as a missed target. Hence, tracking of ground targets becomes a challenging research problem with the following challenges: i) Dense clutter, ii) Topographic constraints, iii) Countermeasures utilized by targets like stop and move evasive motion and jamming; or much simpler tactics like dragging concertina wires behind vehicles, iv) Minimum detectable velocity constraint, v) Slow revisit rate [11].

Target tracking using radars detections refers to the process of estimating and predicting the state of the moving object [18, 19]. The "state" is likely to consist of kinematic components and other features of interest.

The obvious approach to ground target tracking is to use the techniques developed for airborne targets. These techniques include alpha-beta filter, Kalman Filter (KF), Interacting Multiple Model (IMM) Filter, Particle Filter, Multiple Hypothesis Tracker (MHT) and so forth [4, 8, 44, 7]. Due to clutter in radar detections, data association techniques must be used as well such as Nearest Neighborhood (NN), Probabilistic Data Association Filter (PDAF), Joint Probabilistic Data association Filter (JPDAF) and the like [4, 6]. However, due to the challenges mentioned before, tracking algorithms should be developed directly addressing the requirements of ground target tracking. The literature on this has been richer in late 1990s with the development of tailored IMM methods such as the Variable Structure-IMM (VS-IMM) method for ground targets [22, 23, 29, 35, 38, 36, 49, 50, 56]. Later particle filters, hybrid IMM/MHT methods also arised [1, 2, 27].

GMTI radars and trackers seem to be classified technologies. The implementation details of practical trackers and radars are not known. The techniques appeared in the literature are general in the sense that a significant amount of work is needed in order to turn them into practical trackers.

1.1 Motivation and Scope

The motivation of the thesis derives from the fact that, simple, elegant, computationally efficient tracker structures are always sought in the application domain. Although theoretically involved techniques such as particle filters, multiple hypothesis trackers exist and elaborated a lot in the literature, none of them lends itself to practical application. Simply stated, the problem in GMTI tracking is to "develop a state estimator that provides the location of the target with minimum position error promptly, computationally efficiently and if possible, in a theoretically tractable way". Theoretical basis is important in the sense that performance of the tracker can be predicted.

The scope of this thesis is to develop a ground target tracker that is implementable in practical systems under realistic scenarios. A single target of interest is considered. Realistic here refers to almost all possible cases that a GMTI radar can encounter. The tracker developed here can act as a fundamental basis to develop a tracker to be used in airborne platforms.

The basic state observer Kalman Filter lay the foundation of target tracking which is nothing but predicting the target's state. The elegant Kalman Filter solution is the optimal tool for target tracking if all modelling assumptions in the Kalman Filter are satisfied. If the deviations from assumptions are at an acceptable level the Kalman Filter can still be used with modifications [4, 8, 9].

False detections due to non-targets bring a critical problem in Kalman Filter tracking. This has to be separately addressed by the so-called data association techniques [4, 5]. The most common one is the probabilistic data association (PDA) which weights the detections in the validation gate instead of selecting a single one. Hence a soft association is formed which helps reducing the effect of clutter. If there are multiple targets around, a joint PDA must be used. JPDA also compute weights in overlaying gates of multiple targets which in turn help the tracker maintain multiple targets [6].

A single target dynamic model is usually a restraining factor in Kalman Filter target tracking. To mitigate this problem, multiple dynamic models are used in parallel to build up multiple model structures [9, 51]. For example, if the target is maneuvering, a single dynamic model that captures a constant velocity or constant acceleration motion of the target is not enough. A different dynamic model which should capture the maneuver of the target behaviour must be run as well. At each time step of tracking, model transitions are modelled using a Markov matrix. The state prediction is computed using a weighted average of individual model outputs. Multiple model configurations such as AMM, GPB1, GPB2, IMM along with different estimators such as Minimum Mean Square Error (MMSE) and maximum A Posteriori (MAP) create opportunities to develop new state estimators [33]. The most common multiple model structure is the IMM structure [9, 33, 51].

The PDA and the JPDA algorithms assume the existence of tracks. If this assumption is not valid, PDA and JPDA must be modified so that a measure of "track quality" is computed. "Track quality" indicates three distinct states: track exists, track does not exist and track undecided. This leads to the development of Integrated PDA (IPDA) [14, 47] and Joint Integrated PDA (JIPDA) [48] techniques. The combination of the KF, PDA and IMM constitutes the IMM-IPDAF algorithm [14, 46] which is then regarded as a ground target tracking algorithm in cluttered environments.

The literature is rich in variations of the aforementioned tracking and data associations, the details of which are not given here. The interested reader can refer to [25, 43, 46, 53].

An important performance metric of a ground target tracker is that true target life should be as long as possible [11]. If the target is dropped unnecessarily, this brings unnecessary computational burden and complexity to the tracker. An important contribution of this thesis is to minimize potential track drop conditions in ground target tracking by enhancing the tracker with solutions to address these conditions. For example, tunnel crossing and mountainous areas that are not in the line of sight of the radar as well as target stops are potential track drop conditions.

1.2 Original Contributions

To keep the tracker structure simple, a variable structure multiple model based approach is utilized. This helps to keep the computational complexity at acceptable levels [33]. Although the variable structure techniques appeared in the literature earlier, further research is needed to implement a practical responsive ground variable structure multiple model tracker.

- The original contribution here is that the state estimation problem is solved by developing a new composite Variable Structure - Autonomous - Interacting Multiple Model Filtering (VS-A-IMM) technique. The theoretical basis of this configuration correspond to composite MMSE - MAP estimation of the variable structures involved in state estimation to support on-road off-road transitions, junction crossings, stops etc. Consequently, the VS-A-IMM tracker is responsive to sudden transitions.

- A stop model is developed and integrated to the VS-A-IMM tracker to address the evasive motion of ground targets in dense clutter for realistic scenarios.

- A hide model is developed and integrated to the VS-A-IMM tracker to address the non-detectable cases of ground targets such as topographic obstructions, e.g., non-line of sight beyond hills and in tunnels. The hide model is independent of the duration of the non-observable period. However, the location of the end of obstruction must be known a priori.

- Gating in ground target tracking is an important problem which has not attracted the attention it deserved. Particularly in VS-A-IMM, a priori information combined with tracker output lead to effective gating methods. It is shown in this thesis that "smart" gating methods reduce the position error in on-road positions.

- In some cases when detections are missed in consecutive scans track initiation is needed. This is actually a temporary loss of target, not a real track drop. Such cases are addressed by the IPDAF method [47, 14]. The IPDAF is adapted to the VS-A-IMM tracker because target detection is not always possible in GMTI radars. Targets might not be detected for a number of consecutive steps which obviously affects track quality.

- The overall resulting architecture is a new practical ground target tracker that integrates the tracking capability of special cases such as stops, topographic obstructions, on-road/off-road off-road/on-road transitions in the new methodology. To the best knowledge of the author, such a structure has not appeared in the literature yet.

1.3 The organization of the thesis

The organization of the thesis is as follows: Chapter 2 presents the VS-IMM based algorithms and formulations in the literature. A comprehensive survey of ground target tracking techniques, particularly the variable structure methods are elaborated. The building blocks of target tracking namely, Kalman Filtering, multiple models, data association, validation regions are discussed.

In Chapter 3 the state estimation problem is modelled as a new composite MMSE-MAP estimation structure. This structure lends itself to a mathematically tractable model. GMTI sensor model, autonomous multiple model filtering, interacting multiple model filtering and the corresponding estimation approaches are described.

Chapter 4 develops the VS-A-IMM based single target ground tracker. The branches of the composite multiple model structure are described. The way that variable structures are involved including the stop model and the hide model are elaborated and a complete description of the resulting tracker is given.

Chapter 5 is dedicated to the performance analysis of the VS-A-IMM algorithm using simulated realistic scenarios. The comparisons with respect to baseline IMM methods are given. It is seen that the VS-A-IMM algorithm outperformed the baseline methods.

Finally, Chapter 6 concludes the thesis and presents future work.

CHAPTER 2

GROUND TARGET TRACKING TECHNIQUES

2.1 SENSOR CHARACTERISTICS - GMTI RADAR

GMTI radar distinguishes moving targets from background clutter by virtue of Doppler returns. Apparently a moving target with respect to the radar has a Doppler shift in the return signal, whereas the stationary background has zero Doppler. When the return signal for each range cell is processed using the appropriate signal processing techniques, the existence of the target can be detected.

In a pulsed Doppler radar detection problem, range and velocity ambiguities cannot be avoided at the same time. If range is to be determined without ambiguity, velocity measurements will turn out to be ambiguous or vice versa. Since the measurement of velocity is not a concern for GMTI radars, low pulse repetition frequency (PRF) pulses can be used safely [12] to avoid range ambiguities. However, it is unavoidable that Doppler blindness would occur anyway due to folding of frequencies that result from periodic repetition of pulses. Integer multiples of the following equation fold onto the origin (zero velocity region) which results in zero Doppler frequency for the signal,

$$\frac{C \times PRF}{2 \times \text{Transmit Frequency}} \tag{2.1}$$

where C is the speed of light.

The important performance metrics of a GMTI radar is as follows [11]:

- Probability of Detection: Defined as the probability of detecting a target whenever a radar pulse hits a target. It is dependent on the radial velocity (range rate) obtained by the sensor, which is determined by the sensor/target geometry [24, 58].
- Target Location Accuracy: Determined by the design of the radar operating characteristics such as azimuth resolution, range resolution.
- Minimum Detectable Velocity: Slow moving targets below a certain threshold cannot be detected due to limitations of signal filtering in range cells.
- Target Range Resolution: If targets are moving in close proximity the radar might not be able to separate those targets.
- Stand-off Distance: This is the distance from the radar to the coverage area. The radars with more power has large stand off distances.

- Coverage Area Size: This is the area that the radar can cover at the stand-off distance. It is dependant on the antenna aperture and the power radar radiates.
- Coverage Area Revisit Rate: The radar beam revisits a range cell at a certain frequency. Frequent revisits are important to maintain healthy tracks.

2.2 SINGLE MODEL TECHNIQUES

The main goal in the single-target tracking is to evaluate the posterior probability density function (pdf) with given set of observations. At each scan or time step, target state posterior pdf is collapsed to a single Gaussian.

Tracking of a single target is dependent on the assumption that a single model is chosen to describe the full motion of the target and the target motion obeys the unique model selected. Hence, a single filter is processed to obtain the state estimate of the target.

When there is no motion uncertainty and measurement origin uncertainty, Kalman filter is the optimal single target tracking algorithm when white process and measurement noise is injected into the linear dynamic system and forms the basis of the many of high level tracking algorithms.

2.2.1 Kalman Filter

The basic state observer Kalman Filter [19] forms the basis of target's state estimation. The state may consist of position, velocity, acceleration, angular velocity and the like, depending on how the motion of the target is modelled. In ground target tracking using GMTI reports, the observation model consists of unambiguous range (position) measurements only (no velocity measurement).

Kalman Filtering for one cycle is described below [63]:

The Kalman Filter assumes that initial state estimate, $\hat{x}_{0|0}$, and its estimate error covariance, $P_{0|0}$, are known as

$$\hat{x}_{0|0} = E\{x_0\} \tag{2.2}$$

$$P_{0|0} = E(x_0 - \hat{x}_{0|0})(x_0 - \hat{x}_{0|0})^T \tag{2.3}$$

respectively.

The state estimate which describes the mean of the state distribution at scan k with observations z^k is given by

$$\hat{x}_{k|k} = E\left\{x_k|z^k\right\}. \tag{2.4}$$

The z^k indicates a sequence of measurements accumulated from the radar and is denoted as $z^k = \{z_i\}_{i=1}^k$

The corresponding state covariance projects the variance of the state distribution and is obtained at scan k as:

$$P_{k|k} = E\left\{(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T|z^k\right\}. \tag{2.5}$$

The state prediction $\hat{x}_{k+1|k}$ is then evaluated by updating the current state estimate in Eq. 2.4 for the next time step [63]

$$\hat{x}_{k+1|k} = F_k\hat{x}_{k|k} \tag{2.6}$$

where F_k is the state transition matrix.

The state prediction covariance, $P_{k+1|k}$, is updated in time by the current error covariance estimate $P_{k|k}$:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_{k+1} \tag{2.7}$$

where Q_{k+1} is the process noise covariance.

The measurement prediction is evaluated by combining the measurement matrix H_{k+1} and the state prediction Eq. 2.6 as follows

$$\hat{z}_{k+1|k} = H_{k+1} \hat{x}_{k+1|k}. \tag{2.8}$$

The measurement prediction covariance (also called as innovation covariance or residual covariance matrix) is given by:

$$S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}. \tag{2.9}$$

The innovation or residual which projects the difference between observed measurement, z_{k+1}, and the predicted measurement, $\hat{z}_{k+1|k}$, in Eq.2.8 is as follows

$$\tilde{z}_{k+1} = z_{k+1} - \hat{z}_{k+1|k}. \tag{2.10}$$

The filter gain calculation is then evaluated as

$$W_{k+1} = P_{k+1|k} H_{k+1}^T S_{k+1}^{-1}. \tag{2.11}$$

As it can be seen in Eq. 2.11, the filter gain, W_{k+1}, is directly proportional to the state prediction covariance, $P_{k+1|k}$ and inversely proportional to the innovation covariance S_{k+1}. Thus, the filter gain becomes large if the state prediction has a large variance and the measurement has a relatively small variance. A large gain indicates that, to update the state estimate, it is given more weight to the measurement.

The resulting filter gain and innovation is then multiplied to update the state estimate for the next step. Then the state estimate update at scan $k + 1$ is evaluated as:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + W_{k+1} \tilde{z}_{k+1}. \tag{2.12}$$

The updated state covariance associated with the state estimate update in Eq. 2.12 is given by [63]

$$P_{k+1|k+1} = P_{k+1|k} - W_{k+1} S_{k+1} W_{k+1}^T. \tag{2.13}$$

Although Kalman Filter is the optimal algorithm for a single target tracking when the target motion characteristics do not change and is known, and there is no clutter, two main problems still arise which Kalman Filter does not take into account: i) the algorithm ignores the motion change even though there is a mismatch between the motion model and the true target mode, ii) model selection is performed a priori in the algorithm and the estimation errors are not taken into account to catch the target motion in a recursive manner. When the target evolves in the presence of the clutter, measurement origin uncertainty problem occurs and target tracking problem has to be handled taking the clutter into the consideration.

2.3 TRACKING IN CLUTTER

Clutter is one of the major challenges encountered in tracking of ground targets. The GMTI sensor detects a huge number of measurements at each scan. For example, coastal areas, wet soil, reflections due to the topography, birds, vegetated areas are sources of clutter. Interestingly, the target of interest could well be the cause of clutter such that the concertina wire carried by the target causes false detections [11].

There are two stages in tracking of a single target in a cluttered environment:

i) The most feasible region where the measurement from the target is likely to be placed needs to be identified. This process is referred to as “gating”or “validation region”.

ii) The measurements validated in the previous step is taken into account which is then used in the tracker algorithm.

In section 2.3.1, the gating approach is discussed.

2.3.1 Defining Validation Regions (Gating)

In order for tracker to calculate the state estimate as close as possible to the position and velocity of the true target, it is needed to eliminate the unwanted measurements. Therefore, the strongest measurements which are presumably to have been originated from the true target are selected around the predicted state as shown in Figure 2.1.

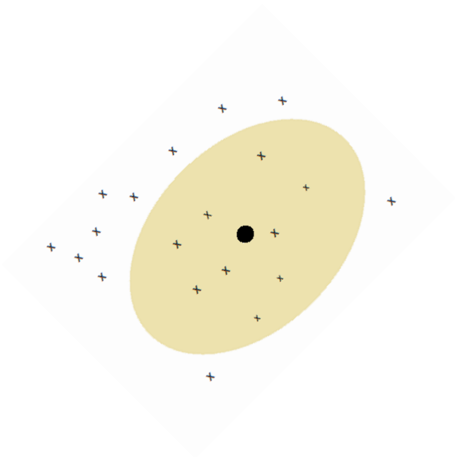

Figure 2.1: Validated measurements

There are several gating techniques [8, 61] among which the chi-square test described below is the most common one:

i) Measuring / calculating the normalized distance, d_ℓ^2, between the predicted measurement and the measurement of interest as follows

$$d_\ell^2 = (z_k^\ell \; - \; H \, \hat{x}_{k|k-1}) \, S_k^{-1} \, (z_k^\ell \; - \; H \, \hat{x}_{k|k-1}) \tag{2.14}$$

where S_k is the residual covariance matrix or innovation covariance at scan k and is evaluated by $S_k = HP_{k|k}H + R$.

ii) Compare calculated distance, d_l^2, with gate threshold, γ,which denotes the size of the validated gate around the predicted measurement,

$$d_\ell^2 \leq \gamma. \tag{2.15}$$

The clutter density is evaluated by dividing the number of measurements in the validation gate to the volume of the gate as follows

$$\lambda = \frac{N_k}{V_k}. \tag{2.16}$$

This is referred to as the non-parametric computation of the clutter density.

The gate volume [8] is defined as

$$V_k = c_{n_z} \, \gamma^{\frac{n_z}{2}} \, |S_k|^{\frac{1}{2}} \tag{2.17}$$

where c_{n_z} is a volume of a n_z dimensional unit hypersphere given by

$$c_{n_z} \ = \ \frac{\pi^{n_z/2}}{\Gamma(n_z/2+1)} \tag{2.18}$$

where $\Gamma(.)$ is the gamma function. For two dimensions, $c_2 = \pi$.

The gate center is usually taken as the location of the predicted measurement.

In Kalman Filtering, the measurement is assumed to have been obtained from the target only. If false detections exist in the validation region, it is not easy to decide which measurement comes from the true target. This problem would be described as the measurement origin uncertainty.

The GMTI radar can not detect which measurement is likely to have been originated from the true target. Therefore, a mechanism referred to as "data association" is required to estimate the track state which is also corrupted by noise for each time step.

The milestone development to address measurement origin uncertainty problem is the Probabilistic Data Association Filter (PDAF) [4, 5, 8, 21] which enables the tracker to work in a cluttered environment in the tracking of a single target.

2.3.2 Probabilistic Data Association Filter

The PDAF algorithm combines a Kalman Filter with a data association mechanism in a probabilistic manner based on the assumption that target already exists. The main approach for the PDAF would be described as:

- Form a gate which has a certain gate threshold around the predicted measurement,
- Find probabilities of each hypothesis for each measurement in the validated gate,
- Find weighted innovation,
- Apply the Kalman Filter.

The PDAF resolves measurements origin uncertainty by calculating the probabilities of the below hypotheses at each validated measurements: i) none of the measurements are target originated, ii) one of the measurements is target originated.

Let us define the possible events in the construction of the probabilities of the hypotheses:

- ξ_k^o is the event that the none of the observations are relevant to the target.
- ξ_k^1 is the event that the observation 1 is originated from the true target.
- $\vdots$
- ξ_k^ℓ is the event that the observation N_k is originated from the true target.

Hypothesis 1: None of the validated measurements is target originated

β_k^0 represents the conditional probability that none of the observations are relevant to the target of interest:

$$\beta_k^0 = \frac{b}{b + \sum_{\ell=1}^{N_k} e^{-d_\ell^2}}. \tag{2.19}$$

Hypothesis 2: Any of the validated measurements is target originated

β_k^ℓ is probability that the ℓ^{th} measurement would be obtained from the true target ($\ell = 1, \cdots, N_k$),

$$\beta_k^\ell = P\left\{\xi_k^\ell | \chi_k, z^k\right\} = \frac{P_d e^{-d_\ell^2}}{b + \sum_{\ell=1}^{N_k} e^{-d_\ell^2}}. \tag{2.20}$$

where d_ℓ^2 is the Mahalanobis distance,
$b = (1 - P_d P_g)\lambda(2\pi)^{M/2}$,
M is the dimension of the measurement vector,
λ is the clutter density,
P_d is the probability of detection,
P_g the probability that a correct target return will fall within the track gate.

The state estimation is then obtained by combining the association probabilities, β_k^ℓ, and the updated state estimate for each measurement, $\hat{x}_{k|k}^\ell$, as

$$\hat{x}_{k|k} = \sum_{\ell=0}^{N_k} \beta_k^\ell \, \hat{x}_{k|k}^\ell. \tag{2.21}$$

The PDAF algorithm assumes that the target being tracked exists and is not able to initiate a target or provide probability of track existence.

In most cases, the target measurements are not reliable and show up with a certain probability of detection. Moreover, the target can be sometimes dropped due to clutter, maneuvering motion, stop-move motions and so forth. Hence, it is required to initialize the target again. This leads to the development of the Integrated Probabilistic Data Association Filter [14, 47].

2.4 TARGET OBSERVABILITY - Integrated Probabilistic Data Association Filter

The Integrated Probabilistic Data Association Filter (IPDAF) is an algorithm that supports track initiation and maintenance by calculating the track existence probability in addition to the PDAF algorithm at each time step.

The IPDAF filter handles the problem of track existence by modelling it using a Markov process which arranges the model transition in two possible ways: i) Markov chain one, ii) Markov Chain two.

2.4.1 Markov Chain One

Markov chain one process is generally used in track initiation or termination. There are two possibilities in Markov chain one: i) target exists and ii) target does not exist.

The probabilities for Markov chain one is obtained as follows

Hypothesis 1: None of the validated measurements is target originated. β_k^0 represents the conditional probability that none of the observations are relevant to target of interest.

$$\beta_k^o = \frac{1 - P_d P_g}{1 - \delta_k} \tag{2.22}$$

where

$$\delta_k = \begin{cases} P_d P_g & N_k = 0 \\ P_d P_g \left[1 - \frac{V_k}{\hat{N}_k} \sum_{\ell=1}^{N_k} \Lambda_k^\ell \right] & \text{otherwise} \end{cases} \tag{2.23}$$

Λ_k^ℓ is the likelihood associated with observation ℓ, P_d is the probability of detection,
P_g indicates the probability that a correct target return will fall within the track gate,
V_k is the volume of the gate,
N_k is the number of validated measurement,
$\hat{N}_k$ is the estimate of the expected number of false observations for Markov Chain One:

$$\hat{N}_k = \begin{cases} 0 & N_k = 0 \\ N_k - P_d P_g \ \psi_{k|k-1} & \text{otherwise} \end{cases} \tag{2.24}$$

where $\psi_{k|k-1}$ is the track existence probability.

Hypothesis 2: Any of the validated measurements is target originated. β_k^ℓ is the probability that the observation in the gate is arisen from the true target.

$$\beta_k^\ell = \frac{P_d P_g \frac{V_k}{\hat{N}_k} \Lambda_k^\ell}{1 - \delta_k} \tag{2.25}$$

where $\ell = 1, \ldots, N_k$.

The state estimation, $\hat{x}_{k|k}$, for the target is denoted by

$$\hat{x}_{k|k} = E\left\{ x_k \mid \chi_k, z^k \right\} = \sum_{\ell=0}^{N_k} \beta_k^\ell \hat{x}_{k|k}^\ell \tag{2.26}$$

where,
χ_k and $\hat{\chi}_k$ indicate the complementary events that the track exists and does not exist,
$\hat{x}^{\ell}_{k|k}$ is the state estimation at scan k for which measurement ℓ is originated from true target,
N_k is the number of measurements in the validation region.

The sum of each row in the Markov matrix must satisfy

$$\begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} \tag{2.27}$$

$$\pi_{11} + \pi_{12} = \pi_{21} + \pi_{22} = 1. \tag{2.28}$$

The a priori track existence probability before receiving the new measurement, $\psi_{k|k-1}$, is calculated as

$$\psi_{k|k-1} = \pi_{11}\psi_{k-1|k-1} + \pi_{21}(1 - \psi_{k-1|k-1}). \tag{2.29}$$

Upon receipt of data at scan k, we obtain the updated or filtered track existence probability as

$$\psi_{k|k} = \frac{1 - \delta_k}{1 - \delta_k\ \psi_{k|k-1}}\ \psi_{k|k-1} \tag{2.30}$$

where δ_k is evaluated as in Eq. 2.23.

2.4.2 Markov Chain Two

It is recommended that Markov chain one assumption is relaxed and the “track exists and is observable” and the “track exists but is not observable” conditions are added to the tracking process. This process is referred to as Markov Chain Two [14, 47].

There are three assumptions in Markov chain two: i) target exists and is observable, ii) target exists but is not observable, iii) target does not exist.

The total filtered track existence probabilities for Markov chain two at scan k is evaluated as below:

$$\psi_{k|k} = \psi^o_{k|k} + \psi^n_{k|k}. \tag{2.31}$$

The track existence probability that the target exists and is observable $\psi^o_{k|k}$ at scan k is updated as

$$\psi^o_{k|k} = P\left\{x^o_k \mid z^k\right\} \tag{2.32}$$

$$= \frac{1 - \delta_k}{1 - \delta_k\ \psi^o_{k|k-1}}\ \psi^o_{k|k-1} \tag{2.33}$$

where z^k is the sequence of measurements sets till scan k, $\psi^o_{k|k-1}$ indicates a priori track existence probability that the target exists and is observable described as:

$$\psi^o_{k|k-1} = \pi_{11}\psi^o_{k-1|k-1} + \pi_{21}\psi^n_{k-1|k-1} + \pi_{31}(1 - \psi_{k-1|k-1}). \tag{2.34}$$

The track existence probability that the target exists and is not observable at scan k is

$$\psi^n_{k|k} = P\left\{x^n_k \mid Z^k\right\} \tag{2.35}$$

$$= \frac{\psi^n_{k|k-1}}{1 - \delta_k\ \psi^n_{k|k-1}} \tag{2.36}$$

where

$$\psi^n_{k|k-1} = \pi_{12}\psi^o_{k-1|k-1} + \pi_{22}\psi^n_{k-1|k-1} + \pi_{32}(1 - \psi_{k-1|k-1}). \tag{2.37}$$

The coefficients appear in the Matkov matrix as follows

$$\begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix}. \tag{2.38}$$

The sum of the Markov chain coefficients in each row must obey

$$\begin{aligned} \pi_{11} + \pi_{12} + \pi_{13} &= \pi_{21} + \pi_{22} + \pi_{23} \\ &= \pi_{31} + \pi_{32} + \pi_{33} \\ &= 1. \end{aligned} \tag{2.39}$$

The probabilities that the measurement ℓ is likely to have been originated from the true target in Markov chain two are calculated as follows

$$\beta^o_k = \frac{(1 - P_d P_g)\, \psi^o_{k|k-1} + \psi^n_{k|k-1}}{(1 - \delta_k)\, \psi^o_{k|k-1} + \psi^n_{k|k-1}} \tag{2.40}$$

$$\beta^\ell_k = \frac{P_d P_g\, \frac{V_k}{\hat{N}_k} \Lambda^\ell_k \psi^o_{k|k-1}}{(1 - \delta_k)\, \psi^o_{k|k-1} + \psi^n_{k|k-1}} \tag{2.41}$$

where δ_k is calculated as in Eq. 2.23, Λ^ℓ_k is the probability density function in which (that) the measurement ℓ (true target) is in the validated region.

The expected number of false observations $\hat{N}_k$ for Markov Chain Two is

$$\hat{N}_k = \begin{cases} 0 & N_k = 0 \\ N_k - P_d P_g\, \psi^o_{k|k-1} & \text{otherwise.} \end{cases} \tag{2.42}$$

Even together with the incorporation of track existence, Kalman Filtering using a single target motion model is still not usually enough to describe the full range of possible target motions. For example, if the target is maneuvering, but the motion model assumes a constant velocity model, it is likely that the track would be dropped. To make up for this shortcoming, multiple models are processed in parallel. This leads to the development of the multiple model filtering. In reality, it is a necessity to use multiple models if there is a motion uncertainty in the target dynamics [51].

2.5 MULTIPLE MODEL TECHNIQUES

Multiple model techniques come in to play when there is a motion uncertainty in the motion of the target and the use of more than one model is required to describe the full motion of the target. Each model consists of a filter, all filters are executed in parallel and the outputs of each filter are merged by different types of methods to obtain an overall state estimation of the target.

The main approach in the multiple model techniques can be specified as:

- Form a set of motion models which the target of interest is likely to follow,
- Use a group of filters, every filter can work on a different model,
- Obtain an overall estimate.

Many types of multiple model target tracking approaches are introduced in the literature which are discussed briefly in the following sections: Autonomous Multiple Model (2.5.1), First Order Generalized Pseudo Bayesian Estimation (2.5.2), Second Order Generalized Pseudo Bayesian Estimation (2.5.3), Interacting Multiple Model Estimation (2.5.4), Variable Structure Interacting Multiple Model (2.6).

2.5.1 Autonomous Multiple Model Estimation

The autonomous multiple model (AMM) technique [51] is known as the first generation multiple model technique which has a fixed structure model set. Each filter in the AMM works in an individual and independent manner as shown in Figure 2.2.

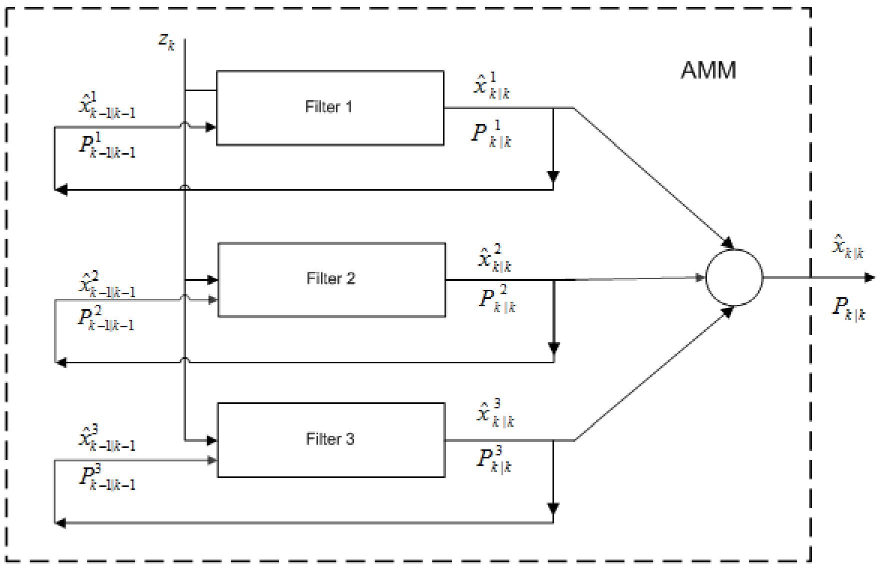

Figure 2.2: The structure of the AMM algorithm with three motion models

It does not consider a switching mechanism among motion modes, therefore, there is no interaction within the model set. The overall estimate would be a combination of the individual filters to obtain a state estimate of the target. The output of the each filter is directly fed back to the filter input for the next time step in an autonomous manner.

2.5.2 First Order Generalized Pseudo Bayesian Estimation

First Order Generalized Pseudo Bayesian (GPB1) [51] is a second generation multiple model technique in which motion modes would be able to switch between the models according to the Markov transition probabilities:

$$\pi_{ij} = P\left\{M^j_{k+1}|M^i_k\right\} \tag{2.43}$$

where M_k^i indicates the model i in effect for time step k and M_{k+1}^j is the model j in effect for $k+1$.

A general structure of GPB1 for three models is indicated in Figure 2.3:

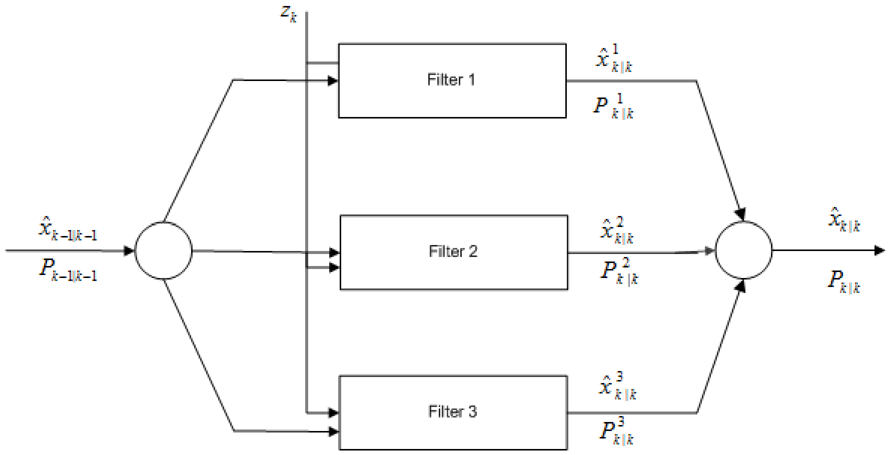

Figure 2.3: The structure of the GPB1 algorithm

The GPB1 design structure is defined as follows

- The algorithm executes M filters in parallel and evaluates model conditioned state estimates and model probability for each model,
- Model probability updates allow model transitions through Markov transition probability,
- The overall state estimate is used to reinitialize for the next time step.

GPB1 lacks mixing of filter outputs to reinitialize the filters as seen in Figure 2.3. However, model transitions are allowed. GPB1 has a time depth of one instant such that it considers the possible models only at the latest time instant. This corresponds to merging all previous model sequences into one [51].

2.5.3 Second Order Generalized Pseudo Bayesian Estimation

The Second Order Generalized Pseudo Bayesian (GPB2) algorithm [51] accounts for the possible models by taking latest two time steps and carries all previous mode sequences to the current step as a state estimate and their associated covariance.

The GPB2 design structure has the following properties:

- Each initial condition for the state estimate and the covariance is maintained,
- The algorithm runs M filters in parallel and evaluates a total of M^2 model conditioned state estimates and model probabilities for each model,
- Model probability updates are merged,

- The overall state estimate is fused.

The interested reader can refer to [51] for further details.

The complexity of the algorithm grows as the degree of the Generalized Pseudo Bayesian Algorithm increases [4]. It is not a practical algorithm to implement and it is not recommended to use it with more than two model filters. Obviously, this algorithm is eliminated when the concern is the ground target tracking which accounts for onroad and offroad segments by using different model filters at each time step.

2.5.4 Interacting Multiple Model Estimation

Interacting Multiple Model is one of the most effective and simple mechanisms for estimating the state of a dynamic system with several predetermined behavior models which can switch from one model to another [8, 9]. During one sampling period, one of the models may describe the target's motion.

The IMM algorithm is implemented based on a set of multiple parallel filters, where each filter is set up for one of the assumed (predetermined) models. These models may have different dynamic equations and process and measurement noises and are determined before the estimation process is started. The system model is assumed to evolve through the modes according to a Markov chain. Due to switching between the models, there is an interaction of information between filters used in the IMM.

The IMM algorithm consists of a filter for every single model, a model evaluator, an estimation mixer at the input of the filters and an estimate combiner at the output of the filters. A typical IMM structure with three models is given in Figure 2.4.

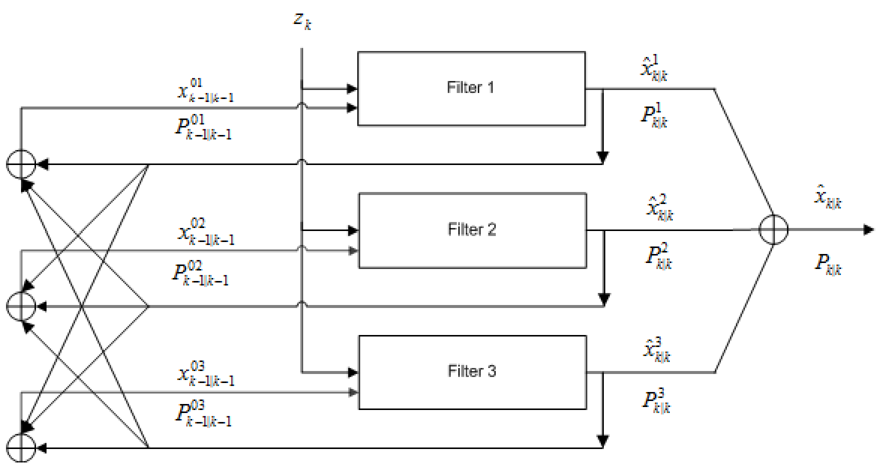

Figure 2.4: The structure of the IMM algorithm with three models

It is also important to mention that bringing the advantages of interacting multiple modelling and track observability together leads to the IMM-IPDAF algorithm [47, 14].

The IMM technique provides better estimates compared to single model approaches when moderate number of models is used. However, when there is a need for a large number of models, not only is the computational load increased but also estimation accuracy is degraded. The limitations of the IMM filter are that the number of models should be as small as possible, the models should represent the true target motion dynamics as much as possible [22, 23]. When there is a need for a large number of models, the problem is circumvented by employing variable structure techniques.

2.6 VARIABLE STRUCTURE TECHNIQUES

When a priori information is available, it is obviously to the advantage of the tracking process to use it for better estimating the state. Topographic information, road maps are among the a priori information that can be brought into play in state estimation. This is sometimes referred to as "knowledge assisted tracking" [58, 59]. In the framework of IMM filtering, the use of a priori information has led to the growth of Variable Structure IMM (VSIMM) techniques [22, 23, 49, 50].

Models are adaptively changed, added, or removed from the algorithm based on the terrain topography in ground target tracking. The added uncertainty at junctions is handled by temporarily augmenting the IMM mode set with modes that represent motion along all possible roads. These additional IMM modes are removed from the mode set after the target passes the junction. At each scan, the structure of the estimator for every target is modified individually based on the known topology of the region and the predicted location of the target [22, 23, 49, 50]. Basic properties of the VSIMM filters are

- The mode set not only differs across targets, but also varies with time for a given target.
- At each scan, the structure of the estimator for every target is individually modified based on the known topography of the surface region and the predicted location of the target.
- At each subsequent revisit time, models in the estimators are added or deleted based on the topography as follows
  - On road / off road motion
  - Junctions
  - Entry /Exit Conditions
  - Obscuration Conditions

Map Information
All road coordinates are converted into the Topographic Coordinate Frame (TCF). The TCF frame is defined such that the TCF axes are respectively oriented in the east, the north and the up direction. The origin O (ϕ_o, λ_o, h_o) of the TCF frame[8] is expressed in the World Geodetic System (WGS84) frame as in Figure 2.5.

Measurement Model
The measurements from GMTI radar in the WGS84 frame is projected on the Digital Terrain Elevation Data (DTED) in TCF format.

Target State
The target state at time k is given as local coordinated frame (X_{TCF}, Vx_{TCF}, Y_{TCF}, Vy_{TCF}) and the altitude in the X-Y plane of the TCF is eliminated [49, 50].

Road Constraint
Roads are comprised of sections in the TCF frame such that each road section has a road start and

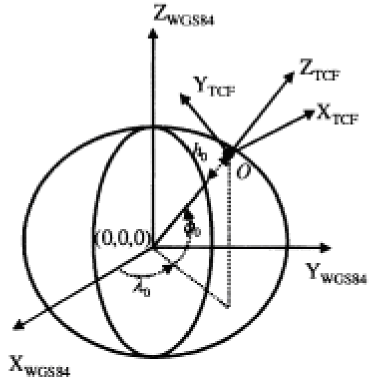

Figure 2.5: TCF Frame

road end and is defined by a set of linear segments. The dynamics of the target moving on the road are modelled by a first order system [22, 23].

A target evolving on a road segment is expected that its position belongs to the linear segment s and the velocity vector is in the road segment s direction.

Therefore, the constraint on the target state is

$$ax_k + by_k + c = 0 \tag{2.44}$$

where a, b, c are the coefficients of the first order linear equation.

The constraint on the velocity is

$$< [Vx_k, Vy_k]\, n_s >= 0 \tag{2.45}$$

where n_s is the orthogonal vector to the road segment s.

The constraint in the matrix form is

$$\tilde{D}x_k = L \tag{2.46}$$

where $\tilde{D} = \begin{bmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \end{bmatrix}$ and $L = \begin{bmatrix} -c \\ 0 \end{bmatrix}$.

Offroad Condition
If a priori information of the road is not considered, off-road process noise components of the constant velocity model along X and Y are given by v_x and v_y and their variances are σ_x^2 and σ_y^2 respectively. For offroad motion, it is chosen that $\sigma_x^2 = \sigma_y^2$.

Off-road state vector with no road constraint is given as

$$x_{k+1} = Fx_k + Gw_k \tag{2.47}$$

where w_k is the independent Gaussian random process which adds uncertainty in the motion of the target as in Figure 2.6.

Onroad Condition
Considering that the target is onroad motion, the state vector would have road segment constraints

$$x_{k+1} = F_k^s x_k + G_k^s w_k \tag{2.48}$$

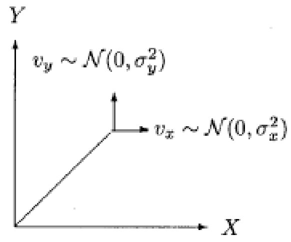

Figure 2.6: The process noise model for offroad motion

where w_k is the directional process noise used on the road segment s as shown in Figure 2.7 [49, 50]. For on road conditions, the state transition matrix F_k^s associated to the road segment s is

$$F_k^s = \begin{bmatrix} C & 0_{2x2} \\ 0_{2x2} & C \end{bmatrix} + P_\perp \begin{bmatrix} D & 0_{2x2} \\ 0_{2x2} & D \end{bmatrix} \tag{2.49}$$

where,

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 0 & T \\ 0 & 0 \end{bmatrix}, \tag{2.50}$$

$P_\perp = I - \tilde{D}^T(\tilde{D}\tilde{D}^T)^{-1}\tilde{D}$ is the orthogonal projection of the state vector on the associated road segment.

In the on-road target motion model, the process noise and its variance along the direction of the road are v_a and σ_a^2. The process noise and its variance orthogonal to the road segment are v_o and σ_o^2.

Due to the higher motion uncertainty along the road, the variances σ_a^2 and σ_o^2 are selected as $\sigma_a^2 \gg \sigma_o^2$.

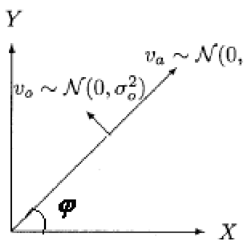

Figure 2.7: Bidirectional process noise for onroad motion

Since the estimation is carried out in the X-Y coordinate system, the model of the process noise components along the road and orthogonal to the road in the on-road model must be converted and incorporated into covariance matrix in that frame as in [49]

$$Q_k^s = R_\varphi \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_o^2 \end{bmatrix} R_\varphi^T \tag{2.51}$$

where R_φ is 2x2 rotation matrix used for process noise covariance.

Projection of the Target State on the Road Segment

Using a probabilistic approach [49, 50] the orthogonal projection of the state estimate on the road segment s is

$$\underset{x \in s}{\text{argmin}} ||x_k - \hat{x}_{k|k}||_{P_{k|k}}. \tag{2.52}$$

Using the Lagrangian approach, the analytical expression for the constrained Maximum A Posteriori (MAP) estimate is given by

$$\hat{x}_k^s = \hat{x}_{k|k} - P_{k|k}\tilde{D}^T(\tilde{D}P_{k|k}\tilde{D}^T)^{-1}(\tilde{D}\hat{x}_{k|k} - L) \tag{2.53}$$

and its covariance is

$$P_{k|k}^s = \left(I - P_{k|k}\tilde{D}^T(\tilde{D}P_{k|k}\tilde{D}^T)^{-1}\tilde{D}\right)P_{k|k}\ \left(I - P_{k|k}\tilde{D}^T(\tilde{D}P_{k|k}\tilde{D}^T)^{-1}\tilde{D}\right)^T. \tag{2.54}$$

2.7 MULTIPLE TARGET CASE

Although the initial development of tracking algorithms are done using single targets, in reality however tracking of multiple targets is required. The basic PDAF in this case is extended to the Joint PDAF when there are a known number of targets in the clutter [6]. The multiple target coverage is out of scope in this thesis, however the JPDAF is used in track initiation which is described in Section 4.1.4.

Joint Probabilistic Data Association Filter

Joint Probabilistic Data Association (JPDAF) is a technique used in the tracking of multiple targets. When there are several targets in the same region, a measurement from one target can fall inside the validation region of a neighboring target.

The JPDAF algorithm computes the probabilities of association only for the latest set of measurements for various targets. Targets are not treated independently; if two or more targets have nonzero probability of the same measurement, then the JPDAF calculation of each target is dependent on the others. These are then combined into the state estimate. It is a target oriented approach.

The assumptions in the JPDAF algorithm are

- Number of targets in the clutter is known,
- The measurements from one target can fall inside the validation region of a neighboring target, known as persistent interference. The targets validation gate may overlap,
- The past is summarized by sufficient statistics for the state estimate and covariance for each target,
- The probability density function based on the measurements and state estimates are assumed to have Gaussian distribution.

2.8 OTHER TECHNIQUES

2.8.1 Extended Kalman Filter

The aim of the Extended Kalman Filter (EKF) is to estimate the state under nonlinear measurement processes and/or nonlinear target dynamics conditions by a linearization around the current estimate (mean and covariance) [31].

Let us consider the system with dynamics

$$x_{k+1} = f(k, x_k) + w_k. \tag{2.55}$$

The measurement is denoted as

$$z_k = h(k, x_k) + v_k \tag{2.56}$$

where w_k and v_k are zero mean, white Gaussian process and measurement noises respectively. The idea here is to linearize the system around the estimated state as

$$F_k = \frac{\partial f(k, x_k)}{\partial x} \tag{2.57}$$

where F is the Jacobian matrix of partial derivatives of f wrt x and

$$H_k = \frac{\partial h(k, x_k)}{\partial x} \tag{2.58}$$

where H is the Jacobian matrix of partial derivatives of h wrt x.

The main difference from the Kalman Filter is the evaluation of the Jacobians of the state transition matrix and the measurement matrix. Due to this, the covariance computations are not decoupled anymore from the state estimate calculations and can not be done offline.

Because Kalman Filtering is preserved only around the estimated state, the EKF suffers from instability at certain regions, especially when there is excessive noise. It introduces bias and the covariance calculation based on a Taylor Series is not always accurate.

If the pdf is non Gaussian, this can introduce large errors in the posteriori mean and covariance which may lead to divergence (unbounded estimation errors) of the EKF. If the pdf is Gaussian, Unscented Kalman Filter (UKF) gives a better solution [17].

2.8.2 Multiple Hypothesis Tracker

The Multiple Hypothesis Tracker (MHT) is an algorithm used for multiple target tracking in a cluttered environment [8]. It is a measurement-oriented approach and computes the probabilities that there is a target from which a sequence of measurements originated [7].

The MHT does not assume a known number of targets as in the JPDAF algorithm and starts track initiation as the inherent property of the algorithm [8].

2.9 MULTI SENSOR FUSION

Multi sensor fusion is designed to combine information from different sensor such as electronic support measures (ESM), radar, data links and to improve the target state estimates in military applications. It also resolves the uncertainty in the received data. Different types of sensors have different strengths and weaknesses. Therefore, fusion of data from multiple sensors of different types provides a more sensitive and accurate information than using one sensor which may sometimes be strong on an important parameter for data and hence the other sensor can compensate for the weaknesses of another sensor. Multiple sensors of the same type can improve the coverage and give us a broader picture. Multiple sensors of different type can provide more accurate and complete information [3, 16, 18].

Multi sensor fusion is a level one data fusion level process combining both correlation and fusion processes to convert the sensor measurements into the updated states and covariances for multiple target tracking [13].

There are three alternatives for fusing locational information to estimate the position and the velocity:

Centralized Fusion: (Data Level Fusion) Raw data from each sensor are transmitted from sensors to the central processing unit. The data from each sensor are aligned to a common coordinate system and units for central processing. The data are then associated and correlated to determine which represent the measurements of the same target. This method gives accurate ways to fuse data by assuming that association and correlation can be done correctly. In dense target cases, the association and correlation problem may be complicated [13].

Autonomous Fusion (Distributed Fusion) Each sensor provides an estimate of state (positional / velocity) These state estimates from each sensor are sent to fusion processor to obtain a complete description of the environment. Data alignment, association and correlation stages of data fusion are also required, but in this case they are performed in state estimates in each sensor. The distribution fusion architecture reduces the communications between sensors and the fusion processor and the makes the association/correlation level easier. Nevertheless, the accuracy in the distributed fusion is worse than the raw data level fusion since there is an information loss between the sensors and the fusion [13].

Hybrid Fusion This approach includes data level fusion and the state estimation level fusion. It utilizes the advantage of both approaches. When the accuracy is important in a process e.g., a heavy cluttered environment, data level fusion (central fusion) is performed. If the clutter is not a big deal, in order to reduce the computational burden, distributed fusion is preferred. Thus, it provides flexibility by selecting the fusion process according to the situational requirements [13].

In this thesis a single radar sensor is considered. The extension of the methods presented in this thesis can be generalized to multiple radar scenarios. However, to add more value to the tracker the processing must take place at the plot level, not at the track level. This indicates centralized fusion.

2.10 VARIOUS APPROACHES

A number of multiple model applications can be found in the literature. In [62], models are selected from a model set to support maneuvering target tracking for air surveillance. Only one gate is formed around the predicted measurement for all models. Minimum Sub Model set switching algorithm is proposed.

A comparison of multiple models is made for maneuvering targets in terms of computational complexity and tracking performance. The IMM algorithm stands out among all the multiple models with low computational complexity and acceptable tracking errors [51].

Selecting best possible models among a predefined set is studied in a sequel of papers. In [35] model set adaptation is considered. Model-group switching algorithm is described in [38] whereas [36] describes the design and evaluation of the model-group switching algorithm in much finer detail. Likely model set algorithm is defined in [40]. Expected-mode augmentation is yet another approach to model selection [34]. Best model augmentation [26] is the newest approach to model selection. These variable structure techniques do not consider cluttered environments, hence concentrating mostly on model adding and deleting.

A key observation in the literature is that variable structure techniques are commonly used to select models among a parametrized set of models of the same kind. For instance 13 constant acceleration models are defined, each of which corresponding to a constant acceleration [34]. The variable structure algorithm tries to select the best model which fits the motion of the target. A similar approach can also be taken for constant velocity or coordinated turn models. In this case the process noise variance or the turn rate is parametrized.

The variable structure multiple model approaches described above do not primarily concern ground target tracking applications. In ground target tracking it is well-known that target motion variability is not as severe as airborne target tracking. Hence using sophisticated techniques described above is not justified in ground target tracking because of unwanted computational complexity. Specifically, for example, the approach in [34] mentioned above as having 13 constant acceleration models is not useful at all. Variable structure techniques work for tracking ground targets as well, however, the problem must be approached from a different perspective. Variable structuring has to take the a priori information into account. Earlier approaches appeared in the literature are [45, 59, 24, 58, 56].

The most sophisticated approach to VSIMM ground target tracking is defined in [49, 50] which build upon the approach of [23].

Some simplified approaches to variable structuring appeared in the literature [15, 54] which circumvent the complexity problem in Li's work such as [35, 38].

Multiple model techniques are also used to detect the end of a manuevering motion of the target [42].

In fault detection, identification and estimation variable structure techniques play an important role too. A sudden appearance of a fault triggers model change in the fault tracking algorithm [55].

Augmented IMM techniques can also be defined. In this approach, the coordinated turn model in the IMM algorithm is augmented with the estimated turn rate [53]. Its variable structure version can also be defined.

IMM techniques find application domains such as air traffic control [28] and road-boundary tracking [25] as well.

CHAPTER 3

ESTIMATION FRAMEWORK

In ground target tracking scenarios the peculiarity of the problem warrants the search for better estimation frameworks. Basic multiple model structures such as the AMM and IMM filters has established a benchmark performance in ground target tracking scenarios. However, composite estimation frameworks that combine multiple model structures are lacking in the literature. A novel idea to address this problem is that, regarding the ground scenario, a mapping of a priori information to a composite estimation framework can be developed. Off road and on road segments would play equal roles in the estimation. This, for example, addresses the problem of sudden exits/entries to onroad/offroad. In this chapter a composite estimation framework is developed addressing the requirements of ground target tracking in which the target could equally be onroad/offroad, stop/move/hide, observable/not-observable, while obtaining better error performance than benchmark algorithms.

3.1 SENSOR AND MAP MODELS

For tracking purposes, the relevant GMTI radar parameter is the probability of detection. This parameter is heavily used in tracking algorithms. In tracking ground targets, velocity is not measured directly. It is a part of the kinematic model and is estimated by the tracking algorithm [59].

It turns out that the probability of detection is dependent on the radial velocity of the target with respect to the radar's position [24, 58]. Figure 3.1 shows this dependency.

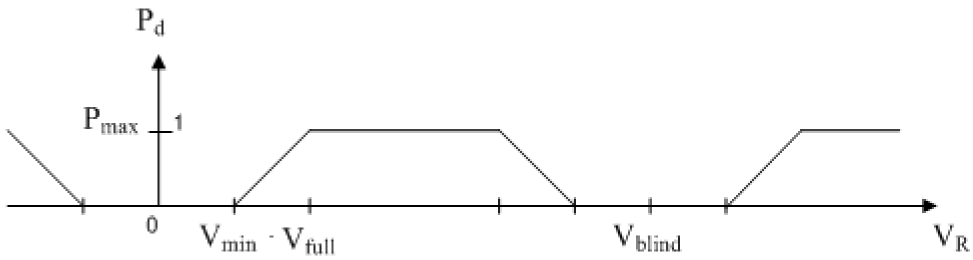

Figure 3.1: Probability of detection of the sensor versus velocity

Below $V_{\min}$, $P_d = 0$ which indicates that the signal processing involved in the GMTI radar cannot separate the moving target from the stationary background. P_d increases linearly to its max value $P_d = P_{\max}$ until $V = V_{\text{full}}$ and then stays constant in the onset. This is also the model used in this thesis with $V_{\min} = 2m/s$, $V_{\text{full}} = 4m/s$, $P_{\max} = 0.99$. $V_{\min}$ is also referred to as the MDV.

It is also obvious that $P_d = 0$ when the target is in a topographic obstruction region [45] and no measurement is received by the radar.

A weakness of the GMTI sensor is known as the Doppler Blindness [10]. The Doppler blindness causes the probability of detection to decrease [45]. Signal processing in the GMTI radar processes the received signals in each range cell. When the target velocity reaches integral multiples of the pulse repetition frequency (PRF), the GMTI output becomes zero. Hence, in Figure 3.1,

$$V_{\text{blind}} = \frac{\lambda_t}{2T_s} \tag{3.1}$$

where λ_t is the wavelength of the transmitted pulse and T_s is the pulse repetition time which implies that $f_{\text{Doppler}} = \text{PRF}$. In general, it turns out that $f_{\text{Doppler}} = n\text{PRF}$ where n is a positive integer. All targets moving at integer multiples of V_{blind} are not detected at all by the GMTI radar. For example, if $V_{\text{blind}} = 30m/s$, a target accelerating towards $V = 30m/s$ would then suddenly disappear in the GMTI radar.

The radar is assumed to be located at the origin of the map (0,0).

A non-parametric algorithm which assumes no prior knowledge of clutter density is used to model the background clutter.

Without loss of generality, but loosing some accuracy, the road map is assumed to consist of linear road segments that are modelled using a first order equation. As shown in Figure 2.7, the union of linear segments build up the whole road map.

3.2 MODELLING THE SYSTEM USING MULTI MODELS

The objective of target tracking is to estimate the state of the target. In reality, the motion model of the target is not known. The algorithms underlying Kalman filter are model based and usually assume that the target obeys a presumed motion model. The success of target tracking is closely related with the selection of the models used in the algorithm [30].

There are mainly two types of motion in the literature: nonmaneuvering and maneuvering motion. In nonmaneuvering motion it is assumed that target evolves on a straight motion at a constant velocity [30, 31].

In maneuvering mode of the target, target can suddenly change its motion, hence the tracker must recognize these changes and adapt itself by adding different models for which target is assumed to obey one of the suitable models among the predetermined models in the tracker.

In reality, an accurate knowledge of the track's motion is not available to the tracker which is referred to as motion uncertainty. Therefore, a process noise or error is incorporated to the state space model for the uncertainty in the motion of the target in the corresponding motion model.

The tracker in this thesis consists of a constant velocity model, a coordinated turn model, a stop model and a hide model. Constant velocity model with low process noise is used to handle slow moving targets while constant velocity model with high process allow capturing target's acceleration. Ground targets are, in general, assumed to evolve in accordance with constant velocity and coordinated turn models. Target can deliberately stop and move to be invisible to the tracker or move at very low speeds that GMTI radar can not detect. A stop model is embedded in the tracker to track targets moving below minimum detected velocity. Due to the topographic constraints in the ground, target may not be observed, for instance, in tunnels. A hide model is added to the model set to handle such cases.

A state space model is a mathematical representation which is comprised of two models: i) process model or motion model, ii) measurement model.

3.2.1 Motion Models

The motion models for a maneuvering target is not known a priori and one has to make an assumption that the target of interest is likely to move in one of the predetermined motion models defined as below:

3.2.1.1 Constant Velocity Model

The state space model or target state for a constant velocity model is assumed to propagate in time interms of noisy signal with the following recursive equation

$$x_{k+1} = F\, x_k + G\, w_k \tag{3.2}$$

where x_k is true state vector. The true state of the target for a discrete time constant velocity model in Cartesian coordinate system [30, 31] consists of position and velocity in this thsesis and is modelled by $x_k = \begin{bmatrix} x & V_x & y & V_y \end{bmatrix}^T$ where x, y are the position and V_x, V_y are the velocity of the target respectively.

F is the state transition matrix which describes the transition from previous state to the current state and G is the process noise matrix, defined as follows

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}$$

where T is the sampling time. The process noise w_k is defined as a white Gaussian noise vector distributed with zero mean and covariance Q, i.e., $w_k \sim N(0, Q)$ where

$$Q_k = E\left\{w_k w_k^T\right\} \tag{3.3}$$

$$E\left\{w_k\right\} = 0 \tag{3.4}$$

$$E\left\{w_k w_l^T\right\} = 0 \quad \text{if} \quad k \neq l. \tag{3.5}$$

The process noise is injected to the system to compensate the uncertainty on the motion of the target. The selection of the process is important in the tracking performance. If the process noise is below than what it must be, tracker may be unable to follow the track and this leads to miss of the target. If the process noise is given higher, the target of interest can be tracked. However, the position error increases, which is not preferable. This indicates a trade off situation which must be resolved and fine-tuned in each tracking application.

3.2.1.2 Coordinated Turn Model with Constant Turn Rate

This model assumes that the target moves with nearly constant angular (turn) rate w and the the state vector is [8] provided as

$$x_k = [x, y, V, \omega, \phi] \tag{3.6}$$

where x, y are the position, V is the heading angle of the target velocity, ϕ is the turn rate, $\phi = tan^{-1}(\frac{y}{x})$, ω is the magnitude of the velocity.

Target state is assumed to evolve in accordance with the following recursive equation

$$x_{k+1} = F^c\ x_k + G^c\ w_k \tag{3.7}$$

where F^c is the state transition matrix and G^c is the process noise matrix of the coordinated turn model respectively which are defined as follows

$$F^c = \begin{bmatrix} 1 & 0 & \Phi_{13} & \Phi_{14} & \Phi_{15} \\ 0 & 1 & \Phi_{23} & \Phi_{24} & \Phi_{25} \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad G^c = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \tag{3.8}$$

When the common model is chosen as the constant velocity model, the parameters of coordinated turn is converted as follows

$$\begin{aligned} x &= x & \qquad y &= y \\ V_x &= V\cos\phi & \qquad V_y &= V\sin\phi \end{aligned}$$

The turn rate ω is constant and does not affect the V_x and V_y components [8]. Therefore, an EKF is not required in the tracking algorithm.

3.2.1.3 Stop Model and Hide Model

The state space vector of the target when in stop condition is given [64] as

$$x_{k+1} = F^{\text{stop}}\ x_k + G^{\text{stop}}\ w_k \tag{3.9}$$

where F^{stop} is the state transition matrix for the stop model and G^{stop} is the noise matrix defined as

$$F^{\text{stop}} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad G^{\text{stop}} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \tag{3.10}$$

The stop model is replaced with the constant velocity model when the target is in stop mode. The state estimate, associated covariance and likelihoods are evaluated considering that the target is not visible to the tracker, i.e., probability of detection is zero.

The hide model also uses the same state space vector defined in Eq. 3.9. However, the operation details of the hide model are different than the stop model and are further elaborated in Section 4.1.3.

3.2.2 Measurement Model

Measurements are received from the GMTI radar and formed as a random set $z_k = \left\{z_k^\ell\right\}_{\ell=1}^{N_k}$ at each time step to be used by the tracker. The measurements consists of, if detected, the target detection and also the background clutter.

The measurement model of the true target [32] is obtained as

$$z_k = Hx_k + v_k \tag{3.11}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.12}$$

where H is the measurement matrix, v_k is the measurement noise representing a white Gaussian noise with zero mean and the measurement covariance matrix R, i.e., $v_k \sim N(0, R)$, where

$$R_k = E\left\{v_k v_k^T\right\} \tag{3.13}$$

$$E\left\{v_k\right\} = 0 \tag{3.14}$$

$$E\left\{v_k v_l^T\right\} = 0 \quad \text{if} \quad k \neq l. \tag{3.15}$$

The measurement noise v_k is assumed to be independent of the process noise w_k.

In polar coordinates, the measurement originated from the target is defined by:

$$z_k = h(x_k) + v_k \tag{3.16}$$

where

$$\begin{aligned} h(x) &= \begin{bmatrix} \sigma_R \\ \sigma_{Az} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(\frac{y}{x}) \end{bmatrix} \end{aligned}$$

where σ_R denotes the standard deviation for range that is the range measurement error in meters and σ_{Az} is the azimuth measurement error in degrees.

The measurement errors are provided in X-Y plane as σ_x (in meters) and σ_y (in meters) and are integrated into the measurement covariance matrix R in Cartesian by

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{3.17}$$

where σ_x^2 indicates the variance in x direction and σ_y^2 indicates the variance in y direction.

The measurement error variance is generally used to describe the known or unknown errors originated from the detection device or from statistical variations. A measurement error variance defined in X-Y plane is the deviation within a certain tolerance from the true target position. For example, if the measurement error variance is 20 m and 20 m in x and y directions, the true target location is assumed to be reported to the tracker with a dispersion by ± 20 m error in X-Y plane.

3.2.2.1 Background clutter

In modelling the background clutter two different approaches can be used: i) Parametric algorithms, ii) Non-parametric algorithms.

Parametric algorithms assume prior knowledge of the clutter density. To include this density in designing trackers, firstly the statistical model of the clutter density has to be obtained. Then the tracker can be tuned to use this knowledge to improve the tracking quality. However, because the clutter density is heavily dependent on terrain characteristics, weather conditions, man-made disturbances, it is quite restricting to assume prior knowledge of the clutter density. It is a better idea to estimate the density online as the tracking progresses. This leads to the non-parametric approach. When there is no prior knowledge, it is best to assume a uniform density in the validation region which is evaluated as:

$$\lambda = \frac{\hat{N}_k}{V_k} \tag{3.18}$$

where N_k is the number of observations included in the gate and V_k is the volume of the gate. Therefore, the observations which form the clutter are assumed to be uniformly distributed and the nonparametric approach is used to obtain the clutter density in the thesis.

Now that the description of the motion models and the measurement models have been completed, the theoretical background for the fusion of the overall state estimate both in the IMM and AMM algorithms is provided in the next section.

3.3 MODEL AND STATE ESTIMATION USING MULTI MODELS

This section describes the MAP and MMSE methods which are widely used to find estimates of random variables. These methods are also used in the AMM and IMM structures.

The development in this section is derived from the discussion in [33]. Let us define a hybrid random variable (RV)

$$\xi = (x, m) \tag{3.19}$$

where x is a continuous RV and m is a discrete random variable. Let z denote the observations of this hybrid random variable. Then, the joint mixed (pdf, pmf) given the observations is

$$p(x, m|z) = f(x|m, z)p(m|z). \tag{3.20}$$

The MAP estimations of the discrete random variable m and the continuous random variable x are defined as

$$\hat{m}^{\text{MAP}} = \underset{m}{\text{argmax}}\ p(m|z) \tag{3.21}$$

$$\hat{x}^{\text{MAP}}(\hat{m}^{\text{MAP}}) = \underset{x}{\text{argmax}}\ f(x|z, \hat{m}^{\text{MAP}}) \tag{3.22}$$

respectively. It is important to note that other MAP estimators can also be defined such as

$$(\hat{x}^{\text{JMAP}}, \hat{m}^{\text{JMAP}}) = \underset{(x,m)}{\text{argmax}}\ p(x, m|z) \tag{3.23}$$

where JMAP denotes joint MAP estimation.

The MMSE estimation of the continuous random variable x is defined as

$$\hat{x}^{\text{MMSE}} = E[x|z]. \tag{3.24}$$

The MMSE and MAP criteria minimize the following cost functions respectively

$$C_{\text{MMSE}}(x - \hat{x}) = (x - \hat{x})^2 \tag{3.25}$$

$$C_{\text{MAP}}(x - \hat{x}) = \lim_{\epsilon \to \infty} \mathbf{1}(|x - \hat{x}| - \epsilon) \tag{3.26}$$

where

$$\mathbf{1}(|x - \hat{x}| - \epsilon) = \begin{cases} 0 & \text{if } |x - \hat{x}| < \epsilon \\ 1 & \text{if } |x - \hat{x}| > \epsilon \end{cases} \tag{3.27}$$

and $\hat{x}$ indicates the expected x in terms of MMSE.

3.3.1 MMSE and MAP estimations in the AMM tracker

Model sequence of an AMM tracker through time k is given in Figure 3.2.

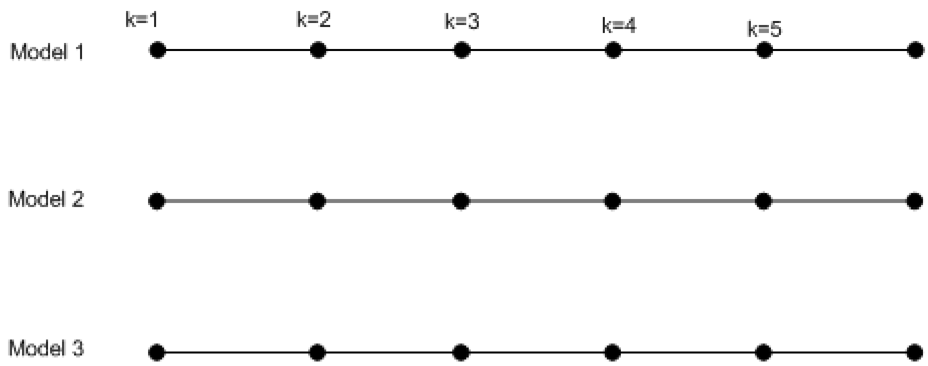

Figure 3.2: A generic sequence in the AMM algorithm

MMSE: In the AMM, in Figure 2.2, the MMSE estimator of the state x is defined as

$$\begin{aligned} \hat{x}_{k|k}^{\text{MMSE}} &= E[x_k|z_k] \\ &= \sum_{i=1}^{m} E[x_k|z^k, m_{(i)}^k] P\left\{m_{(i)}^k|z^k\right\} \\ &= \sum_{i=1}^{m} \hat{x}_{k|k}^{(i)} \, \mu_k^{(i)} \end{aligned} \tag{3.28}$$

where,
$z^k = (z_1 \ldots z_k)$ are the measurements,
$\mu_k^{(i)}$ is the posterior mode probability,
$\hat{x}_{k|k}^{(i)}$ is the MMSE estimation from i^{th} filter assuming $m_{(i)}$ is true through time. Because the state estimate is a weighted average of the model outputs, it does not make sense to define a MMSE output of the model m.

MAP: The MAP estimates of the state x and the model m are defined as follows

Let the mixed pdf-pmf of the base state and mode at time k be

$$\begin{aligned} p(x_k, m_k|z^k) &= f(x_k|z^k, m_k)p(m_k|z^k) \\ &= \left\{f_{(i)}\ (x_k|z^k)\ \mu_k^{(i)},\ i \le m\right\} \end{aligned} \tag{3.29}$$

where $f_{(i)}(x_k|z^k) = f(x_k|z^k, m_{(i)}^k)$ is the density assuming the mode sequence is $m_1^{(i)}, m_1^{(i)}, \ldots, m_k^{(i)}$ i.e., $m^{(i)}$ the true model.

From the total probability theorem, the base state posterior mixture density is

$$\begin{aligned} f(x_k|z^k) &= \sum_{i=1}^{m} f(x_k|z^k, m_{(i)}^k)\ p(m_k^{(i)}|z^k) \\ &= \sum_{i=1}^{m} f_{(i)}(x_k|z^k)\mu_k^{(i)}. \end{aligned} \tag{3.30}$$

Then, the MAP estimate of the model m is

$$\hat{m}_{\text{MAP}}^k = (\hat{m}_1, \ldots, \hat{m}_k) \tag{3.31}$$

where

$$\begin{aligned} \hat{m}_1 = \hat{m}_2 = \ldots = \hat{m}_k &= \hat{m}_k^{\text{MAP}} \\ &= \underset{m^{(i)}}{\operatorname{argmax}}\ \mu_k^{(i)}. \end{aligned} \tag{3.32}$$

Note that the MAP estimations of the model at all times from 1 to k are identical, hence we have a constant sequence of this mode throughout time from 1 to k.

It follows that the MAP estimate of the position is

$$\hat{x}_{k|k}^{\text{MAP}}(\hat{m}_k^{\text{MAP}}) = \underset{x_k}{\operatorname{argmax}} f_{(i)}(x_k|z^k) \tag{3.33}$$

if $\hat{m}_k^{\text{MAP}} = m^{(i)}$. This is the peak location of the component density $f_{(i)}(x_k|z^k)$ corresponding to the model $\hat{m}_k^{\text{MAP}}$.

3.3.2 MMSE and MAP estimations in the IMM tracker

Notationwise, let us define the generic event of a model sequence $m_{(i^k)}^k$ as

$$\begin{aligned} m_{(i^k)}^k &= m_{(i_1, i_2, \ldots, i_k)}^k \\ &= \left\{m_1^{(i_1)}, m_2^{(i_2)}, \ldots, m_k^{(i_k)}\right\}. \end{aligned} \tag{3.34}$$

For illustration purposes an example model sequence in the IMM tracker of Figure 2.4 is given in Figure 3.3 where the sequence of $m_{(1,2,1,3)}^4$ is highlighted in bold. Here, $k = 4$ and there are 3 models in the IMM algorithm resulting in three possible model realizations at each time.

MMSE: In the IMM target tracking state estimation, the MMSE estimator of the position x is defined as

$$\begin{aligned} \hat{x}_{k|k}^{\textbf{MMSE}} &= E[x_k|z^k] \\ &= \sum_{i^k \in M^k} E[x_k|z^k, m_{(i_k)}^k]\ P\left\{m_{(i_k)}^k|z^k\right\} \\ &= \sum_{i^k \in M^k} \hat{x}_{k|k}^{(i^k)} \mu_{(i^k)}^k \end{aligned} \tag{3.35}$$

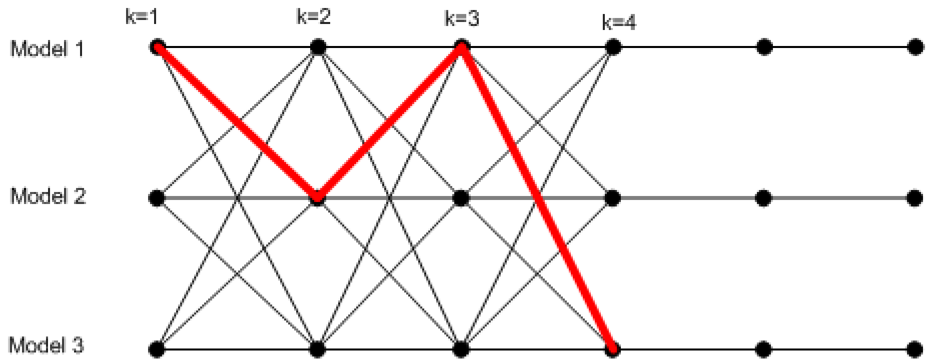

Figure 3.3: An example of model sequences in the IMM algorithm

$$
\begin{aligned}
&= \sum_{i \in M} E[x_k|z^k, m_k^{(i)}] \;\; P\left\{m_k^{(i)}|z^k\right\} \\
&= \sum_{i \in M} \hat{x}_{k|k}^{(i)} \, \mu_k^{(i)}
\end{aligned}
\tag{3.36}
$$

where,
$\mu^k_{(i_k)} = P[m^k_{(i_k)}|z^k]$ is the posterior mode-sequence probability,
$\hat{x}^{(i^k)}_{k|k} = E[x_k|z^k, m^k_{(i^k)}]$ is the conditional MMSE estimate assuming sequence $m^k_{(i_k)}$ is true,
$\mu^{(i)}_k = P\left\{m^{(i)}_k|z^k\right\}$ is the posterior mode probability,
$\hat{x}^{(i)}_{k|k} = E[x_k|z^k, m^{(i)}_k]$ is the conditional MMSE estimation assuming model $m^{(i)}$ is in effect at time k.

MAP: The MAP estimates of the state m and the model x are defined respectively as follows

$$
\begin{aligned}
\hat{m}^{k|k}_{\text{MAP}} &= \underset{m^{(i^k)}}{\text{argmax}} \; \{\mu^k_{(i^k)}, i^k \in M^k\} \\
&= (\hat{m}_{1|k}, \hat{m}_{2|k}, \ldots, \hat{m}_{k|k})^{\text{MAP}}
\end{aligned}
\tag{3.37}
$$

$$
\hat{x}^{\text{MAP}}_{k|k} \left(\hat{m}^{k|k}_{\text{MAP}} = (\hat{m}_{1|k}, \hat{m}_{2|k}, \ldots, \hat{m}_{k|k})^{\text{MAP}}\right) = \underset{x_k}{\text{argmax}} \; f_{(i^k)}(x_k|z^k)
\tag{3.38}
$$

where the right hand side of the last equation is evaluated for $\hat{m}^{k|k}_{\text{MAP}}$. Note that $\hat{m}^{k|k}_{\text{MAP}}$ is a sequence estimate. The last element in the sequence, namely, $\hat{m}_{k|k}$ is taken as the MAP estimate at time k. The MAP sequence $\hat{m}^{k|k}_{\text{MAP}}$ is used then to obtain $\hat{x}^{\text{MAP}}_{k|k}$ which turns out to be the maximum of the posterior density when the mode sequence in effect is $\hat{m}^{k|k}_{\text{MAP}}$.

In summary, the MAP estimate of the model at time k is the one with maximum mode probability and the MAP estimate of the state x is the maximum of the posterior density coming from the model $\hat{m}^{k|k}_{\text{MAP}}$.

3.4 COMPOSITE ESTIMATION USING AMM and IMM TECHNIQUES TOGETHER

Figure 3.4 illustrates the new composite approach to the state estimation. In this approach there are n branches of IMM estimators that consist of the branches of the AMM estimator. In an AMM estimator, there are n branches of filters that correspond to different motion models. Hence each filter outputs the state estimation that is the result of the Kalman Filter using the appropriate motion model. In AMM filtering there is no interaction in branches to initialize the filters in the next time step. In the composite IMM-AMM approach described here, the n basic filters (state estimators) are replaced by n IMM state estimators. In compliance with the AMM approach, the n IMM filters do not interact. The output of the AMM estimator can be computed using various estimators, the most appropriate of which in ground target tracking seem to be MMSE and MAP estimators. It turns out that this estimation structure provides a computationally efficient, responsive technique to track ground targets, the details of which are explained in the next chapters.

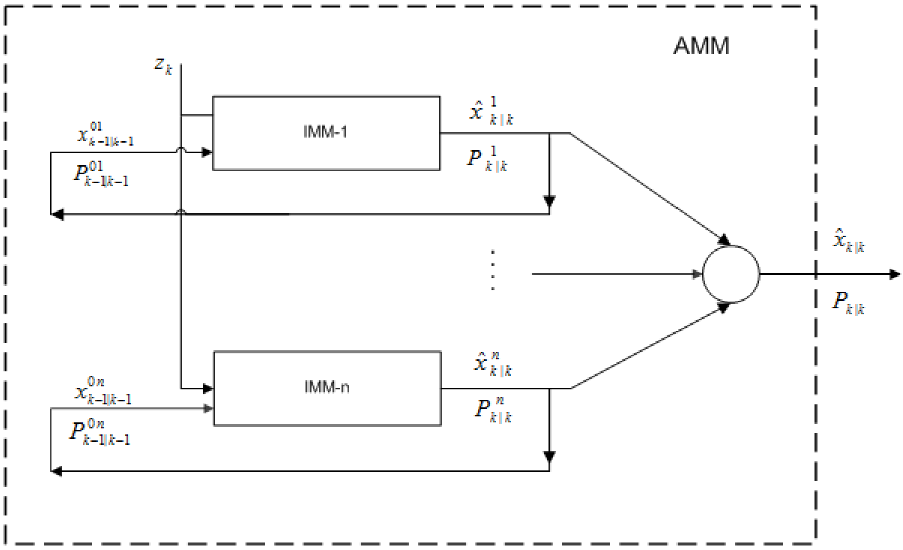

Figure 3.4: Composite IMM-AMM Estimation Structure

3.5 INCORPORATION OF VARIABLE STRUCTURES IN THE COMPOSITE ESTIMATION

The composite estimation structure can further be modified to include variable structure multi model techniques. This is achieved in two ways which extend the composite IMM-AMM estimation structure to a flexible ground.

- **Variable structure IMM included in branches:** Variable Structure IMM techniques can be used in the IMM branches in the composite structure, i.e., Any or all of the branches in Figure 3.4 can be replaced with VSIMM branches.
- **Variable structure AMM to add/drop IMM branches:** The composite structure can have a variable number of branches in Figure 3.4 which results in a variable structure AMM algorithm (VS-AMM).

3.6 VALIDATION REGIONS IN CLUTTER FOR COMPOSITE ESTIMATION

Ground target tracking surely includes variable structure techniques which are tuned to topographic constraints, road constraints and so forth. Taking those constraints into account, the validation regions (gates) where the true measurement from target together with some unwanted clutter measurements lie, can be modified as well. The purpose of this modification is to let a minimum number of clutter measurements in gates. The most common form of gating is performed by using a chi-square test as described in Section 2.3.1.

Other gating approaches can be developed, referred to as “smart gating" which use a priori information such as road constraints and topographic constraints. For example, if the target is moving along a mountainous road we are sure that the validation region must not include far away regions from the road. The validation region to the sides of the road must only include the region that the measurement uncertainty defines, as seen in Figure 3.5. The width of the triangle is determined by the measurement uncertainty of the radar. For example if the variance of the measurement uncertainty is 20m, the width of the rectangle is 40m plus the width of the road.

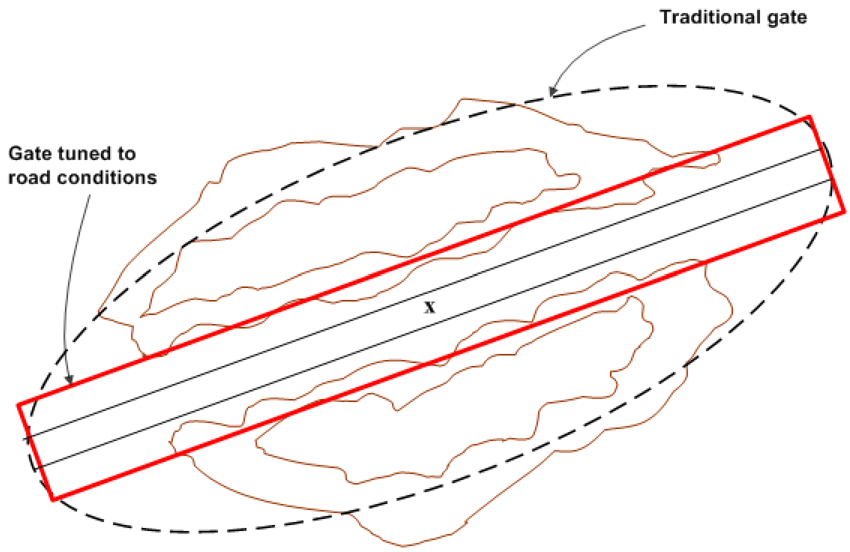

Figure 3.5: The rectangle indicates the region where the target detection is likely to fall

The target may move in constant velocity or accelerate during onroad motion. When there is a sudden increase in the velocity and when there are topographic constraints, we assume that the target would move along the road and would not leave the road position. Therefore, we may add an offset value as seen in Figure 3.6 and try to catch the target along the road.

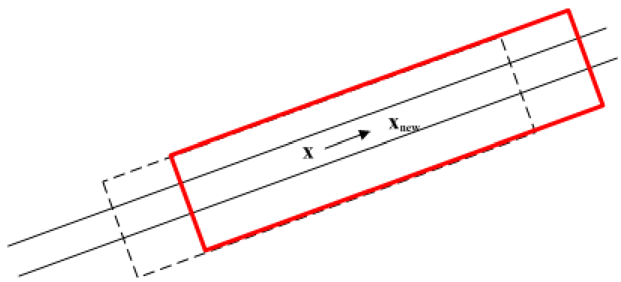

Figure 3.6: Adding an offset to the gate center

A relation between gate volume and velocity can be developed as well. When the velocity is small an assumption can safely be introduced which limits the maximum distance that the target can travel as shown in Figure 3.7. Hence, when the velocity is small, the gate size can be chosen small as well.

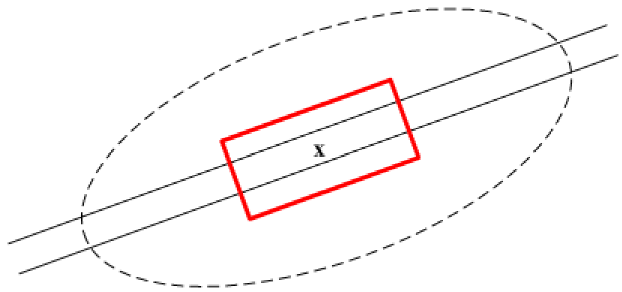

Figure 3.7: The length of the rectangular gate is selected smaller when the velocity of the target is small

In composite estimation, these gates are incorporated in the appropriate branches. For example, if one of the IMM branches represent a road segment where a bridge or mountainous region with no off road exits appear, the gating there is performed accordingly.

CHAPTER 4

VARIABLE STRUCTURE COMPOSITE IMM-AMM TRACKING ALGORITHM

This chapter describes the VS-A-IMM single target tracking algorithm in detail. The algorithm originates from the composite estimation framework discussed in Chapter 3. In developing the VS-A-IMM algorithm, various ingredients has been used on top of the estimation framework. These are the IPDAF algorithm, stop model, JPDAF algorithm, variable structure algorithm and smart gating. The VS-A-IMM algorithm is described in a structured way, putting a building block on top of another starting from the basic building block PDAF algorithm. After the development of the algorithm is completed, the operation of the algorithm is discussed and the control algorithms that pulls everything together in the VS-A-IMM algorithm is elaborated in detail.

The standard IMM-PDAF and VSIMM-IPDAF algorithms and their implementation detail are described in addition to the development of VS-A-IMM as well.

4.1 BRANCHES OF THE COMPOSITE IMM-AMM TRACKER

VS-A-IMM algorithm is developed by adding or deleting more than one branches that includes IMM/VSIMM IPDAF algorithms according to the topographic constraints or target motion.

The development of the VS-A-IMM tracker starts from the incorporation of the basic IMM-PDAF algorithm in the composite estimation framework. Firstly, it is assumed that the track exists, which is relaxed later in the development.

4.1.1 IMM-PDAF Algorithm

The IMM-PDAF is a multiple model algorithm used in the tracking of a single maneuvering target in the presence of the clutter.

The IMM-PDAF combines M number of PDAFs and the outputs of each PDAF is combined with the IMM algorithm to obtain the overall estimate as illustrated in Figure 4.1. The implementation details of the IMM-PDAF algorithm is as follows:

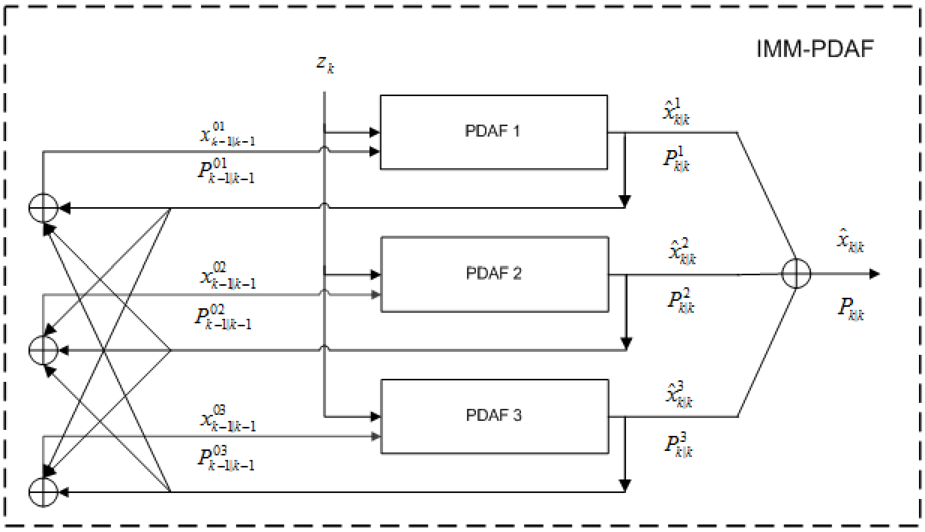

Figure 4.1: The structure of IMM-PDAF algorithm with three motion models

4.1.1.1 PDAF

PDAF is a suboptimal algorithm developed by Bar Shalom [5]. Figure 4.2 focuses the one branch of the IMM-PDAF structure shown in Figure 4.1.

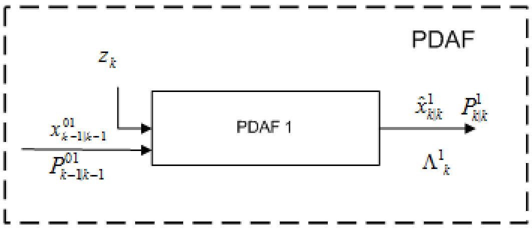

Figure 4.2: The PDAF branch in the IMM-PDAF algorithm

Each PDAF consists of one motion model and produces one state estimation and associated covariance in addition to calculation to the likelihood and therefore mode probability to be used in the IMM algorithm.

As it can be seen from Figure 4.2, PDAF takes initial state estimates and initial associated covariance evaluated as a result of the interaction and the measurement set for the current time step as an input and yields the filtered state estimate, filtered covariance and the likelihood to be used in the IMM portion of the composite IMM-PDAF algorithm. The algorithm is based on the assumption that there is at most only one target among the observations and target already exists. Due to this assumption, it is used for track maintenance.

The PDAF algorithm is specified as follows:

Step 1: Obtain initial measurement prediction

Let $\hat{z}_{k|k-1}$ be the initial measurement prediction and innovation covariance S_k evaluated as in Eq. 2.9.

Step 2: Obtain new measurements at each time step

The innovation of each measurement $(z_k^\ell - \hat{z}_{k|k-1})$ is obtained and used in statistical distance or normalized distance. The normalized distance d_ℓ^2 is then obtained as the follows:

$$d_\ell^2 = \left(z_k^\ell - \hat{z}_{k|k-1}\right)^T \underbrace{S_k^{-1}}_{\text{innovation covariance}} \underbrace{\left(z_k^\ell - \hat{z}_{k|k-1}\right)}_{\text{innovation}}. \tag{4.1}$$

The gate threshold γ calculated in each step [8] as follows:

$$\gamma = 2\ln\left[\frac{P_d}{(1-P_d)(2\pi)^{M_d/2}\sqrt{(|S_k|)}}\right] \tag{4.2}$$

where P_d denotes the probability of detection, M_d is the dimension of the measurement, S_k is the innovation covariance.

Then, the statistical distance of each measurement is compared if it is below the gate threshold as:

$$d_\ell^2 \leq \gamma. \tag{4.3}$$

The observations which are less than the gate threshold are chosen as the validated measurements to be utilized at the input of the PDAF box in Figure 4.2. The number of measurements are expressed as N_k.

Therefore, the set of measurements which falls in the validation region at scan k is denoted as follows:

$$z_k = \left\{z_k^\ell\right\}_{\ell=1}^{N_k}. \tag{4.4}$$

The performance of tracking is directly influenced by the validated measurements. A wise selection of the measurements increase the tracking performance.

Step 3: Find probabilities of each hypothesis

This step calculates the weights that project the contribution of each measurement at the output of the PDA filter. Hypotheses arise from the assumption of the PDAF algorithm such that at most one of the measurements in the validation gate is originated from the true target and the other measurements in the gate are false measurements. This assumption can be interpreted by two hypotheses as follows:

3.A. The probability that none of the validated measurements is target originated

The probability or weight of the hypothesis $\acute{\beta}_k^o$ is provided as:

$$\acute{\beta}_k^o = \lambda^{N_k}(1 - P_d P_g) \tag{4.5}$$

where,
λ is the clutter density which is evaluated nonparametrically as in Eq. 3.18,
P_g is the probability that a correct target return will fall within the gate threshold,
P_d is the probability of detection,
N_k is the number of measurements relevant to the target in the gate.

3.B. The probability that the measurement ℓ in the gate is target originated Any of the validated measurements in the gate could be the true target itself. The probability of each measurement is taken into account which are used in the weighted innovation and then in the weighted state estimation. $\acute{\beta}_k^{\ell}$ is the probability that the ℓ^{th} measurement is obtained from target for $\ell = 1, \cdots, N_k$

$$\acute{\beta}_k^{\ell} = \underbrace{\lambda^{N_k - 1}}_{\text{Clutter density for } N_{k-1} \text{ measurement}} \quad \underbrace{\frac{P_d \; e^{-d_\ell^2/2}}{(2\pi)^{M/2}\sqrt{|S_k^{\ell}|}}}_{\text{Likelihood for the } \ell^{th} \text{ measurement}} \tag{4.6}$$

where d_ℓ is the Mahalanobis Distance of the ℓ^{th} measurement.

Step 4: Normalize all weights
As weights are defined as probabilities, weights are normalized to satisfy the probability axioms [20] in which the probability must lie between zero and one.

$$\beta_k^o = \frac{\acute{\beta}_k^o}{\acute{\beta}_k^o + \sum_{\ell=1}^{N_k} \acute{\beta}_k^{\ell}} \qquad \beta_k^{\ell} = \frac{\acute{\beta}_k^{\ell}}{\acute{\beta}_k^o + \sum_{\ell=1}^{N_k} \acute{\beta}_k^{\ell}}$$

Normalized weights are then used in the weighted innovation.

Step 5: Find weighted innovation
The measurement innovation $(z_k^{\ell} - \hat{z}_{k|k-1})$ projects the difference between the predicted measurement $\hat{z}_{k|k-1} = H\hat{x}_{k|k-1}$ and the measurement ℓ.

A weighted innovation $\tilde{z}_k$ is found by substituting normalized weights β_k^{ℓ} and the measurement innovation into the following equation:

$$\tilde{z}_k = \sum_{\ell=1}^{N_k} \beta_k^{\ell}(z_k^{\ell} - \hat{z}_{k|k-1}). \tag{4.7}$$

The weighted innovation is then incorporated into the state estimation filter and used in defining the covariance as well.

Step 6: Find weighted state and associated covariance

The state estimation for each PDAF is obtained as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + W_k \tilde{z}_k W_k^T \tag{4.8}$$

where W_k is the Kalman Gain given in Eq. 2.11. Once every possibility in the validation region is elaborated, the resulting state estimate projects the contribution of each measurement in the gate. If the number of measurements increases, the position error grows. If however, the gating is evaluated in a wise manner, a smaller number of measurements fall in the gate, which leads to a smaller position error.

The associated covariance is found as:

$$P_{k|k} = \beta_k^o \; P_{k|k-1} \; + \; (1 - \beta_k^o) \; \acute{P}_{k|k} + \underbrace{DP_k}_{\text{uncertainty in the data association}} \tag{4.9}$$

where

$$\acute{P}_{k|k} = P_{k|k-1} - W_k S_k W_k^T \tag{4.10}$$

and the spread of the mean is given by:

$$DP_k = W_k \left[\sum_{\ell=1}^{N_k} \beta_k^\ell (z_k^\ell - \hat{z}_{k|k-1})(z_k^\ell - \hat{z}_{k|k-1})^T - \tilde{z}_k \, \tilde{z}_k^T \right] W_k^T. \tag{4.11}$$

The associated covariance is a critical factor in the state estimation of the algorithm such that, if the covariance grows, the inverse of the innovation covariance dramatically decreases and this decrease directly increases the size of the gate threshold obtained in Eq. 4.2. This causes incorporation of more measurements into the algorithm. This might help to catch the maneuvering target, however, it causes position error to increase which is the discrepancy between the expected state estimate and the trajectory of the true target.

Step 7: The likelihood function

The likelihood function, Λ_k is the value of each measurement that originates from the Gaussian probability density function as

$$\Lambda_k = \mathcal{N}(\tilde{z}_k; 0, S_k) \tag{4.12}$$

where $\mathcal{N}(\tilde{z}_k; 0, S_k)$ indicates the Gaussian probability density function with argument $\tilde{z}_k$ (innovation), and it has zero mean and covariance S_k. It can also be expressed as cost of assigning an observation to a track (predicted measurement) as demonstrated in Figure 4.3. This means that if the likelihood value is far away from the peak of the pdf, it has a smaller likelihood.

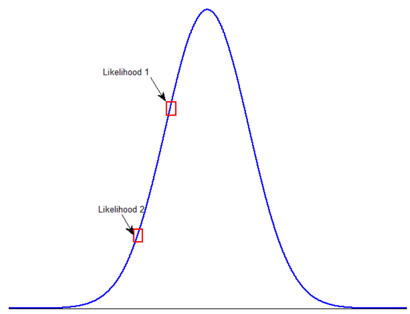

Figure 4.3: Likelihood Function: Likelihood 1 > Likelihood 2

The combination of the contribution of each likelihood in the PDA filter indicates how good the model performance is and provided as follows:

$$\Lambda_k = (1 - P_d P_g)\lambda + \sum_{\ell=1}^{N_k} \frac{P_d \, e^{-d_\ell^2/2}}{(2\pi)^{M_d/2} \sqrt{|S_k|}} \tag{4.13}$$

where, λ is the clutter density ($\lambda = \frac{N_k}{V_k}$), N_k is the number of measurements that are relevant to the target, V_k is the volume of the gate in Eq. 2.17, d_ℓ^2 is the normalized distance or Mahalanobis distance, S_k is the innovation covariance matrix for each filter at scan k, M_d is the dimension of the measurement vector. The detection probability P_d and the probability of the target falls in the gate P_g are needed to calculate likelihood as well.

Likelihood has a significant effect on the tracking performance that likelihood is used in the calculation of the mode probability, and then in the estimation of the state for each filter in the IMM/VSIMM algorithms. This helps the VS-A-IMM algorithm to select the model which has the maximum mode probability as the in-branch's state estimate. Finally, the overall state estimate projects one of the in-branch's state estimate.

This brings the discussion of the PDAF stage of the IMM-PDAF algorithm to an end. In the next section, the cooperation of the PDAF and IMM algorithms are elaborated.

4.1.1.2 IMM

The IMM is a method that provides model transition among the predetermined motion model set in a probabilistic manner along the full motion of the target and aims to adapt to maneuvering motion of the target. It assumes that the model can be changed at each time step. This model transition is modelled by Markov process.

Step 1 : Determine model probability transition matrix
A 3×3 Markov Matrix for a 3 model case is defined as follows:

$$\pi_{ij} = \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix}. \tag{4.14}$$

The sum of the probabilities at each row must be unity in the Markov Matrix for any given model. Therefore, for model j

$$\sum_{j=1}^{M} \pi_{ij} = 1. \tag{4.15}$$

Each column indicates one model in the matrix. Therefore, the 3×3 Markov matrix can be interpreted as a three-model IMM. The elements in the diagonal gives an idea about how the algorithm adheres to its model at each step. The rest of the elements in the matrix are the prior probabilities of model transition probability from one model to the other model.

Step 2: Obtain likelihoods and updated model probabilities
The Likelihood function Λ_k^j is obtained ($j = 1, \ldots, M$) using Eq. 4.13 for the measurements.

Using the Bayes' rule, the updated model probabilities become

$$\mu_k^j = \frac{\Lambda_k^j \, \mu_{k|k-1}^j}{\sum_{i=1}^{M} \Lambda_k^i \, \mu_{k|k}^i} \tag{4.16}$$

where $\mu_{k|k-1}^j$ is the predicted mode probability for model j given as

$$\mu_{k|k-1}^j = \sum_{j=1}^{M} \pi_{ij} \, \mu_{k-1}^j. \tag{4.17}$$

The predicted mode probability indicates the probability after interaction that the target is in model j.

Step 3: Calculate overall state estimate and associated covariance matrix
The overall state estimate is a weighted sum of the state estimates of each filter:

$$\hat{x}_{k|k} = \sum_{j=1}^{M} \mu_k^j \hat{x}_{k|k}^j \tag{4.18}$$

where $\hat{x}^{j}_{k|k}$ is the state estimate that PDAF produces for model j as in Eq. 4.8 and μ^{j}_{k} is the updated model probability.

The overall covariance matrix is evaluated as:

$$P_{k|k} \;=\; \sum_{j=1}^{M} \mu^{j}_{k} \, \left[P^{j}_{k|k} + \; DP^{j}_{k}\right] \tag{4.19}$$

where $P^{j}_{k|k}$ is the covariance generated as in Eq. 4.9 in the PDAF algorithm.

The term added to compensate the uncertainty in the gate DP^{j}_{k} is defined as:

$$DP^{j}_{k} = \left[\hat{x}_{k|k} - \hat{x}^{j}_{k|k}\right]\left[\hat{x}_{k|k} - \hat{x}^{j}_{k|k}\right]^{T}. \tag{4.20}$$

To clarify, this step is intended to compare the true target trajectory and the estimation and hence to calculate the error. This step is not a part of the recursive algorithm.

Step 4: Calculate mixing probabilities
The mixing probability that the target makes the transition from model i to j is given as:

$$\mu^{ij}_{k} = \frac{\pi_{ij}\,\mu^{i}_{k}}{\mu^{j}_{k|k-1}} \tag{4.21}$$

where,
μ^{i}_{k} is the probability for mode i,
$\mu^{j}_{k|k-1}$ is the predicted mode probability for model j,
π_{ij} is the model transition probability matrix that the target would make a transition from model state i to state j depending on the coefficients of the matrix.

The mixing probability is the mechanism providing the interaction amount for the model changes and is used to update the initial state estimate to be used at the next time step.

Step 5: Find mixing states and covariance estimates

The initial state estimation for model j, $\hat{x}^{0j}_{k|k}$, is calculated in order to provide an input to the PDAF in Eq. 4.8.

The initial state estimation of the model j is obtained by using mixing probabilities, μ^{ij}_{k}, and state estimates $\hat{x}^{i}_{k|k}$ that are converted into a common dimensionality

$$\hat{x}^{0j}_{k|k} = \sum_{i=1}^{M} \mu^{ij}_{k}\,\hat{x}^{i}_{k|k} \tag{4.22}$$

where $\hat{x}^{i}_{k|k}$ is the estimate for model i.

The initial covariance of the model j, $P^{0j}_{k|k}$, is given by:

$$P^{0j}_{k|k} = \sum_{i=1}^{M} \mu^{ij}_{k} \, \left[P^{j}_{k|k} \; + \; DP^{ij}_{k}\right]. \tag{4.23}$$

The spread of the mean DP^{ij}_{k} which is the difference in the state estimates from model i to j is

$$DP^{ij}_{k} = \left[\hat{x}^{i}_{k|k} - \hat{x}^{0j}_{k|k}\right]\left[\hat{x}^{i}_{k|k} - \hat{x}^{0j}_{k|k}\right]^{T}. \tag{4.24}$$

Step 6: Find predicted state and predicted error covariance

The state estimate obtained after interaction in the IMM state, also named as the mixing estimate, is used to calculate the predicted state that is used in the PDA filter as follows

$$\hat{x}^{j}_{k+1|k} = F^{j}\, \hat{x}^{0j}_{k+1|k} \tag{4.25}$$

and the associated predicted error covariance is found as

$$P^{j}_{k+1|k} = F^{j}\, P^{0j}_{k|k}\, F^{j} + Q^{j}_{k} \tag{4.26}$$

where F, G, w_{k-1}, Q_{k-1} indicate the state transition matrix, the noise matrix, the noise vector, and the process noise covariance respectively as defined in section 3.2.

The predicted state and the associated covariance carry previous states' tracking kinematics and provide the base for the current state estimation.

The measurement prediction is then evaluated as

$$\hat{z}^{j}_{k+1|k} = H^{j}\, \hat{x}^{j}_{k+1|k}. \tag{4.27}$$

The measurement prediction, also named as the predicted measurement, is then used as the center of the validation gate as mentioned in Step 1 in the PDAF algorithm.

Step 7: State prediction and covariance prediction used in gating
When there are more than one model in the IMM algorithm, forming a gate to obtain measurements at each time step can be done for each filter as well as one common gating for all filters. This step demonstrates the case of having one set of measurements for all models in the algorithm:

The predicted state $\hat{x}_{k+1|k}$ to be used in the gating at time step $(k+1)$ is evaluated in the following formula

$$\hat{x}_{k+1|k} = \sum_{j=1}^{M} \mu^{j}_{k|k} \hat{x}^{j}_{k+1|k}. \tag{4.28}$$

The predicted covariance of the target $P_{k+1|k}$ is evaluated as:

$$P_{k+1|k} = \sum_{j=1}^{M} \mu^{j}_{k|k} \left[P^{j}_{k+1|k} + DP^{j}_{k+1} \right] \tag{4.29}$$

where DP^{j}_{k+1} is the spread of the mean that is the difference in the state estimates of the model j and the predicted state.

The IMM-PDAF has been a very effective target tracking algorithm in tracking of maneuvering mode of a single target in the cluttered environment. When this algorithm is applied for ground targets, one needs to resolve track drop problem that would be encountered due to the MDV which is one of the design characteristic of the GMTI sensor used in the tracking of the ground targets. Even though the motion uncertainty problem is handled for a single target in the presence of the clutter by extending the IMM to the PDAF algorithm, move-stop motion is not taken into account in the IMM-PDAF algorithm and still requires attention.

Now that the IMM-PDAF algorithm has been developed completely, we can integrate the stop model to the IMM-PDAF algorithm used to address the track drop problem caused by stopping targets.

4.1.2 IMM-PDAF with Move-Stop Model

The GMTI sensor distinguishes the moving ground targets from background clutter by the Doppler effect and cannot detect the targets stops or move slowly under the minimum detected velocity. In rural areas, target may avoid being tracked by the sensor by deliberately stopping and moving. To tackle this problem, a state dependent stop model is described in [64, 65] by using a standard VSIMM technique. However, there are a number of shortcomings of the method:

i) The state dependent stop model [65] is not evaluated in cluttered environments.

ii) The scenario does not include any special problems in which VSIMM mostly generates position error peaks when compared to the standard IMM techniques. These are the junction, offroad exit, onroad entry cases.

The above cases are elaborated and incorporated into the IMM-PDAF algorithm to obtain the IMM-PDAF-Stop model as an intermediate step to construct the VS-A-IMM method.

Another approach is given in [43] but the details of the procedure that how stop behaviour problem is tackled and resolved is not discussed at all.

The integration of a stop model described below is inspired by the state dependent approach [65] and enhances that approach by using validation regions (gating) and by addressing junction, onroad offroad entry/exit cases.

Step 1: Define a threshold velocity level

A threshold velocity level for stop behaviour is defined to compare it with the estimated velocity of target that is obtained from the current state estimate as

$$V_{\text{stop}} = a_d T + V_0 \tag{4.30}$$

where a_d is the maximum amount of deceleration in the velocity, T is the sampling interval and V_0 is the maximum speed deviation selected as $V_0 = \sqrt{6}\sigma_0 T$.

The velocity of the target under the stop model is defined by the Rayleigh distribution and the probability density function when target is at the stop mode is evaluated as

$$f(V) = \frac{V}{\sigma_{\text{stop}} T^2} exp\left(\frac{V^2}{2\sigma_{\text{stop}} T^2}\right) \tag{4.31}$$

where σ_{stop} is the standard deviation for the stop model it must satisfy $\sigma_{stop} \leq \frac{1}{\sqrt{6}T} MDV$.

To keep the analysis simple, V_{stop} is chosen as the minimum detectable velocity.

The MDV mentioned in section 2.1 is the critical threshold for a GMTI sensor to detect moving ground targets and the actual value of the MDV is based on the signal processor embedded in the sensor [58]. It is usually assumed as between 2 m/s - 4 m/s. In this thesis the MDV is assumed as 2 m/s.

Step 2: Obtain state estimates

The state estimate is obtained from the output of the PDAF for stop model. During the stop mode of the target, IMM-PDAF with included stop model is executed whereas the IMM-PDAF is the main algorithm in the tracking (non-stop case).

Step 3: Derive the radial velocity and compare it with the MDV

The state estimate vector is comprised of the position and the velocity of the target in the X-Y plane as $\hat{x}_k = \left[x \; V_x \; y \; V_y\right]^T$ where x and y are the positions of the target in the X-Y plane respectively while V_x and V_y are the velocities in X and Y coordinates. Then, the radial velocity is found using the current estimated filter, $\hat{x}_k$, as

$$V = \sqrt{V_x^2 + V_y^2}. \tag{4.32}$$

This radial velocity is calculated at each time step and compared with the MDV. If the radial velocity does not approach to the MDV, the standard IMM-PDAF algorithm continues to provide the state estimate.

Step 4: Add stop model into the IMM model set if $V \leq$ MDV

If the target slowly moves under the MDV or stops, one of the models in the model set of the IMM algorithm is replaced with the stop model. The stop model state space vector described in Eq. 3.9 is

$$x_{k+1} = F^{\text{stop}} \, x_k + G^{\text{stop}} \, w_k \tag{4.33}$$

where the state transition matrix F and the noise matrix G respectively are given as follows

$$F^{\text{stop}} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad G^{\text{stop}} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}. \tag{4.34}$$

The process noise for the stop model w_k is assumed generally as zero. However, it is recommended that a non-zero process noise must be added to support the case that the target would not stop suddenly at one sampling period [65]. The second row in the state transition matrix causes the velocity part become zero and hence, the state estimate for the current scan is very much close to the previous state estimate.

Step 5: Obtain combined model transition probability matrix

The standard IMM algorithm utilizes one Markov matrix in which the number of columns indicates the number of models and provides the probabilities of transition from one model to another model. When the stop model is defined, the Markov matrix has to keep a zero column in the Markov matrix. However, considering that a target deceleration do not take place in one sampling interval, one more Markov matrix has to be defined to support the deceleration transitions. Therefore, target movement can be defined in two stages:

i) The slow stage $E = V \leq V_{\text{stop}}$ at which the radial velocity is under the MDV is described. The transition probability matrix for the slow stage is denoted as

$$\left[p_{ij}\right]^E = \begin{bmatrix} \pi_{00}^E & \pi_{01}^E & \pi_{02}^E \\ \pi_{10}^E & \pi_{11}^E & \pi_{12}^E \\ \pi_{20}^E & \pi_{21}^E & \pi_{22}^E \end{bmatrix}. \tag{4.35}$$

ii) The fast stage $\bar{E} = V > V_{\text{stop}}$ is the case where the radial velocity of the target is above the MDV. The transition probability matrix for the fast stage is described as

$$\left[p_{ij}\right]^{\bar{E}} = \begin{bmatrix} 0 & \pi_{01}^{\bar{E}} & \pi_{02}^{\bar{E}} \\ 0 & \pi_{11}^{\bar{E}} & \pi_{12}^{\bar{E}} \\ 0 & \pi_{21}^{\bar{E}} & \pi_{22}^{\bar{E}} \end{bmatrix}. \tag{4.36}$$

The structure of the IMM algorithm utilizes one Markov matrix only to support model transitions from one model to the other at each time step. Therefore, a combined mode transition probability matrix is required.

Step 6: Obtain a combined mode transition probability matrix

The combined probability matrix π^c_{ij} is calculated as the weighted sum of the matrices that belong the slow and fast stages

$$\pi^c_{ij} = \pi^E_{ij}\, p^{E,i}_{k-1} + \pi^{\bar{E}}_{ij}\, p^{\bar{E},i}_{k-1} \tag{4.37}$$

where π^E_{ij} and $\pi^{\bar{E}}_{ij}$ are the mode transition probabilities (Markov) matrices under E and $\bar{E}$ respectively.

The slow stage probability on mode i at scan k, updated by the Rayleigh distribution, is denoted by:

$$p^{E,i}_{k-1} = P\left\{E_{k-1}|M^i_{k-1}, z^{k-1}\right\} \tag{4.38}$$

$$= P\left\{V \leq\ V_{stop}\ |\hat{x}^i_{k-1|k-1}, P^i_{k-1|k-1}\right\} \tag{4.39}$$

where E indicates the slow stage, $\hat{x}^i_{k-1|k-1}$ and $P^i_{k-1|k-1}$ are the state estimate and the associated covariance for mode i at scan $k-1$ respectively.

The fast stage probability is obtained as follows:

$$p^{\bar{E},i}_{k-1} = 1 - p^{E,i}_{k-1}. \tag{4.40}$$

Step 7: Obtain mixing probability after stop model is activated

The resultant combined mode transition probability matrix is used to update the mixing probability in the calculation of the initial state estimate after the interaction in the IMM algorithm as

$$\mu^{ij}_k = \frac{\pi^c_{ij}\, \mu^i_k}{\mu^j_{k|k-1}} \tag{4.41}$$

where,
μ^i_k is the probability that the target is in model i as evaluated after data are received at scan k,
$\mu^j_{k|k-1}$ is the predicted mode probability,
π_{ij} is the mode transition probability matrix (Markov) in which the target would make a transition from model state i to state j.

The stop model is terminated when the target moves above the MDV and the standart IMM-PDAF algorithm comes into play to obtain the state estimates. The development of this algorithm is an important intermediate step to construct the branches of the VS-A-IMM algorithm.

In the next section, the hide model is described to further enhance the IMM-PDAF algorithm.

4.1.3 IMM-PDAF with Hide Model

Targets sometimes might be invisible to the radar due to topographic constraints such as elevation, tunnel, underground subway or tree areas which might also seem like tunnels. If a priori information is not provided to the tracker, the track is immediately dropped and then must be reinitialized properly.

If the topographic constraint is available to the tracker, one conventional approach that can be used is to wait for the target to reappear at the location where it would possibly be visible again.

In this study, a tunnel is chosen as the topographic constraint and its location is assumed to be known. Two approaches are presented that can observe the target going out of the tunnel without dropping the track.

The first one (Approach I) is based on the fact that the target of interest moves at a constant speed when it is invisible to the tracker.

The second one, Approach II, has no assumption regarding the target motion in the tunnel.

Approach I (Traditional): Constant Velocity Assumption

Approach I is a method that can track the target of interest along the topographic constraint in which the target is obscured and then seen again. This approach is incorporated in to the IMM-PDAF algorithm in which it only becomes active when the target moves close to the tunnel. It is deactivated if the target is visible after the tunnel. This algorithm is described as follows

i) The motion of the target along the tunnel would be constant and known. The direction of the movement is assumed to be only in one direction.

ii) The normalized distance between the predicted measurement and the entry point of the tunnel is measured at each time step.

iii) When the target approaches to the tunnel, the tracking process is carried out with the assumption that the target keeps moving at a constant velocity along the tunnel. No real measurement is provided to the tracker. The probability of the detection becomes zero. The state is virtually estimated by interpolation and is propagated along the movement in the tunnel and then a gate is opened at the predetermined location at the expected time step to obtain new measurements.

iv) When the target exits the tunnel, the standard IMM-PDAF algorithm is executed.

The advantage of this approach makes tracking of a ground target possible without considering track loss in its simplest way. If the target of interest does not obey the assumption, a missed track occurs.

In realistic situations, this is not the case. One can not hope or expect a driver's motion to be constant velocity if it is a suspected target which aims to mislead the tracker. To resolve this problem, a novel hide model is proposed in this study.

Approach II: Hide Model Proposed

The hide model is a simple and practical approach which supports ground target tracking if the topographic constraints are available to the tracker. The approach does not care about the speed of the target and about when it would quit the terrain obscuration. Therefore, the algorithm is independent of the time spent in topographic constraints.

The hide model is implemented in the VS-A-IMM algorithm when there is a priori information about topographic constraints like tunnels, elevation around the roads or forests.

The approach proposed here is not dependent on the assumption that the target moves in the topographic obscuration with a constant velocity and therefore resolves the target drop problem when the target is moving in the obscuration with an unpredictable behaviour.

The hide model is activated if the normalized distance between current state of the target and the tunnel is in the vicinity of the tunnel threshold.

The new approach is developed by using the on-standby and not-to-advance characteristics of the stop model and implemented into the in-branches of the VS-A-IMM algorithm.

The IMM-PDAF algorithm with Approach II is described as follows

Step 1: The state of the target is obtained by the IMM-PDAF algorithm at each time step.

Overall state estimate is found as in Eq. 4.18

$$\hat{x}_{k|k} = \sum_{j=1}^{M} \hat{x}^j_{k|k} \mu^j_k \tag{4.42}$$

Overall covariance is obtained from Eq. 4.19

$$P_{k|k} = \sum_{j=1}^{M} \mu^j_k \left[P^j_{k|k} + \left(\hat{x}_{k|k} - \hat{x}^j_{k|k} \right) \left(\hat{x}_{k|k} - \hat{x}^j_{k|k} \right)' \right]. \tag{4.43}$$

The filtered track existence probability is also gathered if the track maintenance feature is required

$$\psi_{k|k} = \sum_{j=1}^{M} \psi^j_{k|k} \mu^j_k. \tag{4.44}$$

Step 2: The normalized distance between the predicted state of the target and the tunnel starting point is found at each time step as

$$d^2 = \left(\hat{x}_{k+1|k} - x_{\text{TunnelStart}} \right)^T S_k^{-1} \left(\hat{x}_{k+1|k} - x_{\text{TunnelStart}} \right). \tag{4.45}$$

Step 3: When the target comes close to the tunnel entry, i.e.,

$$d^2 \leq \gamma_t \tag{4.46}$$

where γ_t is the threshold for tunnel, the visibility of the target is removed and hence the measurement is not available to the tracker.

A point at the output of the tunnel exit position is treated as the predicted measurement for the next time step and gating is applied as demonstrated in Figure 4.4.

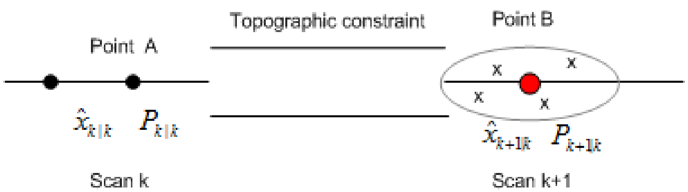

Figure 4.4: Topographic constraint on the road and the hide model mechanism.

The algorithm turns out to be the IMM-PDAF/IMM-IPDAF algorithm with stop model when there is no detection at the visible location. If there is no detection, the state and the covariance are evaluated with the IMM-PDAF/IMM-IPDAF with stop model algorithm as shown in Figure 4.5.

Step 4: When the target exits the tunnel, the GMTI radar detects the moving target. This helps tracker to change the existing algorithm from stop case to the moving case and the standard IMM-PDAF algorithm is activated.

This new approach is superior compared to the traditional Approach I because the driver's behaviour is mostly uncertain and it is impossible to predict the exact time when the target exits the tunnel. The only possible problem is that the tracker might perceive the movement later than the time the target appears at the tunnel exit and hence the tracker might stay in stop mode for sometime.

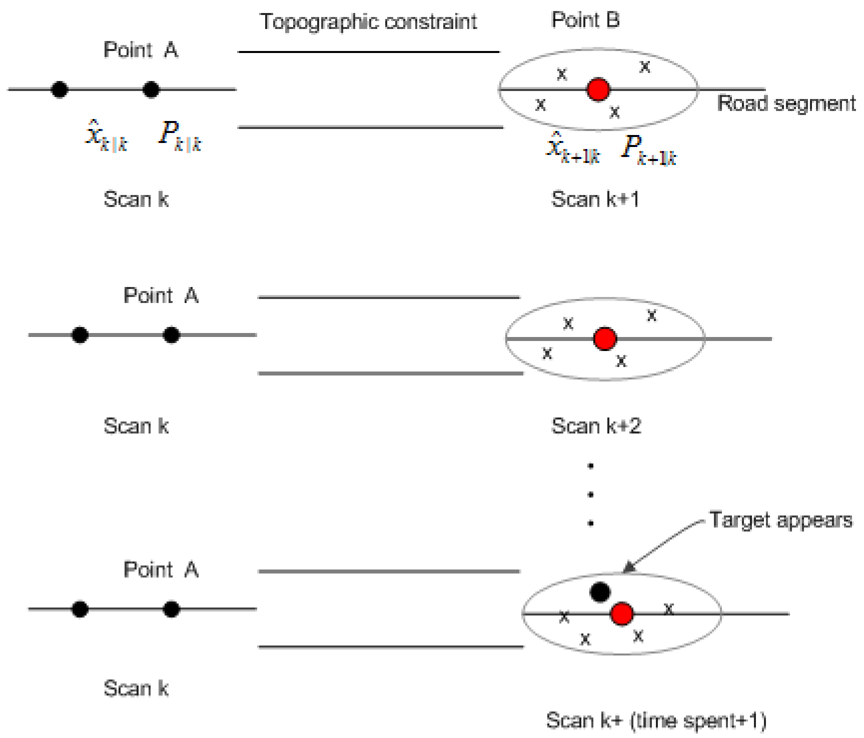

Figure 4.5: Demonstration of Approach II until a detection occurs

4.1.4 JPDAF

The JPDAF is a data association and filtering algorithm to track multiple targets in the presence of the clutter. When there are several targets close to each other, a measurement from one target can fall inside the validation region of a neighbouring target. Assumptions of the JPDAF algorithm are provided in section 2.7.

In the JPDAF algorithm, targets in close proximity are not treated independently. If two or more targets have nonzero probability of the same measurement, then the JPDA calculation of each target is dependent on the others.

The JPDA algorithm is described as follows:

Step 1: Obtain initial measurement prediction

Let $\hat{z}^j_{k|k-1}$ be the initial measurement prediction and innovation covariances that belong to each target evaluated as in Eq. 2.9.

Step 2: Obtain measurements

Normalized distance or Mahalanobis distance of each measurement is evaluated and then is compared to the related targets gate threshold as in

$$d^2_{j\ell} = \left(z^{j\ell}_k - \hat{z}^j_{k|k-1}\right)^T (S^j_k)^{-1} (z^{j\ell}_k - \underbrace{\hat{z}^j_{k|k-1}}_{H^j\ \hat{x}^j_{k|k-1}}) \leq \gamma_j \tag{4.47}$$

where $\hat{z}^j_{k|k-1}$ is the predicted measurement for target j at scan k,
S^j_k is the residual covariance matrix for the target j at scan k,
$d^2_{j\ell}$ is the normalized distance between the ℓ^{th} measurement for j^{th} target,
$z^{j\ell}_k$ is the measurement ℓ for target j,
γ_j is the gate threshold for the target j.

The gate threshold [8] for each target of interest is found as

$$\gamma_j = 2\ln\left[\frac{P_d}{(1-P_d)(2\pi)^{M/2}\sqrt{(|S^j_k|)}}\right] \tag{4.48}$$

and then, distances of each measurement concerning to target j are compared with the gate threshold specified for target j, γ_j, calculated in each step [8] as follows:

$$d^2_{j\ell} \leq \gamma_j. \tag{4.49}$$

The gating around the predicted measurement for each target is formed. Therefore, the set of measurements which falls in the validation regions of each target at scan k are determined. A typical overlapping validation regions of two target is demonstrated in Figure 4.6:

When there are multiple targets in which their validation regions do not intersect, the JPDAF algorithm is handled as in PDAF case in section 4.1.1.1. However, if the overlapping gates are formed, the calculation of hypotheses are performed in a sophisticated manner.

In order to explain the matrix clearly, let the number of targets and the measurements distributed in the validation regions become as in Figure 4.6. There are two targets and three measurements, two of which is shared by two validation regions.

Step 3: Form a hypothesis matrix

A hypothesis matrix is formed at which all the possibilities are taken into account and Table 4.1 shows all sets of feasible measurement to track assignments. The logic of the table can be explained as: i) Hypothesis H1 assumes that none of the measurements belong to Track 1 or Track 2, ii) Hypothesis H3 assumes that the measurement indicated as 2 would possibly belong to target 1 whereas none of the measurements are relevant to the target two. Counter assumption H5 is also taken into account that the measurement 2 may belong to target 2 whereas target 1 has no measurements.

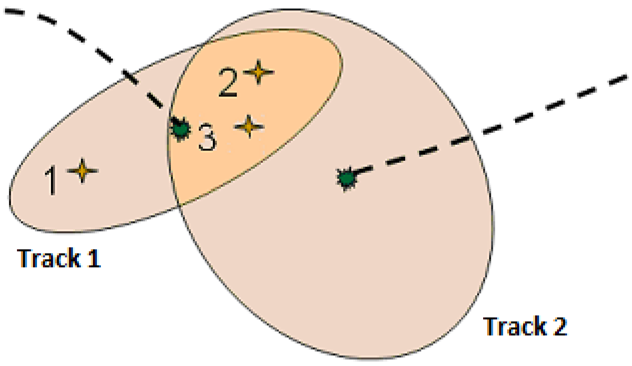

Figure 4.6: Two targets and their validation region

For clarification: A measurement which exists in the intersection region of two validation regions can be assigned to only one target for each hypothesis as in Table 4.1.

Table4.1: Hypothesis matrix

Hypothesis No	Track 1	Track 2
H1	0	0
H2	1	0
H3	2	0
H4	3	0
H5	0	2
H6	1	2
H7	3	2
H8	0	3
H9	1	3
H10	2	3

Step 4: Find probabilities of each hypothesis in the hypothesis matrix

The probability of each hypothesis which reflects each measurement to track association [8] is described as

$$\acute{P}_H = \prod_{\ell=1}^{N} g_{j\ell}\lambda_{\text{total}}^{L-N} \tag{4.50}$$

where λ_{total} is the joint clutter density, L is the number of measurements associated to tracks, N is the number of all measurements, $g_{j\ell}$ is the Gaussian Likelihood Function associated with the assignment of measurement ℓ to target j, i.e.,

$$g_{j\ell} = \begin{cases} (1 - P_d) & \lambda_{\text{total}} = 0 \\ P_d \dfrac{e^{-d_{j\ell}^2/2}}{(2\pi)^{M/2}\sqrt{|S_k^j|}} & \text{otherwise} \end{cases} \tag{4.51}$$

where S_k^j is the innovation covariance matrix for track j at scan k.

If two targets have nonzero probability of the same measurement, then the JPDAF calculation of each target is also based on the other target.

Step 5: Find normalized probabilities

Likelihoods of the hypotheses are normalized to obtain one as a sum of all hypotheses

$$P_{H_m} = \frac{\acute{P}_{H_m}}{\sum_{m=1}^{10} \acute{P}_{H_m}}. \tag{4.52}$$

Table 4.2 shows all sets of feasible measurement to track assignments and corresponding a posteriori probabilities of each hypothesis.

Table4.2: A posteriori probabilities for the possible hypothesis

Hypothesis No	Track 1	Track 2	Probability
H1	0	0	$P_{H1} = (1 - P_d)^2\beta^3$
H2	1	0	$P_{H2} = g_{11}P_d(1 - P_d)\beta^2$
H3	2	0	$P_{H3} = g_{12}P_d(1 - P_d)\beta^2$
H4	3	0	$P_{H4} = g_{13}P_d(1 - P_d)\beta^2$
H5	0	2	$P_{H5} = g_{22}P_d(1 - P_d)\beta^2$
H6	1	2	$P_{H6} = g_{11}g_{22}P_d^2\beta$
H7	3	2	$P_{H7} = g_{13}g_{22}P_d^2\beta$
H8	0	3	$P_{H8} = g_{23}P_d(1 - P_d)\beta^2$
H9	1	3	$P_{H9} = g_{11}g_{23}P_d^2\beta$
H10	2	3	$P_{H10} = g_{12}g_{23}P_d^2\beta$

Step 6: Compute the probabilities of each measurements on the targets

In order to compute the probability that measurement ℓ should be assigned to target j, a sum is taken over the probabilities from those normalized hypotheses probabilities.

The probability of assignments of measurement to target 1 $p_{1\ell}$ where measurements are $\ell = 0, 1, 2, 3$ can be specified as follows:

- The probability that none of the measurements are assigned to target 1, p_{10}, is obtained as:

$$p_{10} = P_{H_1} + P_{H_5} + P_{H_8}. \tag{4.53}$$

- The probability that measurement 1 is likely originated from target 1 p_{11} is evaluated as

$$p_{11} = P_{H_2} + P_{H_6} + P_{H_9}. \tag{4.54}$$

The probabilities that measurement 2 and 3 are reflected to target 1 are obtained as follows in the same way

$$\begin{aligned} p_{12} &= P_{H_3} + P_{H_10} \\ p_{13} &= P_{H_4} + P_{H_7}. \end{aligned}$$

The probabilities or weights for target 2 $p_{2\ell}$, where measurements are $\ell = 0, 1, 2, 3$ are calculated in the same manner

$$\begin{aligned} p_{20} &= P_{H_1} + P_{H_2} + P_{H_3} + P_{H_4} \\ p_{21} &= 0 \\ p_{22} &= P_{H_5} + P_{H_5} + P_{H_7} \\ p_{23} &= P_{H_8} + P_{H_9} + P_{H_10}. \end{aligned}$$

The probabilities indicate the weight of each measurement on the filtering process to obtain the state estimate and the covariance for the target of interest.

Step 7: Obtain state estimate

Once the weights/likelihoods of measurements are computed, these are then combined into the state estimate and each target state and its associated covariance are updated as in the PDA algorithm.

The JPDAF algorithm is then combined with the IMM algorithm as IMM-JPDAF to check if there is a possible track in the environment. Therefore, the measurements are treated as a potential track for track initiation at each scan [6].

The derivatives of IMM based algorithms IMM-PDAF, IMM-PDAF with stop model included, IMM-PDAF hidden model and finally IMM-JPDAF are discussed. All of the algorithms discussed above assume that the target already exists which means that the existence of the target is not evaluated during the full motion of the target and in turn the tracker is informed. The algorithm does not consider if any new target is available around or the target is dropped, and also the algorithm suddenly awaits target information to be provided after the track is initialized by other mechanisms. To tackle this problem, another technique is developed to initialize the target by the contradiction approach [3]. This technique incorporates a target detectability parameter. If this value becomes low at any time step, it assumes that the target does not exist.

The integrated PDAF and the IMM-IPDAF algorithms are developed [14, 47, 46] as well to address the track existence probability parameter which is discussed in the forthcoming section.

4.1.5 IMM-IPDAF Algorithm

The IMM-IPDAF algorithm provides the track formation and maintenance for the tracking of maneuvering ground targets in clutter. The initiation of the true target, especially in clutter, introduces another concept known as “track quality” in the extension of the IMM-PDAF algorithm. The track quality is modelled by using the track existence probability as a random variable [47].

The algorithm IMM-IPDAF uses a group of IPDAF filters in the IMM structure as illustrated in Figure 4.7.

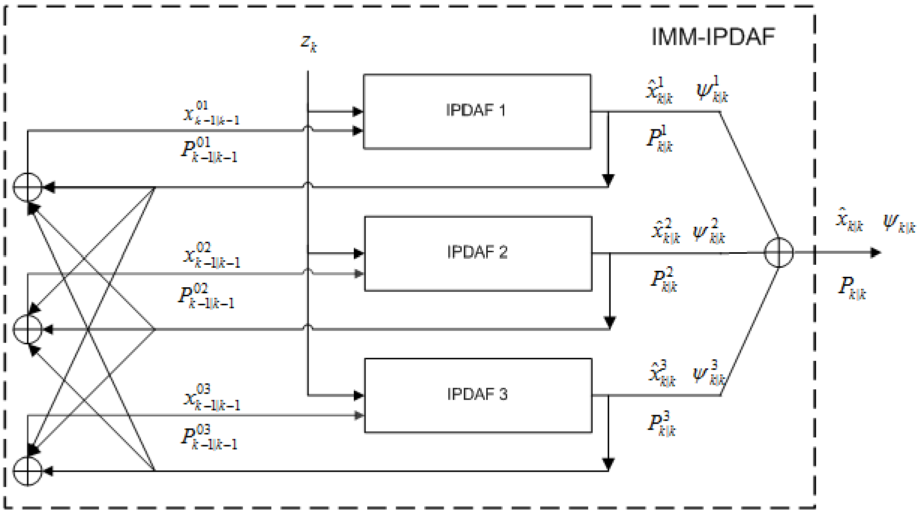

Figure 4.7: The structure of the IMM-IPDAF algorithm with three model

4.1.5.1 Track Formation for IMM-IPDAF Algorithm

The track initiation has a significant influence of the tracking of a true target when there is clutter in the tracking environment false track initiations can be encountered which causes the miss of the target. Therefore, initially false detections and the true target itself are assumed as the initial track candidates and are subject to the sequential ratio probability test (SPRT) for determining the true target at each time step. There are different SPRT methods in the literature [8, 52, 60]. However, the approach of [8] is used in this thesis for track formation and maintenance. The initiation algorithm is handled in the following manner:

The track existence probability $\psi_{k|k}$ is evaluated by Markov process at each time step

$$\psi_{k|k} = P\left\{\chi_k, z^k\right\} \tag{4.55}$$

and then is used to update the likelihood ratio (LLR) [8, 60], which provides the statistic by comparing the two hypothesis based on the given observations z_k described as follows

$$\text{LLR} = \ln\left(\frac{\psi_{k|k}}{1-\psi_{k|k}}\right). \tag{4.56}$$

After having evaluated the likelihood ratio, track confirmation and deletion thresholds are determined by the Sequential Probability Ratio Test (SPRT) as below

$$\text{LLR} \geq T_2 \qquad \text{declare track confirmation} \qquad T_2 = \ln\left(\frac{1-\beta}{\alpha}\right) \tag{4.57}$$

$$T_1 \leq \text{LLR} \leq T_2 \qquad \text{continue test} \tag{4.58}$$

$$\text{LLR} \leq T_1 \qquad \text{delete track} \qquad T_2 = \ln\left(\frac{1-\alpha}{\beta}\right) \tag{4.59}$$

where α is the probability of false track confirmation and β is the is the probability of true track deletion. Considering that the target is once initialized or deleted, maintenance of the track can be discussed in the next section.

4.1.5.2 Track Maintenance for IMM-IPDAF Algorithm

The IPDAF is a data association algorithm which takes the track existence probability into account used in the tracking of a single target. A bank of IPDAF filters illustrated in Figure 4.8 are integrated in the IMM structure. Apart from PDAF in Figure 4.2, initial track existence probabilities for the assumptions are fed to the IPDAF.

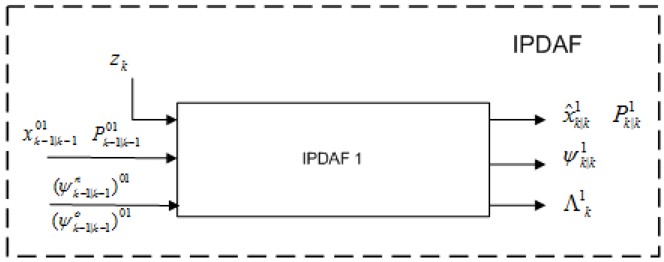

Figure 4.8: The IPDAF branch in the IMM-IPDAF algorithm

The IPDAF has the following steps:

Step 1: Obtain initial measurement prediction
The predicted measurement is set at the beginning of the algorithm or evaluated along the motion of the target as $\hat{z}_{k|k-1} = H\hat{x}_{k|k-1}$.

Step 2: Obtain new measurements for each time instant

The statistical or normalized distance d_ℓ^2 is obtained between locations of each measurement and the predicted measurement as follows:

$$d_\ell^2 = \left(z_k^\ell - \hat{z}_{k|k-1}\right)^T \underbrace{S_k^{-1}}_{\text{innovation covariance}} \underbrace{\left(z_k^\ell - \hat{z}_{k|k-1}\right)}_{\text{innovation}}. \tag{4.60}$$

Afterwards, the gate threshold [8] γ is found as

$$\gamma = 2\ln\left[\frac{P_d}{(1-P_d)(2\pi)^{M_d/2}\sqrt{(|S_k|)}}\right] \tag{4.61}$$

where P_d denotes the probability of detection, M_d is the dimension of the measurement, S_k is the innovation covariance. Then, each measurement is checked if they lie within the gate threshold γ as follows

$$d_\ell^2 \leq \gamma. \tag{4.62}$$

The observations below the gate threshold are named as the validated measurements and fed to the IPDAF algorithm as in Figure 4.2.

Therefore, the set of measurements which fall in the validation region at scan k is denoted as follows

$$z_k = \left\{z_k^\ell\right\}_{\ell=1}^{N_k}. \tag{4.63}$$

Remember that the performance of track formation is directly influenced by the validated measurements. More observations in the gate requires more tracks which would be treated as the initial candidate tracks and therefore increase the computation and the possibility of initializing a false track which leads to the miss of the target.

Step 3: Evaluate a priori track existence probabilities

Track existence is modelled by Markov chain two in Section 2.4.2. Before having the filtered state estimate, the probability of the true target existence probability for each assumption are calculated as described in the following paragraphs.

A priori track existence probability that the target exists and is observable $\psi^{o}_{k|k-1}$ before having received of the new measurement at scan k is calculated as

$$\psi^{o}_{k|k-1} = \pi_{11}\psi^{o}_{k-1|k-1} + \pi_{21}\psi^{n}_{k-1|k-1} + \pi_{31}(1 - \psi_{k-1|k-1}). \tag{4.64}$$

where $\pi_{11}, \pi_{21}, \pi_{31}$ are the coefficients of the Markov matrix which is defined as in Eq. 4.77 and points to all probabilities that a model can interact. $\psi^{o}_{k-1|k-1}$ is the mixing track existence probability based on the assumption that target exists and is observable, $\psi^{n}_{k-1|k-1}$ indicates the mixing track existence probability that target exists but is not observable calculated after the interaction at scan $k-1$. Those values are fed to IPDAF algorithm from previous time step.

A priori track existence probability that the target exists but is not observable $\psi^{n}_{k|k-1}$ before receipt of the measurements is obtained as follows

$$\psi^{n}_{k|k-1} = \pi_{12}\psi^{o}_{k-1|k-1} + \pi_{22}\psi^{n}_{k-1|k-1} + \pi_{32}(1 - \psi_{k-1|k-1}). \tag{4.65}$$

A priori track existence probability $\psi_{k|k-1}$ is then the sum of the Eq. 4.64 and Eq. 4.65 as

$$\psi_{k|k-1} = \psi^{o}_{k|k-1} + \psi^{n}_{k|k-1}. \tag{4.66}$$

A priori track existence probability $\psi_{k|k-1}$ is then used to calculate "a priori track existence probability that the target does not exist" $(1 - \psi_{k|k-1})$ as follows

$$(1 - \psi_{k|k-1}) = p_{13}\psi^{o}_{k-1|k-1} + p_{23}\psi^{n}_{k-1|k-1} + p_{33}(1 - \psi_{k-1|k-1}). \tag{4.67}$$

The initial track existence probability for each IPDA filter is obtained.

Step 4: Find probabilities of each hypothesis

The probabilities of each hypothesis are handled in accordance with the Markov chain two approach described in section 2.4.2 by two assumptions:

i) The probability or likelihood that none of the validated measurements is target originated

$$\beta^{o}_{k} = \frac{(1 - P_d P_g)\,\psi^{o}_{k|k-1} + \psi^{n}_{k|k-1}}{(1 - \delta_k)\,\psi^{o}_{k|k-1} + \psi^{n}_{k|k-1}} \tag{4.68}$$

where δ_k is calculated as in Eq. 2.23 and Λ^{ℓ}_{k} is the probability density function in which the measurement ℓ (true target) is in the validated region.

ii) The probability that each measurement in the gate is likely to have been originated from true target

$$\beta^{\ell}_{k} = \frac{P_d P_g\,\frac{V_k}{\hat{N}_k}\Lambda^{\ell}_{k}\psi^{o}_{k|k-1}}{(1 - \delta_k)\,\psi^{o}_{k|k-1} + \psi^{n}_{k|k-1}} \tag{4.69}$$

where δ_k is calculated as in Eq. 2.23, Λ_k^ℓ is the probability density function in which the measurement ℓ is in the validated region, P_d denotes the probability of detection, P_g is the probability that the true target is in the validation region.

Step 5: Find weighted innovation

The weighted innovation is obtained by the probabilistically weighted sum of the difference of each measurement from the predicted measurements $(z_k^\ell - \hat{z}_{k|k-1})$

$$\tilde{z}_k = \sum_{\ell=1}^{N_k} \beta_k^\ell (z_k^\ell - \hat{z}_{k|k-1}). \tag{4.70}$$

Step 6: Update weighted state estimate and associated covariance

The IPDAF obtains the state estimate $\hat{x}_{k|k}$ and the associated covariance $P_{k|k}$ as in PDAF algorithm 4.1.1.1 by incorporating the association probabilities evaluated at Step 4.

Step 7: Update track existence probability

Upon receipt of data at scan k, the track existence probability that the target exists and is observable $\psi_{k|k}^o$ is updated by

$$\psi_{k|k}^o = \frac{1 - \delta_k}{1 - \delta_k \, \psi_{k|k-1}^o} \, \psi_{k|k-1}^o. \tag{4.71}$$

The track existence probability that the target exists at scan k and is not observable $\psi_{k|k}^n$ is updated after the measurements are received

$$\psi_{k|k}^n = P\left(x_k^n \mid z^k\right) = \frac{\psi_{k|k-1}^n}{1 - \delta_k \, \psi_{k|k-1}^o}. \tag{4.72}$$

Here,

$$\delta_k = \begin{cases} P_d P_g & N_k = 0 \\ P_d P_g \left[1 - \frac{V_k}{\hat{N}_k} \sum_{\ell=1}^{N_k} \Lambda^\ell\right] & \text{otherwise} \end{cases} \tag{4.73}$$

where N_k is the number of observations on scan k, Λ_k^ℓ is the likelihood associated with observation i

$$\Lambda_k^\ell = \frac{\frac{1}{P_g}}{(2\pi)^{M_d/2} \sqrt{|S_k^i|}} e^{d_i^2/2} \tag{4.74}$$

where P_g is the probability that the true target falls within the validation region, V_k denotes volume of the gate on scan k, $\hat{N}_k$ is an adaptive estimate of the expected number of false observations in the gate (False measurements are assumed to have Poisson distribution)

$$\hat{N}_k = \begin{cases} 0 & N_k = 0 \\ N_k - P_d P_g \, \psi_{k|k-1} & \text{otherwise.} \end{cases} \tag{4.75}$$

The total filtered track existence probabilities at scan k are finally evaluated as the sum of both track existence probabilities, $\psi_{k|k}^o$, $\psi_{k|k}^n$

$$\psi_{k|k} = \psi_{k|k}^o + \psi_{k|k}^n. \tag{4.76}$$

The IPDAF algorithm generates the state estimate, the associated covariance and track existence probability for each model at the output.

The IPDAFs are now integrated in the IMM structure to support the maneuvering motion of the single target. The IMM structure combines IPDAF outputs assuming that the target move at one of the finite number of models with a given probability modelled by a Markov matrix.

Step 1 : Determine model probability transition matrix

A 3×3 Markov Matrix for a 3 model case is defined as follows

$$\pi_{ij} = \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix}. \tag{4.77}$$

The sum of the probabilities at each row must be unity in the Markov Matrix for any given model. Therefore, for model j

$$\sum_{j=1}^{M} \pi_{ij} = 1. \tag{4.78}$$

Each column indicates one model in the matrix. Therefore, the 3×3 Markov matrix can be interpreted as a Markov matrix of a three-model IMM. The elements in the diagonal give an idea about how the algorithm sticks to its model at each step. The rest of the elements in the matrix are the prior probabilities of model transition probability from one model to the other model.

Step 2: Obtain likelihoods and updated model probabilities

The Likelihood function Λ_k^j is obtained ($j = 1, \ldots, M$) using Eq. 4.13 for the measurements. Using the Bayes' rule [20], the updated model probabilities become

$$\mu_k^j = \frac{\Lambda_k^j \, \mu_{k|k-1}^j}{\sum_{i=1}^{M} \Lambda_k^i \, \mu_{k|k}^i} \tag{4.79}$$

where $\mu_{k|k-1}^j$ is the predicted mode probability for model j given as

$$\mu_{k|k-1}^j = \sum_{j=1}^{M} \pi_{ij} \, \mu_{k-1}^j. \tag{4.80}$$

The predicted mode probability indicates the probability after interaction that the target is in model j.

Step 3: Calculate overall state estimate and associated covariance matrix and track existence probability

The overall state estimate is a weighted sum of the state estimates of each filter

$$\hat{x}_{k|k} = \sum_{j=1}^{M} \mu_k^j \hat{x}_{k|k}^j \tag{4.81}$$

where $\hat{x}_{k|k}^j$ is the estimate for model j and μ_k^j is the updated model probability.

The overall covariance matrix is evaluated as

$$P_{k|k} = \sum_{j=1}^{M} \mu_k^j \left[P_{k|k}^j + DP_k^j \right]. \tag{4.82}$$

The term added to compensate the uncertainty in the gate, DP_k^j, is defined as

$$DP_k^j = \left[\hat{x}_{k|k} - \hat{x}_{k|k}^j\right]\left[\hat{x}_{k|k} - \hat{x}_{k|k}^j\right]^T . \tag{4.83}$$

To clarify, the overall state estimate is intended to compare the true target trajectory and the estimation and hence to calculate the position error. This step is not a part of the recursive algorithm.

The filtered track existence probability is described as

$$\psi_{k|k} = \sum_{j=1}^{M} \psi_{k|k}^j \mu_k^j \tag{4.84}$$

which is used in track initiation obtained by the track score function [8] which uses the thresholds in Eq. 4.57.

Step 4: Calculate mixing probability
The mixing probability that the target makes the transition from model i to j is given as:

$$\mu_k^{ij} = \frac{\pi_{ij}\,\mu_k^i}{\mu_{k|k-1}^j} \tag{4.85}$$

where μ_k^i is the probability for mode i,
$\mu_{k|k-1}^j$ is the predicted mode probability for model j,
π_{ij} is the model transition probability matrix that the target would make a transition from model state i to state j depending on the coefficients of the matrix.

Step 5: Find mixing state and covariance estimates and mixing track existence probabilities

The initial state estimation for model j $\hat{x}_{k|k}^{0j}$ is calculated in order to provide an input to the PDA Filter in Eq. 4.8.

The initial state estimation of the model j is obtained by using mixing probabilities μ_k^{ij} and state estimates $\hat{x}_{k|k}^i$ that are converted into a common dimensionality

$$\hat{x}_{k|k}^{0j} = \sum_{i=1}^{M} \mu_k^{ij} \hat{x}_{k|k}^i \tag{4.86}$$

where $\hat{x}_{k|k}^i$ is the estimate for model i.

The initial covariance $P_{k|k}^{0j}$ for model j is given by

$$P_{k|k}^{0j} = \sum_{i=1}^{M} \mu_k^{ij} \left[P_{k|k}^j + DP_k^{ij}\right]. \tag{4.87}$$

The spread of the mean DP_k^{ij} which is the difference in the state estimates from model i to j is

$$DP_k^{ij} = \left[\hat{x}_{k|k}^i - \hat{x}_{k|k}^{0j}\right]\left[\hat{x}_{k|k}^i - \hat{x}_{k|k}^{0j}\right]^T . \tag{4.88}$$

The mixing track existence probability that the target exists and is observable is calculated as the weighted sum of the track existence probabilities that the target exists and is observable for each model

$$(\psi_{k-1|k-1}^o)^{0j} = \sum_{i=1}^{M} \mu_{k-1}^{ij} (\psi_{k-1|k-1}^o)^i . \tag{4.89}$$

The mixing track existence probability that the target exists but is not observable is calculated in the same manner by incorporating the mixing probability μ^{ij}_{k-1} in Eq. 4.89 as follows

$$(\psi^n_{k-1|k-1})^{0j} = \sum_{i=1}^{M} \mu^{ij}_{k-1} (\psi^n_{k-1|k-1})^i. \tag{4.90}$$

The mixing track existence probabilities are then used in the calculation of a priori track existence probabilities of both assumption in the next time step.

Step 6: Find predicted state, associated predicted covariance and predicted track existence probabilities

The state estimate obtained after interaction in the IMM state, also named as mixing estimate, is used to calculate the predicted state that is used in the IPDAF as follows:

$$\hat{x}^j_{k+1|k} = F^j\, \hat{x}^{0j}_{k+1|k}. \tag{4.91}$$

The associated predicted covariance is found as

$$P^j_{k+1|k} = F^j\, P^{0j}_{k|k}\, F^j + Q^j_k \tag{4.92}$$

where F^j, G^j, w^j_{k-1}, Q^j_{k-1} indicate the state transition matrix, the noise matrix, the noise vector, the covariance respectively as defined in section 3.2.

The predicted state and the associated covariance carry previous states' tracking kinematics and provide the base for the current state estimation.

The measurement prediction, also known as the predicted measurement, is then evaluated as

$$\hat{z}^j_{k+1|k} = H^j\, \hat{x}^j_{k+1|k}. \tag{4.93}$$

The measurement prediction is then kept as the center of the validation gate as mentioned in Step 1 in IPDAF algorithm.

The predicted or a priori track existence probability that the target exists and is observable is then evaluated by

$$(\psi^o_{k|k-1})^j = \pi_{11}(\psi^o_{k-1|k-1})^{0j} + \pi_{21}(\psi^n_{k-1|k-1})^{0j} + \pi_{31}(1 - \psi_{k-1|k-1})^{0j}. \tag{4.94}$$

The predicted or a priori track existence probability that track exists but is not observable $(\psi^n_{k|k-1})^j$ is obtained as

$$(\psi^n_{k|k-1})^j = \pi_{12}(\psi^o_{k-1|k-1})^{0j} + \pi_{22}(\psi^n_{k-1|k-1})^{0j} + \pi_{32}(1 - \psi_{k-1|k-1})^{0j}. \tag{4.95}$$

The predicted track existence probabilities are used to update the track existence probabilities for the current step. The track existence probability is used to update the log likelihood ratios in Eq. 4.56. The IMM part of IMM-IPDAF is similarly performed as described earlier. One cycle of the IMM-IPDAF algorithm is provided in Table 4.3.

The development so far has established the basis of the VSIMM-IPDAF algorithm (discussed in the next section) by putting together the following components: IMM-PDAF algorithm, IMM-PDAF algorithm with stop model and hide model and the IPDAF algorithm with track initiation. These components together has formed the algorithm described in the present section, referred to as the IMM-IPDAF algorithm.

Table4.3: One cycle of the IMM-IPDAF algorithm in the AMM branches

1. Model-conditioned reinitialization	(for $j = 1, 2, \ldots, M$)
Predicted mode probability	$\mu^j_{k\|k-1} = \sum_{i=1}^{M} \pi_{ij}\mu^i_{k-1}$
Mixing weight	$\mu^{ij}_{k-1} = \frac{\pi_{ij}\mu^i_{k-1}}{\mu^j_{k\|k-1}}$
Mixing estimate	$\hat{x}^{oj}_{k-1\|k-1} = \sum_{i=1}^{M} \mu^{ij}_{k-1}\hat{x}^i_{k-1\|k-1}$
Mixing covariance	$P^{oj}_{k-1\|k-1} = \sum_{i=1}^{M} \mu^{ij}_{k-1}\left[P^j_{k-1\|k-1} + DP^j_k\right]$
Mixed track existence probability that	
Track exists and is observable	$\left(\psi^o_{k-1\|k-1}\right)^{0j} = \sum_{i=1}^{M} \mu^{ij}_{k-1}\left(\psi^o_{k-1\|k-1}\right)^i$
Track exists but is not observable	$\left(\psi^n_{k-1\|k-1}\right)^{0j} = \sum_{i=1}^{M} \mu^{ij}_{k-1}\left(\psi^n_{k-1\|k-1}\right)^i$
2. Model Conditioned Filtering	for ($j = 1, 2, \ldots, M$)
Predicted state	$\hat{x}^j_{k\|k-1} = F^j_{k-1}\,\hat{x}^{oj}_{k-1\|k-1} + G^j_{k-1}w^j_{k-1}$
Predicted covariance	$P^j_{k\|k-1} = F^j_{k-1}P^{oj}_{k-1\|k-1}(F^j_{k-1})' + G^j_{k-1}Q^j_{k-1}(G^j_{k-1})'$
A priori track existence probability that	
Track exists and is observable	$\left(\psi^o_{k\|k-1}\right)^j = \pi_{11}\left(\psi^o_{k-1\|k-1}\right)^{0j} + \pi_{21}\left(\psi^n_{k-1\|k-1}\right)^{0j}$ $+\pi_{31}\left(1-\psi_{k-1\|k-1}\right)^{0j}$
Track exists but is not observable	$\left(\psi^n_{k\|k-1}\right)^j = \pi_{12}\left(\psi^o_{k-1\|k-1}\right)^{0j} + \pi_{22}\left(\psi^n_{k-1\|k-1}\right)^{0j}$ $+\pi_{32}\left(1-\psi_{k-1\|k-1}\right)^{0j}$
Association probability $\beta^{0\|j}_k$	$\beta^{0\|j}_k = \frac{(1-P_dP_g)\left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j}{(1-\delta_k)\left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j}$
Association probability $\beta^{\ell\|j}_k$	$\beta^{\ell\|j}_k = \frac{P_dP_g\,\frac{V_k}{N_k}\Lambda^\ell_k\left(\psi^o_{k\|k}\right)^j}{(1-\delta_k)\left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j}$
Measurement innovation (residual)	$\tilde{z}^{\ell j}_k = z^\ell_k - H^j_k\hat{x}^j_{k\|k-1}$
Weighted measurement innovation	$\tilde{z}^j_k = \sum_{\ell=1}^{N_k} \beta^{\ell\|j}_k\left(z^\ell_k - \hat{z}_{k\|k-1}\right)$
Innovation covariance	$S^j_k = H^j_kP^j_{k\|k-1}(H^j_k)' + R^j_k$
Filter gain	$W^j_k = P^j_{k\|k-1}\,(H_k)'\,(S^j_k)^{-1}$
Updated estimate	$\hat{x}^j_{k\|k} = \hat{x}^j_{k\|k-1} + W^j_k\tilde{z}^j_k(W^j_k)'$
Updated covariance	$P^j_{k\|k} = \beta^{0\|j}_k\,P^j_{k\|k-1} + (1-\beta^{0\|j}_k)\,\grave{P}_{k\|k} + DP^j_k$
Updated track existence probability for $(\psi^o_{k\|k})^j$	$\left(\psi^o_{k\|k}\right)^j = \frac{1-\delta_k}{1-\delta_k\left(\psi^o_{k\|k-1}\right)^j}\left(\psi^o_{k\|k-1}\right)^j$
Updated track existence probability for $\left(\psi^n_{k\|k}\right)^j$	$\left(\psi^n_{k\|k}\right)^j = \frac{\left(\psi^n_{k\|k-1}\right)^j}{1-\delta_k\left(\psi^o_{k\|k-1}\right)^j}$
Updated track existence probability	$\psi^j_{k\|k} = \left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j$
3. Mode Probability Update	(for $j = 1, 2, \ldots, M$)
Model likelihood	$\Lambda^j_k = \mathcal{N}(\tilde{z}^j_k; 0, S^j_k)$
Mode probability	$\mu^j_k = \frac{\mu^j_{k\|k-1}\Lambda^j_k}{\sum_{i=1}^{M}\mu^i_{k\|k-1}\Lambda^i_k}$

4. Non-recursive Output of IMM

Overall state estimate	$\hat{x}_{k\|k} = \sum_{j=1}^{M} \hat{x}^j_{k\|k}\mu^j_k$
Overall covariance	$P_{k\|k} = \sum_{j=1}^{M} \mu^j_k \left[P^j_{k\|k} + \left(\hat{x}_{k\|k} - \hat{x}^j_{k\|k}\right)\left(\hat{x}_{k\|k} - \hat{x}^j_{k\|k}\right)'\right]$
The filtered track existence probability	$\psi_{k\|k} = \sum_{j=1}^{M} \psi^j_{k\|k}\mu^j_k$

4.1.6 VSIMM-IPDAF as a branch of the VS-A-IMM Tracker

Multiple model algorithms depend on a predetermined fixed number of models in general. If the number of models used in the IMM algorithm becomes a large set, it brings an increase in the computational burden, and the use of unnecessary models causes larger position errors in the tracking performance [22]. To overcome this problem, variable structure IMM techniques which provide the use of relevant model sets are used, and hence results in better performance during onroad movements if the models are chosen to reflect true target behaviour. Therefore, it aims to support the changing motion of the target or topographic changes by using variable model sets. The IMM-IPDAF algorithm described in the previous section has to be reformulated as a variable structure algorithm as part of a VS-A-IMM algorithm. This is achieved as follows in each branch:

In the off-road branch there are 3 motion models, namely, the high process noise constant velocity model, the low process noise constant velocity model and the coordinated turn model. When the velocity of the target falls below the minimum detectable velocity the stop model is activated and the low process noise constant velocity model is replaced by the stop model.

In the on road branches there are 3 motion models, namely the high process noise constant velocity model, the medium process noise constant velocity model and the low process noise constant velocity model. When the velocity of the target falls below the minimum detectable velocity the stop model is activated and the low process noise constant velocity model is replaced by the stop model. When the target is hidden behind a terrain obscuration the hide model is activated.

It is important to note that the transitions of the target of interest from onroad to offroad or vice versa are not handled in this algorithm. Each VSIMM-IPDAF branch developed here indicates the state estimate of only the offroad segment or one of the road segment's state estimate and provides one of the ingredients of the variable structure of the autonomous IMM.

4.2 Traditional VSIMM-PDAF

The traditional VSIMM algorithm [22, 23, 29] directly combines the overall state estimate of offroad segments and road segments utilizing the classical MMSE technique, without using a cascade composition of the IMM algorithms.

The VSIMM itself turns out to be an IMM algorithm of multiple models that are added or deleted according to the a-priori information of the road at each time step.

Offroad

When the target is offroad, there is a two model set: i) a constant velocity model with low process noise, ii) a constant velocity model with high process noise.

The PDAF filtered estimates are incorporated into the VSIMM algorithm to find the overall MMSE state estimate. A general structure of the traditional VSIMM-PDAF turns out to be the one in Figure 4.9.

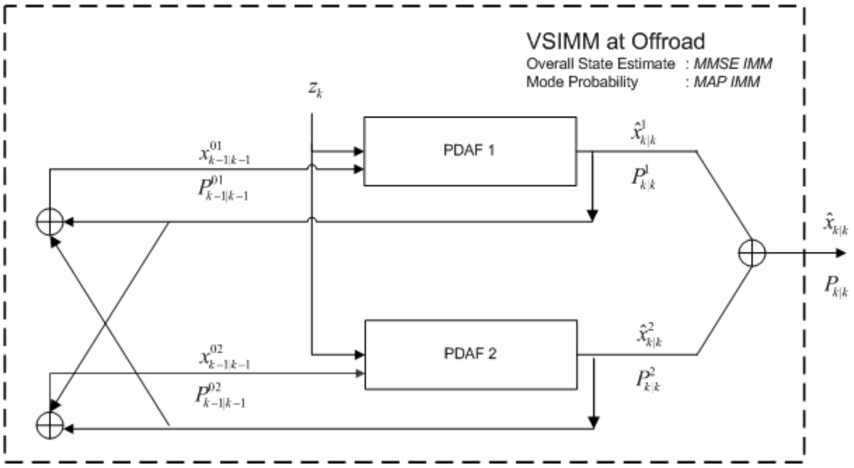

Figure 4.9: VSIMM-PDAF at the offroad region

As can be seen in Figure 4.9, VSIMM-PDAF algorithm adapts itself to the offroad region using the models defined for the offroad. The resulting MMSE estimate is the weighted sum of two filtered outputs.

$$\hat{x}^{\text{MMSE}}_{k|k} = \sum_{j=1}^{M=2} \hat{x}^j_{k|k}\mu^j_k \tag{4.96}$$

When the target moves close by a road segment, the model set of the closest road segment are added to the model set of the algorithm.

Onroad

When the target approaches to the road segment or evolves on the road or starts to move away from the road segment but is still under a calculated threshold, then the VSIMM-PDAF algorithm looks like the one in Figure 4.10.

As it can be seen from Figure 4.10, the number of models in the algorithm is increased by adding road segments' models in addition to offroad models to estimate the overall state. The initial state estimates and associated covariances are obtained by the interaction of four models and then are fed back to each filter for the current time step.

The filtered state estimated at the output of the algorithms is generated as follows:

$$\hat{x}^{\text{MMSE}}_{k|k} = \sum_{j=1}^{M_n} \hat{x}^j_{k|k}\mu^j_k \tag{4.97}$$

where M_n is the union of the model set and denoted as $M_n = M_{\text{offroad}} \cup M_{\text{road}}$.

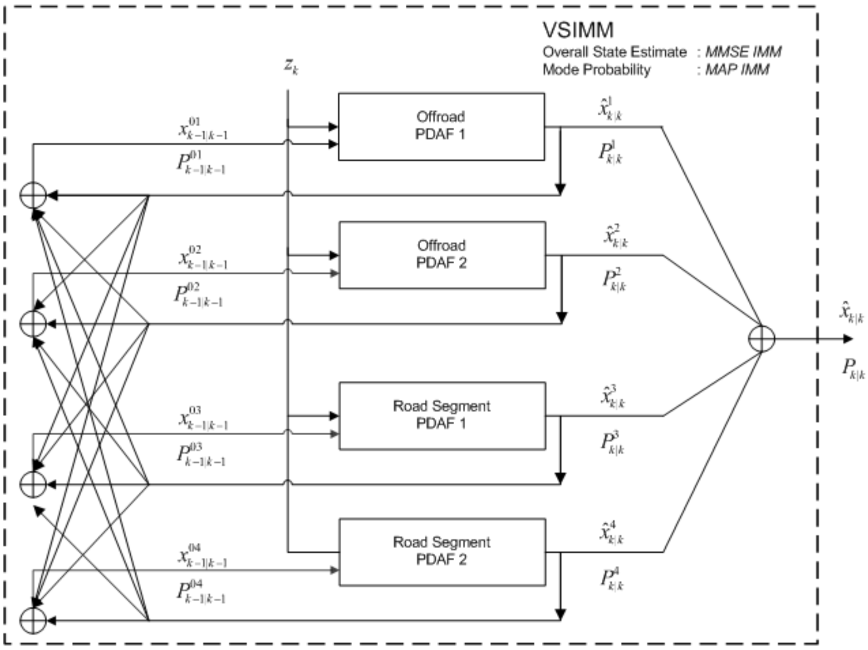

Figure 4.10: VSIMM-PDAF algorithm for Entry point, onroad and exit regions

If this structure is supposed to track sudden transitions from onroad to offroad as well one has to keep all 4 models. In junction areas the model set is enlarged to include models of all possible roads.

The general cycle of the VSIMM-PDAF algorithm is provided in Table 4.4.

Table4.4: One cycle of the VSIMM-PDAF algorithm

1. Model-Conditioned reinitialization	$(\forall m_i \in M_k)$
Predicted mode probability	$\mu^j_{k\|k-1} = \sum_{m_i \in M_{k-1}} \pi_{ij} \mu^j_{k-1}$
Mixing weight	$\mu^{ij}_{k-1} = \dfrac{\pi_{ij}\mu^i_{k-1}}{\mu^j_{k\|k-1}}$
Mixing estimate	$\hat{x}^{oj}_{k-1\|k-1} = \sum_{m_i \in M_{k-1}} \mu^{ij}_{k-1} \hat{x}^i_{k-1\|k-1}$
Mixing covariance	$P^{oj}_{k-1\|k-1} = \sum_{m_i \in M_{k-1}} \mu^{ij}_{k-1} \left[P^i_{k-1\|k-1} + DP^j_k\right]$
	$DP^j_k = \left(\hat{x}^{oj}_{k\|k} - \hat{x}^i_{k-1\|k-1}\right)\left(\hat{x}^{oj}_{k-1\|k-1} - \hat{x}^i_{k-1\|k-1}\right)'$
2. Model Conditioned Filtering	
Predicted state	$\hat{x}^j_{k\|k-1} = F^j_{k-1}\, \hat{x}^{oj}_{k-1\|k-1} + G_{k-1} w_{k-1}$
Predicted covariance	$P^j_{k\|k-1} = F^j_{k-1} P^{oj}_{k-1\|k-1} (F^j_{k-1})' + G_{k-1} Q^j_{k-1} (G_{k-1})'$
Measurement innovation (residual)	$\tilde{z}^{\ell j}_k = z^\ell_k - H^j_k \hat{x}^j_{k\|k-1}$
Weighted measurement innovation	$\tilde{z}^j_k = \sum_{\ell=1}^{N_k} \beta^\ell_k (z^\ell_k - \hat{z}_{k\|k-1})$

Innovation covariance	$S^j_k = H^j_k P^j_{k\|k-1} (H^j_k)' + R$
Filter gain	$W^j_k = P^j_{k\|k-1} (H_k)' (S^j_k)^{-1}$
Updated estimate	$\hat{x}^j_{k\|k} = \hat{x}^j_{k\|k-1} + W^j_k \tilde{z}^j_k (W^j_k)'$
Updated covariance	$P^j_{k\|k} = \beta^o_k \, P^j_{k\|k-1} \; + \; (1-\beta^o_k) \, \grave{P}_{k\|k} + DP^j_k$
3. Mode Probability Update	$(\forall m_j \in M_k)$
Model likelihood	$\Lambda^j_k = \mathcal{N}(\tilde{z}^j_k; 0, S^j_k)$
Mode probability	$\mu^j_k = \dfrac{\mu^j_{k\|k-1} \Lambda^j_k}{\sum_{m_i \in M_k} \mu^i_{k\|k-1} \Lambda^i_k}$
4. State Fusion	$(\forall m_j \in M_k)$
Overall state estimate	$\hat{x}_{k\|k} = \sum_{m_j \in M_k} \hat{x}^j_{k\|k} \mu^j_k$
Overall covariance	$P_{k\|k} = \sum_{m_j \in M_k} \mu^j_k \left[P^j_{k\|k} + \left(\hat{x}_{k\|k} - \hat{x}^j_{k\|k}\right)\left(\hat{x}_{k\|k} - \hat{x}^j_{k\|k}\right)' \right]$

4.3 Smart Gating

When the target is moving on the road, gating techniques can be applied intelligently which is referred to as smart gating techniques. This is motivated by the fact that when the target is on road, the measurement has to come from the measurements on the road if there is an elevation or buildings or if the target is too fast. The measurements can not be originated from offroad locations. This can be used in finding accurate validation regions (gates) when the target is on road. The smart gating approach provides fine tuning in the onroad branches of the VS-A-IMM algorithm without worrying that the track can go off road suddenly. The sudden exits to off road are handled by the off road branch of the VS-A-IMM algorithm.

In all VS-A-IMM branches the validated measurements are denoted as z_k which are the ones falling into the validated region (gate). In multi model tracking applications a number of gating techniques have been developed [61]. Centralized gating, model-based gating, model probability weighted gating, two stage model probability weighted gating methods are proposed. In this thesis, in all methods, the chi-square test 2.3.1 which define elliptic regions in the map is used for defining validation regions (gates) and the centralized gating for multimodels is used as the standard method of gating. Smart gating defined in Section 3.6 is used in the VS-A-IMM algorithm as well to improve the gating process particularly in road segments.

This brings us to the end of the description of the specific components of the algorithm. In the next sections the operation of the VS-A-IMM algorithm is described.

4.4 RESULTING TRACKER: VARIABLE STRUCTURE - AUTONOMOUS - INTERACTING MULTIPLE MODEL ALGORITHM

The Variable Structure - Autonomous - Interacting Multiple Model Algorithm (VS-A-IMM) has the following structure, a typical picture of which is given in Figure 4.11.

The first VSIMM-IPDAF branch is always the offroad IMM estimator. At every time instant, it estimates the state as if the target of interest is evolving at offroad region. It is a VSIMM-IPDAF branch that includes 2 constant velocity models with different process noise variances, one coordinated turn model with fixed turn rate, one stop model and one hide model.

The other branches of the structure are variable, hence resulting in a variable structure AMM algorithm, which represent different road segments, added or dropped according to the tests performed in the algorithm. For example in a junction with 4 roads, there are 5 branches present in the VS-A-IMM algorithm, namely offroad, current onroad, potential roads 1 to 3.

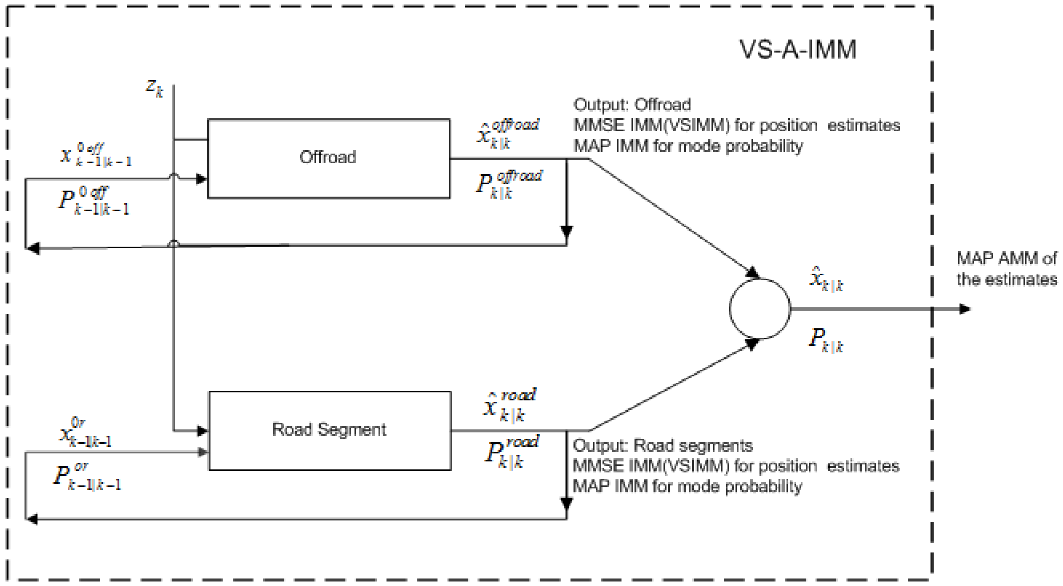

Figure 4.11: VS-A-IMM

The VSIMM branches representing road segments consist of 2 constant velocity models with different process noise variances, one stop model and one hide model.

Table4.5: One cycle of the VSIMM-IPDAF algorithm in the AMM branches of the VS-A-IMM algorithm

1. Model-conditioned reinitialization	$(\forall m_i \in M_k)$
Predicted mode probability	$\mu^j_{k\|k-1} = \sum_{m_i \in M_{k-1}} \pi_{ij} \mu^i_{k-1}$
Mixing weight	$\mu^{ij}_{k-1} = \frac{\pi_{ij}\mu^i_{k-1}}{\mu^j_{k\|k-1}}$
Mixing estimate	$\hat{x}^{0j}_{k-1\|k-1} = \sum_{m_i \in M_{k-1}} \mu^{ij}_{k-1} \hat{x}^i_{k-1\|k-1}$
Mixing covariance	$P^{0j}_{k-1\|k-1} = \sum_{m_i \in M_{k-1}} \mu^{ij}_{k-1} \left[P^j_{k-1\|k-1} + DP^j_k\right]$
Mixed track existence probability that Track exists and is observable	$\left(\psi^o_{k-1\|k-1}\right)^{0j} = \sum_{m_i \in M_{k-1}} \mu^{ij}_{k-1} \left(\psi^o_{k-1\|k-1}\right)^i$

Track exists but is not observable	$\left(\psi^n_{k-1\|k-1}\right)^{0j} = \sum_{m_i \in M_{k-1}} \mu^{ij}_{k-1} \left(\psi^n_{k-1\|k-1}\right)^i$
2. Model Conditioned Filtering	
Predicted state	$\hat{x}^j_{k\|k-1} = F^j_{k-1}\, \hat{x}^{0j}_{k-1\|k-1} + G^j_{k-1} w^j_{k-1}$
Predicted covariance	$P^j_{k\|k-1} = F^j_{k-1} P^{0j}_{k-1\|k-1} (F^j_{k-1})' + G^j_{k-1} Q^j_{k-1} (G^j_{k-1})'$
A priori track existence probability that	
Track exists and is observable	$\left(\psi^o_{k\|k-1}\right)^j = \pi_{11} \left(\psi^o_{k-1\|k-1}\right)^{0j} + \pi_{21} \left(\psi^n_{k-1\|k-1}\right)^{0j}$ $+\pi_{31} \left(1 - \psi_{k-1\|k-1}\right)^{0j}$
Track exists but is not observable	$\left(\psi^n_{k\|k-1}\right)^j = \pi_{12} \left(\psi^o_{k-1\|k-1}\right)^{0j} + \pi_{22} \left(\psi^n_{k-1\|k-1}\right)^{0j}$ $+\pi_{32} \left(1 - \psi_{k-1\|k-1}\right)^{0j}$
Association probability $\beta^{0\|j}_k$	$\beta^{0\|j}_k = \frac{(1-P_d P_g)\, \left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j}{(1-\delta_k)\, \left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j}$
Association probability $\beta^{\ell\|j}_k$	$\beta^{\ell\|j}_k = \frac{P_d P_g\, \frac{V_k}{\hat{N}_k} \Lambda^\ell_k \left(\psi^o_{k\|k}\right)^j}{(1-\delta_k)\, \left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j}$
Measurement innovation (residual)	$\tilde{z}^{\ell j}_k = z^\ell_k - H^j_k \hat{x}^j_{k\|k-1}$
Weighted measurement innovation	$\tilde{z}^j_k = \sum_{\ell=1}^{N_k} \beta^{\ell\|j}_k \left(z^\ell_k - \hat{z}_{k\|k-1}\right)$
Innovation covariance	$S^j_k = H^j_k P^j_{k\|k-1} (H^j_k)' + R^j_k$
Filter gain	$W^j_k = P^j_{k\|k-1}\, (H_k)'\, (S^j_k)^{-1}$
Updated estimate	$\hat{x}^j_{k\|k} = \hat{x}^j_{k\|k-1} + W^j_k \tilde{z}^j_k (W^j_k)'$
Updated covariance	$P^j_{k\|k} = \beta^{0\|j}_k\, P^j_{k\|k-1} + (1 - \beta^{0\|j}_k)\, \grave{P}_{k\|k} + DP^j_k$
Updated track existence probability for $(\psi^o_{k\|k})^j$	$\left(\psi^o_{k\|k}\right)^j = \frac{1-\delta_k}{1-\delta_k \left(\psi^o_{k\|k-1}\right)^j} \left(\psi^o_{k\|k-1}\right)^j$
Updated track existence probability for $\left(\psi^n_{k\|k}\right)^j$	$\left(\psi^n_{k\|k}\right)^j = \frac{\left(\psi^n_{k\|k-1}\right)^j}{1-\delta_k \left(\psi^o_{k\|k-1}\right)^j}$
Updated track existence probability	$\psi^j_{k\|k} = \left(\psi^o_{k\|k}\right)^j + \left(\psi^n_{k\|k}\right)^j$
3. Mode Probability Update	$(\forall m_j \in M_k)$
Model likelihood	$\Lambda^j_k = \mathcal{N}(\tilde{z}^j_k; 0, S^j_k)$
Mode probability	$\mu^j_k = \frac{\mu^j_{k\|k-1} \Lambda^j_k}{\sum\limits_{m_i \in M_k} \mu^i_{k\|k-1} \Lambda^i_k}$
4. Non-recursive Output of VSIMM-IPDAF	$(\forall m_j \in M_k)$
Overall state estimate	$\hat{x}_{k\|k} = \sum\limits_{m_j \in M_k} \hat{x}^j_{k\|k} \mu^j_k$
Overall covariance	$P_{k\|k} = \sum\limits_{m_j \in M_k} \mu^j_k \left[P^j_{k\|k} + \left(\hat{x}_{k\|k} - \hat{x}^j_{k\|k}\right)\left(\hat{x}_{k\|k} - \hat{x}^j_{k\|k}\right)'\right]$
The filtered track existence probability	$\psi_{k\|k} = \sum\limits_{m_j \in M_k} \psi^j_{k\|k} \mu^j_k$
In summary	
MMSE VSIMM-IPDAF	$\hat{x}^{\text{MMSE}}_{k\|k} = \sum\limits_{m_i \in M_k} \hat{x}^j_{k\|k} \mu^j_k$
MAP VSIMM-IPDAF	$\hat{m}^{\text{MAP}}_{k\|k} = \underset{m^j}{\text{argmax}}\, \{\mu^j_k, j \in M_k\}$

The VSIMM-IPDAF branches calculate their output using MAP estimation for the model probability and using MMSE estimation for the state, the details of which are given in Table 4.5. These quantities in each branch are then fed to the AMM estimator to compute the state combination.

The AMM estimator processes the incoming information from the branches and outputs the MAP estimates of the model and the state. Hence, the final state becomes the state estimate coming from branch which has the largest mode probability.

In this thesis, the target moves in rural areas freely unless there is a topographic constraint such as an elevation, a vegetated area and so on. Therefore, while the target moves in the offroad area, the distances to nearby roads are computed by projection techniques. If the target motion is evolving on the road, it is possible that the target can move away from the road segment instantaneously. The structural parts of the AMM estimator depend on the topography and the a-priori road information as follows:

- The offroad condition
- Entry/exit points: offroad/onroad transition
- Onroad condition
- Junction condition
- Obscuration or terrain constraint condition

4.4.1 Offroad

When the VS-A-IMM algorithm runs on off-road where there are no roads nearby which could activate other branches in the VS-A-IMM algorithm, there is only the off-road branch present in the algorithm as seen in Figure 4.12.

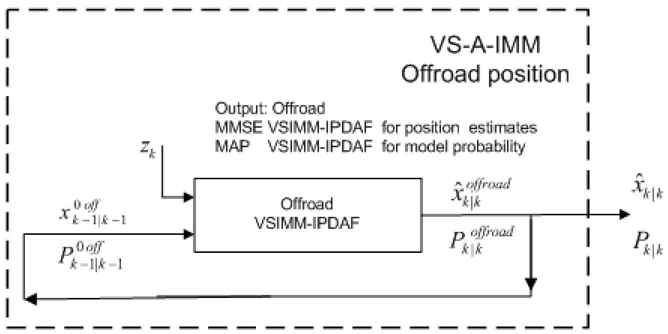

Figure 4.12: VS-A-IMM at Offroad

The overall state estimate turns out to be the filtered state estimate provided by offroad branch VSIMM-IPDAF as :

$$\hat{x}_{k|k} = \hat{x}^{\text{offroad}}_{k|k} \tag{4.98}$$

The associated covariance equals to the covariance of the offroad branch as below:

$$P_{k|k} = P^{\text{offroad}}_{k|k} \tag{4.99}$$

The model probability is the one with the highest mode probability among the offroad model set. The MAP estimate of the mode probability is theoretically expressed as:

$$\hat{m}_{k|k}^{\text{MAP}} = \underset{m^j}{\text{argmax}}\ \{\mu_k^j, j \in M_{\text{offroad}}\} \tag{4.100}$$

where M_{offroad} indicates the model set of the offroad, μ_k^j is the model probability for model j.

4.4.2 Onroad

The target can slow down at the entry point or during onroad motion can move away from the road, which leads to the case that the offroad and the close road segment into account are processed and incorporated into the AMM tracker. Therefore, the conditions i) when the target comes nearby a road segment while moving on off-road or ii) the target moves on a road segment or iii) target is about to exit a road segment, the structure of the VS-A-IMM algorithm looks like as in Figure 4.13. The VS-A-IMM comprises of two VSIMM-IPDAF branches, one for the offroad condition and one for the road segment of interest. The state estimates and the model probability for offroad branch's VSIMM-IPDAF are found by

$$\hat{x}_{k|k}^{\text{MMSE for offroad}} = \sum_{j=1}^{M_{\text{offroad}}} \hat{x}_{k|k}^j \mu_k^j \tag{4.101}$$

$$\hat{m}_{k|k}^{\text{MAP for offroad}} = \underset{m^j}{\text{argmax}}\ \{\mu_k^j, j \in M_{\text{offroad}}\}. \tag{4.102}$$

where $\hat{m}_{k|k}^{\text{MAP for offroad}}$ indicates the model with highest probability among offroad model set.

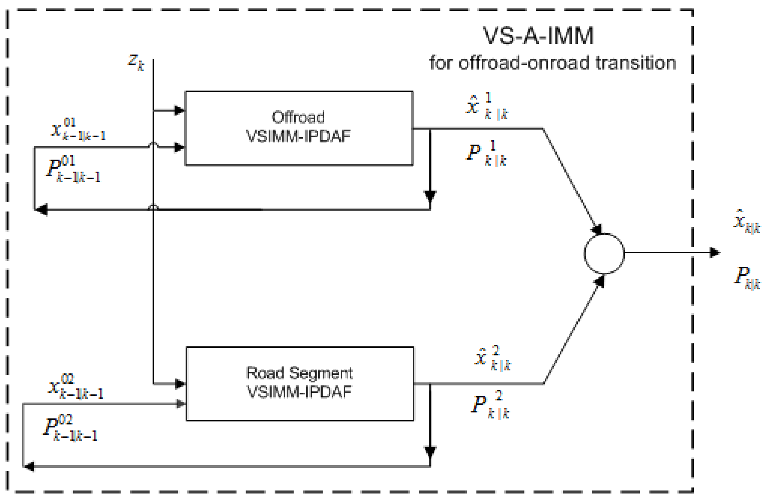

Figure 4.13: VS-A-IMM when target is at entry condition or at exit condition or on the road

The state estimate and the model probability for road segment VSIMM-IPDAF are evaluated as

$$\hat{x}_{k|k}^{\text{MMSE for road 1}} = \sum_{j=1}^{M_{\text{road 1}}} \hat{x}_{k|k}^j \mu_k^j. \tag{4.103}$$

The model which has a highest probability is found by

$$\hat{m}_{k|k}^{\text{MAP for road 1}} = \underset{m^j}{\text{argmax}} \{\mu_k^j, j \in M_{\text{road 1}}\}. \tag{4.104}$$

Both branches are incorporated into the tracker. The VS-A-IMM algorithm first compares each branch's model probability $\hat{m}_{k|k}^{\text{MAP for offroad}}$ and $\hat{m}_{k|k}^{\text{MAP for road 1}}$ and the MAP estimate of the model probabilities of branches at time k are obtained as

$$\hat{m}_{k|k}^{\text{MAP}} = \underset{m}{\text{argmax}} \{\mu_k^{\text{offroad}}, \mu_k^{\text{road}}\}. \tag{4.105}$$

The overall state estimate $\hat{x}_{k|k}$ is the state estimate of the branch which has the highest model probability as below

$$\hat{x}_{k|k} = \hat{x}_{k|k}^{\text{MAP}} \left(\hat{m}_{k|k}^{\text{MAP}}\right). \tag{4.106}$$

In summary, the MAP estimate of the overall state estimate is the maximum of the posterior density coming from the branch $\hat{m}_{k|k}^{\text{MAP}}$.

4.4.3 Terrain constraint

When there is a terrain constraint like elevation or bridge while evolving on the road, the structure of the VS-A-IMM algorithm turns out to be as in Figure 4.14. If there is a-priori information regarding

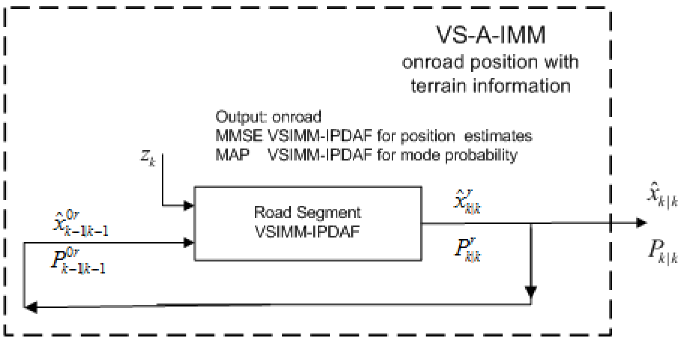

Figure 4.14: VS-A-IMM for onroad position with terrain constraint information

elevation around the road, it is assumed that the target cannot change its direction and keeps moving on the road. This assumption leads the elimiation of the offroad segment automatically and the overall state estimate becomes state estimate of the onroad segment in which target is being tracked. The model probability of the branch is not used to make a decision at the output of the tracker.

4.4.4 Junction

When the target approaches a junction, this means that the target is in the vicinity of all road segments connected to it. When the target reaches a junction all branches that represent the road segments are included in the VS-A-IMM algorithm. The algorithm has the structure in Figure 4.15.

The overall state estimation at the junction is obtained as follows. Road segments are firstly handled and the MMSE state estimate and the MAP estimate of model probabilities are found for each VSIMM-IPDAF road segment. The road segment with the highest model probability is chosen among the road segments. Offroad state estimation and the model probabilities are then obtained. The comparison of the offroad segment and the resulting road segment model probabilities are compared and the state estimate of the branch with the highest probability results in the overall state estimate.

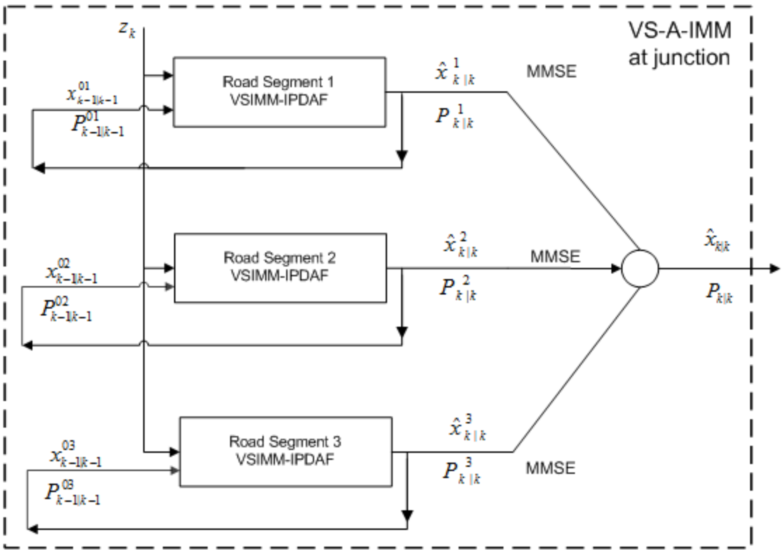

Figure 4.15: MAP Estimate of all road segments at junction point

4.4.5 Track Initiation

When the location of the track is not given it has to be initialized. The initialization is performed using the JPDAF algorithm described in Section 4.1.4 in the following manner:

Initially all the measurements in the first scan are treated as initial track states. Then, the initial values for process noise Q are assigned to those tracks and the initial measurement noises with zero mean and covariance R the initial track state covariance matrices $P_{k|k-1}$ are assigned as well.

In the second scan, another group of observations are obtained and associated with the observations obtained at first scan by the auction algorithm [8]. This is achieved by taking the difference among the associated observations and the tracks assumed in the first scan and hence the velocity of each track is found.

The track existence probability is assigned for each track and as the scan progresses the tracks with low track existence parameter are dropped. Assuming that the target provides measurements at all scans, the target is initialized in four scans.

4.5 MODEL SET ADAPTATION

When the target is off-road, distant away from road segments the only branch available in the VS-A-IMM algorithm is the off-road branch which includes two constant velocity models, one turn model and one stop model. As the target moves on its trajectory the variability in the structure of the algorithm is achieved through various controls. The controls that are used to determine whether a model has to be activated in a specific branch is referred to as in-branch controls. The controls that are used to add or delete road branches in the VS-A-IMM structure is referred to as intra-branch controls which are specifically defined as road segment controls and junction controls.

4.5.1 In-branch Controls

The in-branch controls performed in each branch are used to detect whether a stop model or a hide model has to be used. The velocity estimate of the target is checked up on and the stop model is activated or not as described in Section 4.1.2. The position of the target is also checked up on to detect whether the target is near a topographic obstruction as described in Section 4.1.3. If so, the hide-model is activated.

4.5.2 Road Segment Control

Road control is performed if the target moves on a road segment or leave the road segment at each time step. In accordance with the road condition, the tracker takes into account the model set which is active.

Assuming that the target is off-road, see Figure 4.16;

1) The state of the target is estimated by using the off-road branch VS-IMM algorithm. The MMSE output for the state position of the off-road branch is generated.

2) The state estimate is projected onto all segments on the road map. The Mahalanobis distance is calculated between predicted state and the state projected on the segments.

3) When the distances are less than a specific threshold, it is assumed that the target is related to the road segments used to find those distances. Then, the model set of that road segment is added to the VS-A-IMM algorithm. The target state is estimated by using these models in addition to the off-road branch VS-IMM.

4) Road test is repeated at each time step. If a branch for a specific road segment cannot pass the road test, that branch is removed from the VS-A-IMM algorithm.

If the target is detected in the vicinity of a junction, the road control test is stopped.

4.5.3 Junction Control

The junction control is performed as follows (See Figure 4.17).

1) When the target is evolving on the road, the Mahalonobis distance between the state estimate and

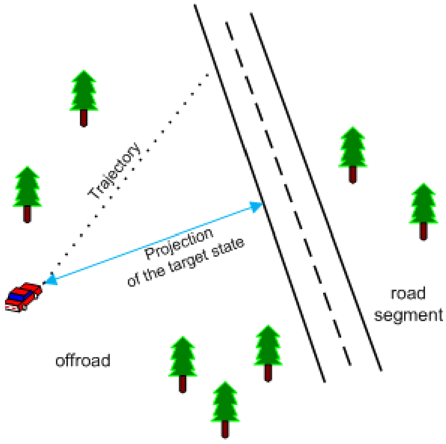

Figure 4.16: Projection of the target state to the road segment

the junction point is calculated.

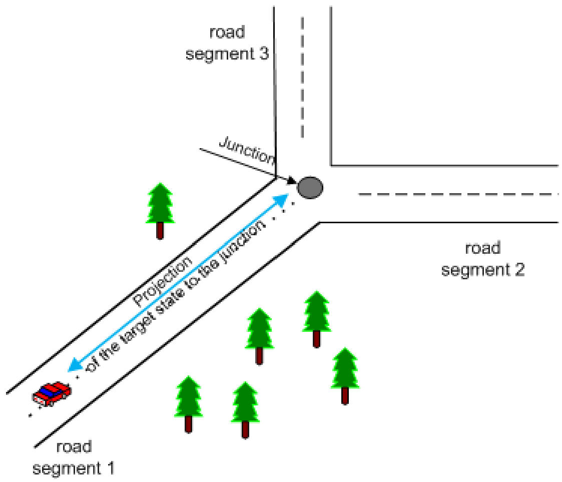

Figure 4.17: Projection of the target state to the junction

2) When the distance calculated is less than a specific junction threshold, it is assumed that the target is in the vicinity of the junction.

3) All branches intersecting in the junction are added in the VS-A-IMM algorithm. The filtered state estimates are projected onto all the segments. The projected state estimates are used as predicted states that belong to those models.

4) Using the projected state estimates, the state estimates and predictions are calculated for each branch.

5) Filtered state estimates are used in the junction control, the target is checked if it is still in the junction area.

The junction control is repeated until the target moves away from the junction area. When the target is in the junction area and its direction is toward a specific road segment, it is assumed that the likelihood of that road segment is higher than that of the others. When the target leaves the junction area, the road control is started again.

CHAPTER 5

SIMULATION RESULTS

The comparison of the simulation performances of the algorithms are handled through computing the Root Mean Square (RMS) error in position and in velocity of the targets, which is a universally accepted measure of evaluation for measuring error performance.

The mathematical formula of the RMS error is given by

$$\text{RMS error} = \sqrt{E\{(x-\hat{x})^2\}} \tag{5.1}$$

where x is the true value and $\hat{x}$ is an estimator of x. In ground target tracking, to compute the position errors, $\hat{x}$ is taken as the position state estimate of the target, to compute velocity errors $\hat{x}$ is taken as the velocity state estimate of the target, which are all inferred from the state vector.

A two dimensional sensor is modelled and located at origin. Each value in the graphs below represents the average of independent 100 Monte Carlo runs. Track initiation is performed at each scan for all algorithms. The common parameters in simulations for all algorithms are listed in Table 5.1. Target motions are studied in two different clutter density scenarios referred to as the "light clutter density"and the "heavy clutter density". The clutter density measurements are obtained from a uniform distribution at each time step.

Table5.1: Common Parameters in All Algorithms

Measurement Error	
Range Error	20 m
Azimuth Error	0.01 rad
T (Sampling interval)	1 sec
P_d (Except stop and hide modes)	0.99
P_g	0.99
Average Clutter Density	
Light Clutter Density	5.5 e^{-5} m^{-2}
Heavy Clutter Density	5.5 e^{-4} m^{-2}

From this point on, the VS-A-IMM IPDAF algorithm is referred to as "Tracker"and compared to the IMM-PDAF algorithm and VSIMM-PDAF algorithm which is resulted in MMSE output in terms of the RMS error performances in position and velocity.

Model Set of the IMM-PDAF algorithm
IMM-PDAF algorithm is implemented using two constant velocity motion models one of which has low process noise with standard deviation 5 m/s^2 and the other has high process noise with standard deviation 30 m/s^2.

Model Set of the VSIMM-PDAF VSIMM-PDAF model set consists of offroad and onroad motion models in which each in-branch has two constant velocity motion models.

- Offroad branch has two models with low process noise with standard deviation 5 m/s^2 and high process noise with standard deviation 30 m/s^2.
- Onroad branch has two models with 2.5 m/s^2 and 20 m/s^2.

Model Set of the Tracker

The target motion is assumed to be covered with the following models in the Tracker:

Offroad motion models

- Constant velocity with low process noise
- Constant velocity with high process noise
- Coordinated turn with low process noise
- Stop model

On-road motion models

- Constant velocity with low process noise
- Constant velocity with medium process noise
- Constant velocity with high process noise
- Stop model
- Hide model

Stop Mode
If the target velocity is less than the MDV or target stops, there is no measurement available and the probability of detection becomes zero. The stop model is activated in the Tracker algorithm. The process noise standard deviation in the stop model is taken as 2.5 m/s^2.

Hidden Mode
If target is not visible due to tunnels or terrain obstructions, there is no measurement available and the probability of detection becomes zero. The process noise for the hide model is chosen as 2.5 m/s^2.

The Markov matrix for the IMM-PDAF algorithm is defined as

$$[\pi_{ij}]_2 = \begin{bmatrix} 0.95 & 0.05 \\ 0.2 & 0.8 \end{bmatrix}.$$

The Markov matrix of the VSIMM-PDAF algorithm is chosen as

$$[\pi_{ij}]_3 = \begin{bmatrix} 0.8 & 0.10 & 0.10 \\ 0.1 & 0.70 & 0.20 \\ 0.15 & 0.15 & 0.70 \end{bmatrix}.$$

Both the stop model and the hide model have the same model probability matrices for slow and fast stages. These are given by

$$\left[p_{ij}\right]^{E} = \begin{bmatrix} 0.95 & 0.02 & 0.03 \\ 0.8 & 0.1 & 0.1 \\ 0.7 & 0.1 & 0.2 \end{bmatrix} \quad \left[p_{ij}\right]^{\bar{E}} = \begin{bmatrix} 0 & 0.95 & 0.05 \\ 0 & 0.8 & 0.2 \\ 0 & 0.7 & 0.3 \end{bmatrix}.$$

The parameters of the Tracker during simulations except the stop mode and the hidden mode are given in Table 5.2.

Table5.2: Parameters of the Tracker

Offroad Process Noise Std. Dev.		
Model 1 CV	5 m/s^2	
Model 2 CV	10 m/s^2	
Model 3 CT	5 m/s^2	
Onroad Process Noise Std. Dev.	**Along**	**Orthogonal**
Model 1 CV	30 m/s^2	3 m/s^2
Model 2 CV	15 m/s^2	1.5 m/s^2
Model 3 CV	10 m/s^2	1 m/s^2

5.1 TRAJECTORY 1

There are two special cases in Trajectory 1 which is shown in Figure 5.1: i) open field to onroad transition at point B, ii) junction J. We consider a single ground target evolving on open field between points AB and joins the road segment at point B, as shown in Figure 5.1. BJ indicates the road segment 1 that the target of interest follows. J shows the junction region and it is separated into two distinct road segments JC (road segment 2) and JD (road segment 3). Three different motion patterns are considered in Trajectory 1 that are denoted as Motion 1, Motion 2 and Motion 3.

5.1.1 Motion 1

The target motion is given in Figure 5.2 for the trajectory in Figure 5.1 as follows:

- The target is initiated with the velocity 16 m/s at point A which is an open field location at (750, 750).
- It continues with its velocity during next 50 secs.

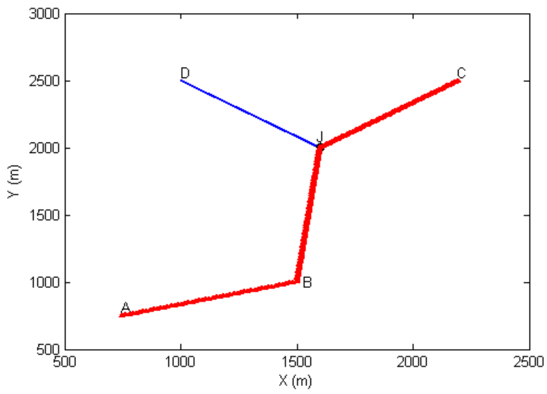

Figure 5.1: Trajectory 1: True Target Scenario

- After entering the road segment 1, it decelerates with $4m/s^2$ and its velocity lasts 80 secs till junction.

- It again decelerates with 4 m/s^2 and terminates its motion at road segment 2. The number of scans over all tracking time interval is 200.

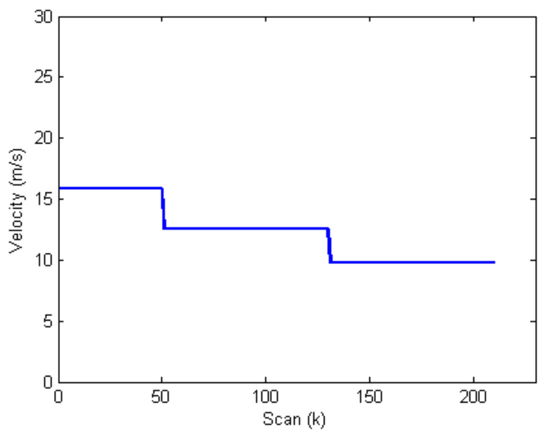

Figure 5.2: Trajectory 1: Motion 1 - True velocity of the target

5.1.1.1 Motion 1 - Light Clutter Density

The RMS position error performances of the algorithms, the RMS velocity error performances of the algorithms and the overall RMS errors of the algorithms obtained by executing 100 independent Monte Carlo runs are provided in Figure 5.3, Figure 5.4 and Table 5.3 respectively.

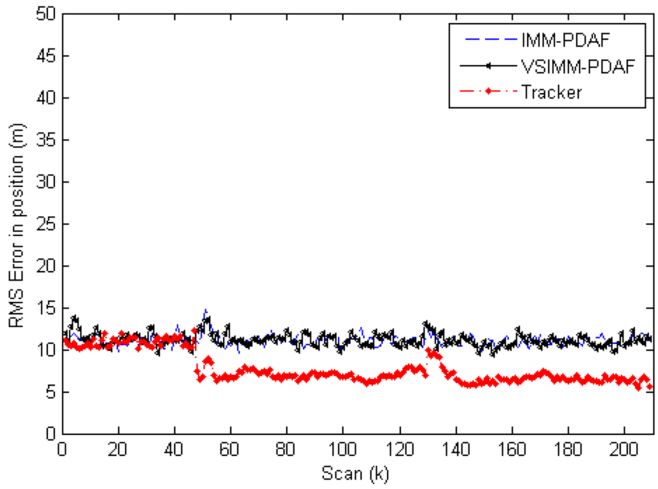

Figure 5.3: Trajectory 1: Motion 1 - RMS error in position for LC

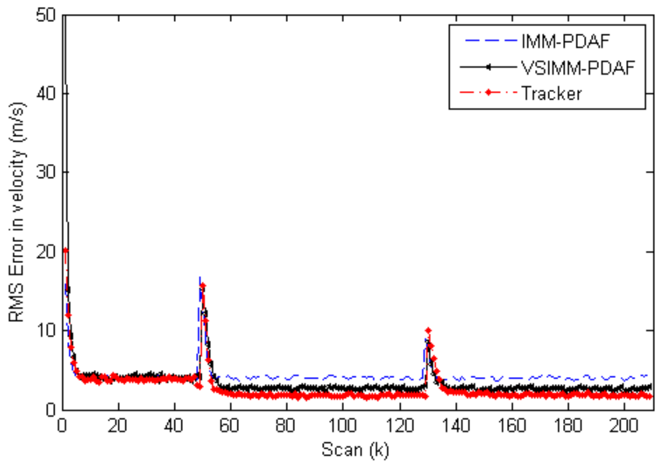

Figure 5.4: Trajectory 1: Motion 1 - RMS error in velocity for LC

Table5.3: RMS errors in position and velocity for Motion 1 - LC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	11.0720	4.2935
VSIMM-PDAF	11.1450	3.6380
Tracker	7.7296	2.7455

5.1.1.2 Motion 1 - Heavy Clutter Density

The RMS errors in position and velocity of all algorithms are demonstrated in Figure 5.5 and Figure 5.6 respectively.

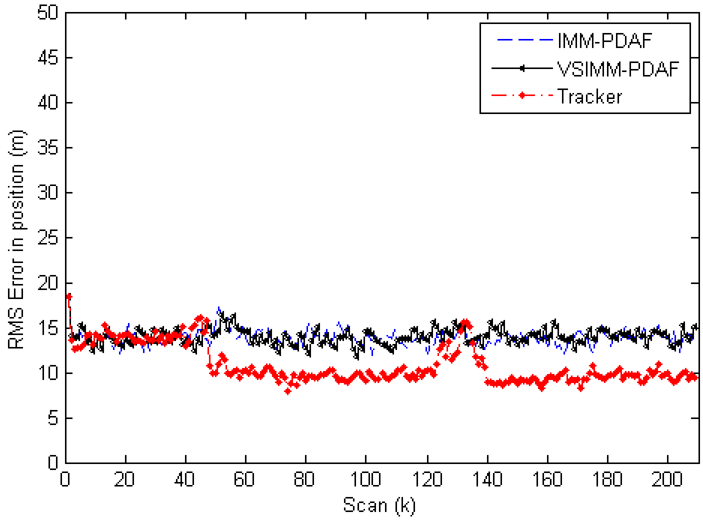

Figure 5.5: Trajectory 1: Motion 1 - RMS error in position for HC

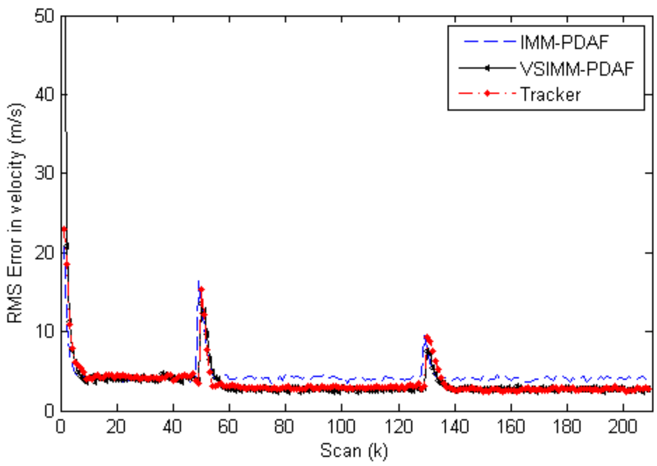

Figure 5.6: Trajectory 1: Motion 1 - RMS error in velocity for HC

The overall RMS errors of the algorithms in heavy clutter density by 100 Monte Carlo runs are provided in Table 5.4.

Table5.4: RMS errors in position and velocity for Motion 1 - HC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	13.8705	4.5705
VSIMM-PDAF	13.9873	3.8157
Tracker	10.8600	3.6435

Observations for Motion 1

Regarding the offroad motions, performances are close to each other whereas the Tracker's performance resulted in better performance during onroad regions. This is due to the fact that gating approach is applied and a priori information is provided for onroad. In the heavy clutter scenario both RMS errors in position and velocity increase, however the Tracker's performance is least effected among all.

The VSIMM-PDAF uses 4 models during the onroad motions to capture the possibility of moving to offroad abruptly. For this reason the performance is worse than the Tracker and close to IMM-PDAF that uses 2 models. Under normal circumstances the performance of a 4 model IMM would be worse than the 2-model IMM-PDAF, however since the 2 models in VSIMM-PDAF are the modified versions of the other 2 models (Process noise is elliptic rather than circular), the performance is not worsened much and is in the acceptable limits. The advantage we have is that the VSIMM-PDAF is responsive to offroad transitions.

All algorithms make a peak at transition points from offroad to onroad and junction areas. It is interesting to note that the velocity RMS error peaks in the road entry and in the junction have similar magnitude.

At the junction area, the Tracker makes a similar peak, however the width of the peak is wider than the others. As to the RMS error in velocity, all algorithms show similar performance for offroad motion and the Tracker has smaller RMS error in velocity during onroad motion. The two peaks in the Figure 5.4 indicate the offroad to onroad and junction point transitions.

It is interesting to note that although the position error performance of the VSIMM-PDAF is similar to the IMM-PDAF, its velocity error performance is close to the Tracker. Using models that match to road map help estimating the velocity better in VSIMM-PDAF whereas this does not help a lot in estimating the position.

5.1.2 Motion 2

The scenario has a stop mode in the offroad region on Trajectory 1 as seen in Figure 5.7. The motion of the target is built in Figure 5.8.

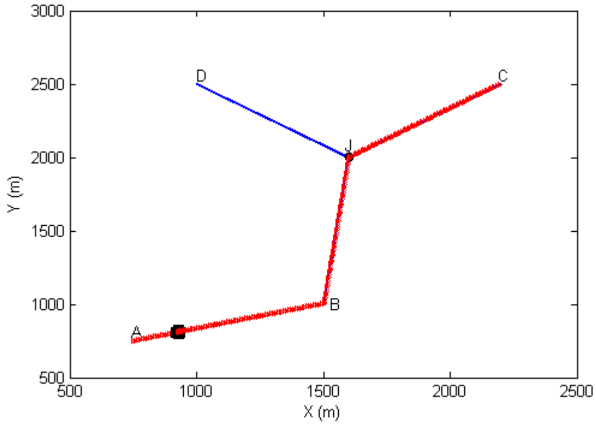

Figure 5.7: Trajectory 1: Motion 2 - True target scenario

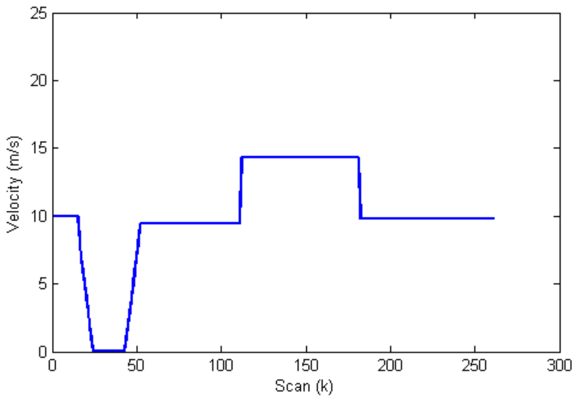

Figure 5.8: Trajectory 1: Motion 2 - True velocity of the target

5.1.2.1 Light Clutter Density

Figure 5.9 presents the RMS position error performances of the algorithms. The overall RMS errors

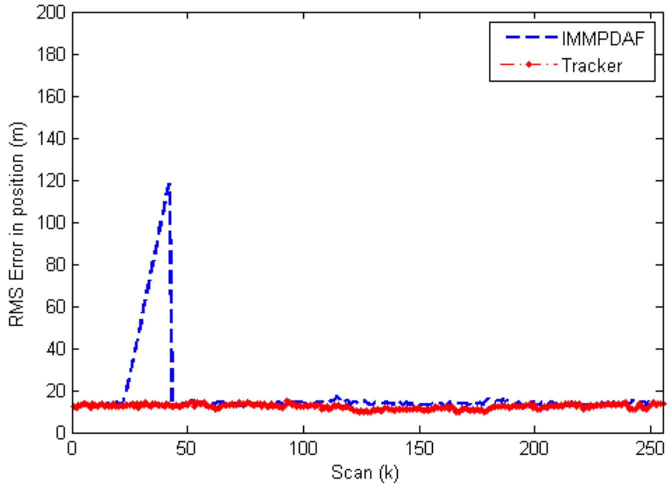

Figure 5.9: Trajectory 1: Motion 2 - RMS error in position for LC

averaged over the stop mode tracking periods in light clutter density are given in Table 5.5.

Table5.5: RMS errors in position during the stop mode for Trajectory 1: Motion 2 - LC

Algorithms	RMS error in position (m)
IMM-PDAF	66.8421
Tracker	13.0193

5.1.2.2 Heavy Clutter Density

The RMS position error performances of the algorithms, the overall RMS errors averaged over the stop mode tracking periods are provided in Figure 5.10 and Table 5.6 respectively.

Table5.6: RMS errors in position during the stop mode for Trajectory 1: Motion 2 - HC

Algorithms	RMS error in position (m)
IMM-PDAF	84.6302
Tracker	24.7760

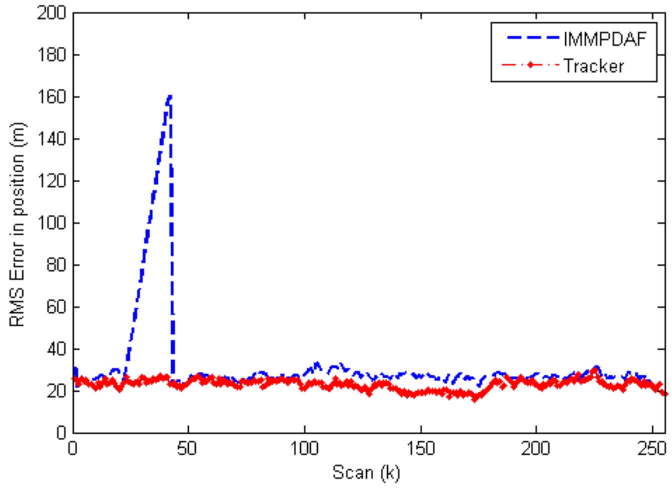

Figure 5.10: Trajectory 1: Motion 2 - RMS error in position for HC

Observations for Trajectory 1: Motion 2
This motion includes a stop in the offroad region. Both in the light and heavy clutter scenarios the stop mode is tracked effectively. As expected, the error level is higher in the heavy clutter scenario than the light clutter scenario. The performance of the Tracker is better during on road motions similar to the previous Motion 1.

The IMM-PDAF algorithm is able to find the stopped target after the stop period ends. This is due to the fact that a track initiation capability is added to the IMM-PDAF algorithm. The sudden rise in the position error in the IMM-PDAF occurs due to gate volume increases when the target stops. Since the IMM-PDAF algorithm assumes target existence, if there is no detection coming from the target, the PDAF output deteriorates rapidly and in turn this increases gate volume.

5.1.3 Motion 3

The trajectory has now 2 steps as shown in Figure 5.11. Motion 3 is illustrated in Figure 5.12. The target of interest performs move stop motions close to a real scenario. In this scenario,

- target evolves with a 1 m/s^2 deceleration while around the entry point B,
- stops for 20 secs just before the entry point at point B in between time interval [64, 83],
- it starts accelerating with 1 m/s^2 at scan $k = 84$, and arrives the on-road joint at scan $k = 85$ and accelerates with 1 m/s^2 and moves at a constant speed during motion on road segment 1,

- it arrives at the junction by decelerating $1m/s^2$ in 8 secs at scan $k = 201$,
- it stops for 20 secs in between time interval [180, 199],
- it again accelerates and continues with constant speed until the termination of the motion.

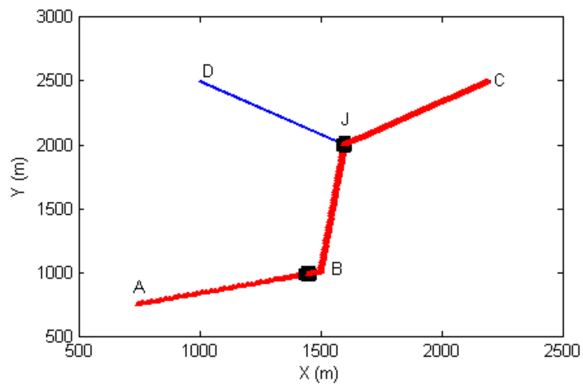

Figure 5.11: Trajectory 1: Motion 3 - Trajectory of the target

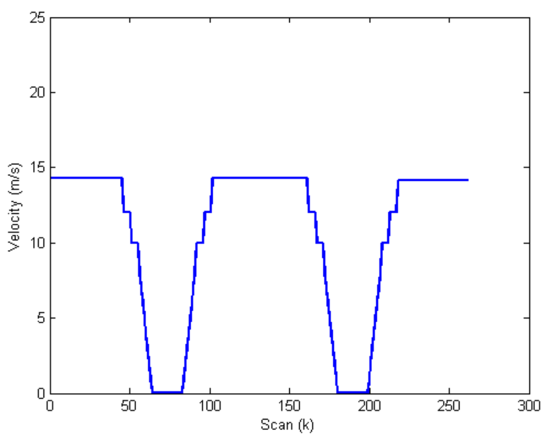

Figure 5.12: Trajectory 1: Motion 3 - Motion of the target

5.1.3.1 Light Clutter Density

The RMS position error performances of the algorithms are illustrated in Figure 5.13.

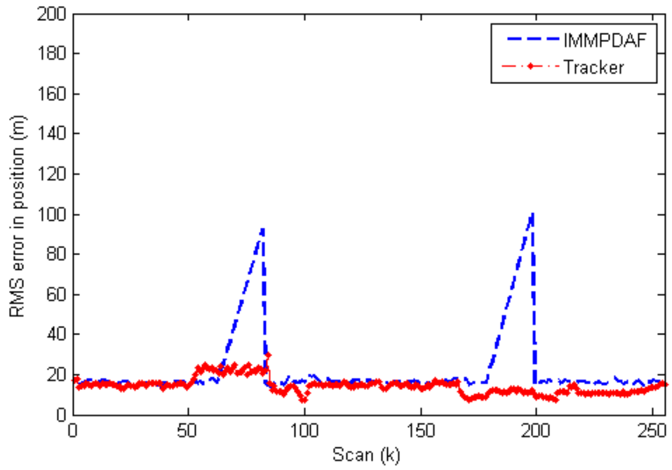

Figure 5.13: Trajectory 1: Motion 3 - RMS error performances - LC

The overall RMS errors of both algorithms are given in Table 5.7.

Table5.7: RMS errors in position during stop modes for Trajectory 1: Motion 3 - LC

Algorithms	RMS error in position (m) at entry Area	RMS error in position (m) at Junction Area
IMM-PDAF	59.2843	58.0139
Tracker	18.3431	12.8969

5.1.3.2 Heavy Clutter Density

The RMS position error performances for the algorithms in heavy clutter denisty are illustrated in Figure 5.14. The IMM-PDAF algorithms has significant peaks at the critical areas similar to the light clutter density case. The target of interest is dropped and initiated thereafter.

The overall RMS errors of the algorithms are given in Table 5.8.

Observation for Trajectory 1: Motion 3

As can be seen in position error performances, the Tracker indicates superior performance at stop durations. However, the Tracker's error level slightly goes up before stop regions due to deceleration of the target. In general, during onroad motions the Tracker's performance is better than the IMM-PDAF. It is also remarkable that in the case where a stop occurs at the junction, the Tracker is not confused and shows responsive performance.

Similar to Motion 2, the IMM-PDAF algorithm is able to find the stopped target after the stop period

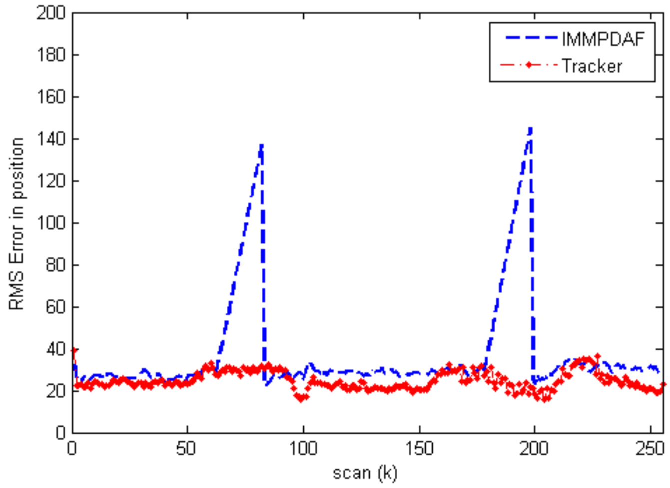

Figure 5.14: Trajectory 1: Motion 3 - RMS error in position for HC

Table5.8: RMS errors in position during stop modes for Trajectory 1: Motion 3 - HC

Algorithms	RMS Error in Position (m) at point B	RMS Error in Position (m) at junction
IMM-PDAF	83.6373	86.6142
Tracker	29.4370	26.9407

ends. The sudden rises in the position error in the IMM-PDAF occur again due to gate volume increases when the target stops.

5.2 TRAJECTORY 2

This trajectory includes turn with an obtuse angle at the entry point from offroad to onroad. The trajectory of the target is seen in Figure 5.15.

The velocity of the target along the trajectory is shown in Figure 5.16. In this scenario,

- Target starts its movement at point A and enters the road segment BJ at point B with an obtuse angle of 135 degrees in between time intervals [1,79],
- It arrives at the entry point at scan k = 80,
- It moves at a constant velocity until the junction point at scan k = 160,
- It continues its motion at a constant velocity as seen in Figure 5.16 and terminates its motion on road segment 2.

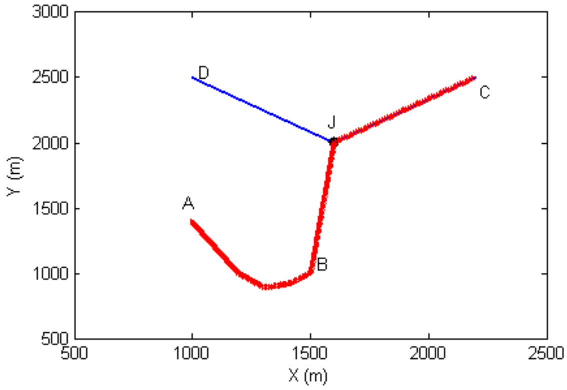

Figure 5.15: Trajectory 2: True target scenario with obtuse angle turn

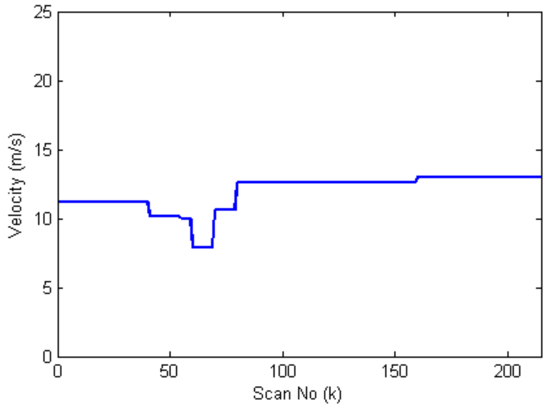

Figure 5.16: Trajectory 2: True velocity of the target

5.2.1 Light Clutter Density

The RMS error in target position and the RMS error in target velocity are illustrated in Figures 5.17 and 5.18 respectively. The overall RMS error values in position and velocity are given in Table 5.9.

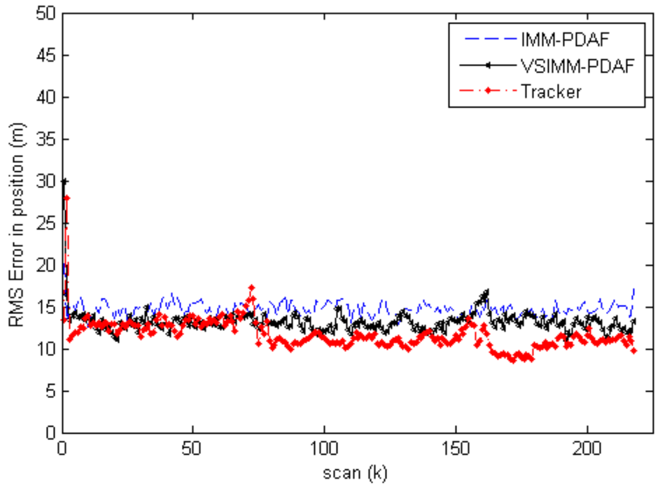

Figure 5.17: Trajectory 2: RMS error in position for LC

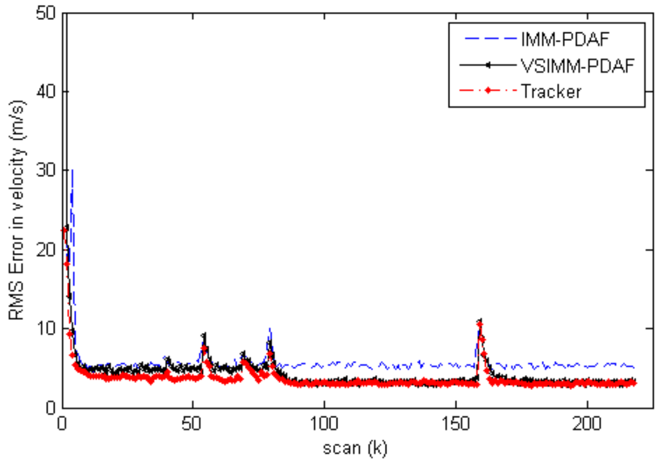

Figure 5.18: Trajectory 2: RMS error in velocity for LC

Table5.9: RMS errors in position and velocity for Trajectory 2 - LC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	14.7248	6.5213
VSIMM-PDAF	13.1904	6.0503
Tracker	11.6670	3.7574

5.2.2 Heavy Clutter Density

The RMS error in target position and the RMS error in target velocity are illustrated in Figures 5.19 and 5.20 respectively.

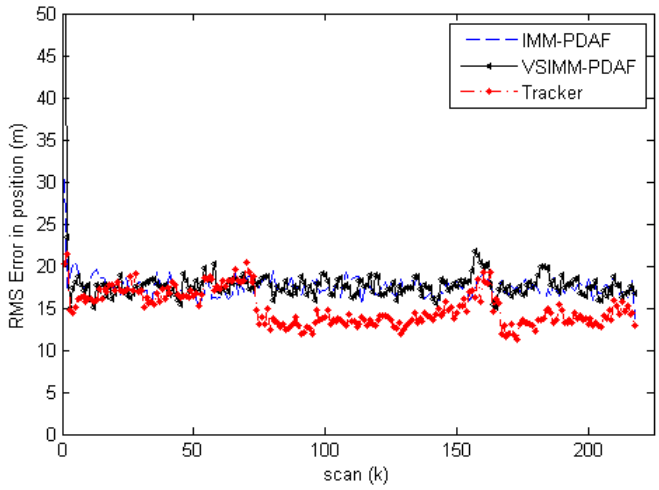

Figure 5.19: Trajectory 2: RMS error in position for HC

Table 5.10 shows the comparison of the algorithms in terms of overall RMS error values in position and velocity.

Table5.10: RMS errors in position and velocity for Trajectory 2 - HC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	17.5059	6.0592
VSIMM-PDAF	17.8085	6.4930
Tracker	14.9981	4.5694

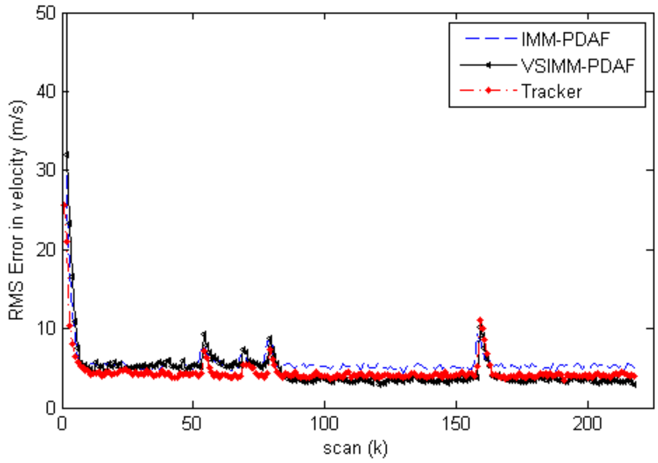

Figure 5.20: Trajectory 2: RMS error in velocity for HC

Observations for Trajectory 2

The observations pertaining to Trajectory 1 Motion 1 is valid for Trajectory 2. Regarding the offroad motions, performances are close to each other whereas the Tracker's performance resulted in better performance during onroad regions. In the heavy clutter scenario, the RMS errors in position and velocity increase, however the Tracker's performance is least effected among all. Also the performance improvement of the Tracker is more evident in the heavy clutter scenario.

All algorithms make a peak at transition points from offroad to onroad and junction areas. It is interesting to note that the velocity RMS error peaks in the road entry and in the junction have similar magnitude. The peaks that appear in RMS velocity during the offroad motions appear due to changes in velocity.

At the junction area, the Tracker makes a similar peak, however the width of the peak is wider than the others.

It is interesting to note that although the position error performance of the VSIMM-PDAF is similar to the IMM-PDAF, its velocity error performance is close to the Tracker. Using models that match to road map help estimating the velocity better in VSIMM-PDAF whereas this does not help a lot in estimating the position.

5.3 TRAJECTORY 3

This trajectory includes a turn with an acute angle at the entry point from offroad to onroad, which is in Figure 5.21.

The target velocity is illustrated in Figure 5.22.

- The target is first initiated at point A and evolves on a curve as in Figure 5.21 and passes to the on road situation at scan $k = 72$.

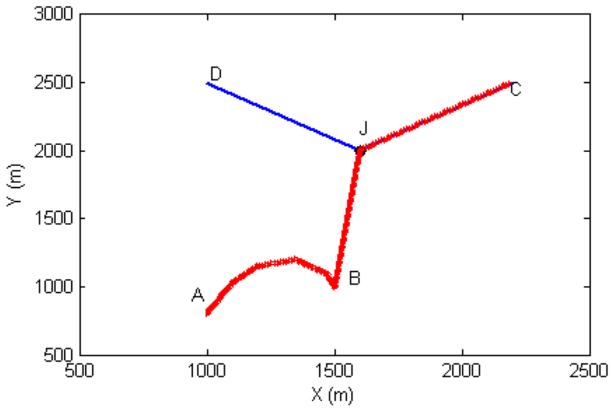

Figure 5.21: Trajectory 3: True target scenario with acute angle turn

- The target enters the road at B with a narrow angle of 45 degrees when in transition from offroad to onroad.
- It continues its motion at a constant velocity as in Figure 5.22.
- It arrives at the junction area at scan $k = 152$ and continues at a constant velocity on road segment 2 and then terminates its motion.

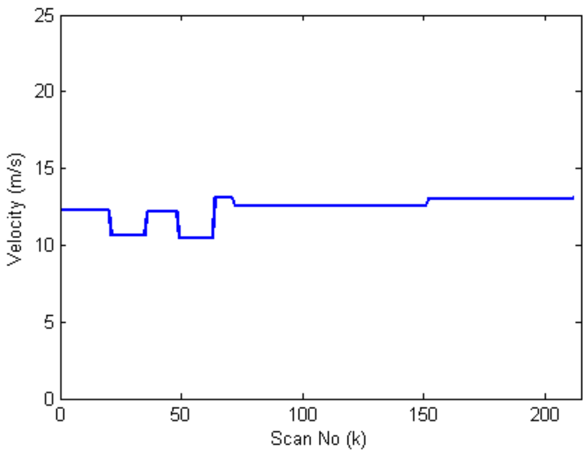

Figure 5.22: Trajectory 3: True velocity of the target

5.3.1 Light Clutter Density

The RMS error in position and the RMS error in veloicty are shown in Figure 5.23 and Figure 5.24 respectively.

The overall RMS error values in position and velocity are given in Table 5.11.

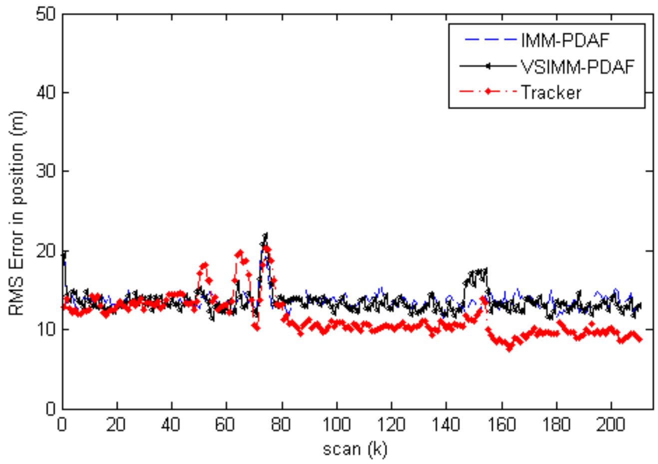

Figure 5.23: Trajectory 3: RMS error in position for LC

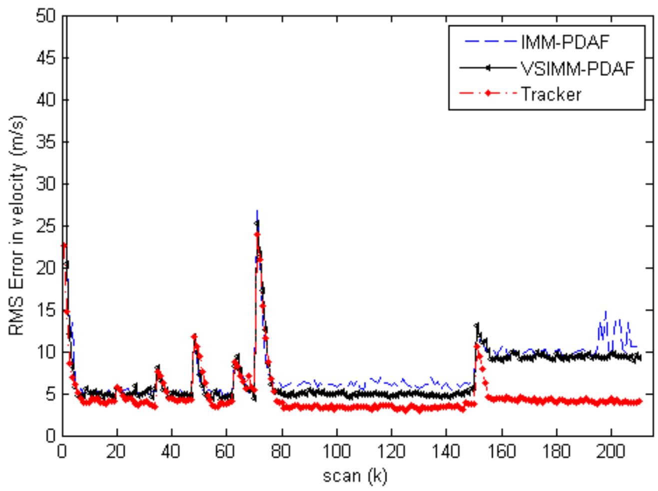

Figure 5.24: Trajectory 3: RMS error in velocity for LC

Table5.11: RMS errors in position and velocity for Trajectory 3 - LC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	13.6565	8.4627
VSIMM-PDAF	13.5102	7.7632
Tracker	11.6026	4.7238

5.3.2 Heavy Clutter Density

The RMS error in position and the RMS error in velocity performances are shown in Figure 5.25 and Figure 5.26 respectively.

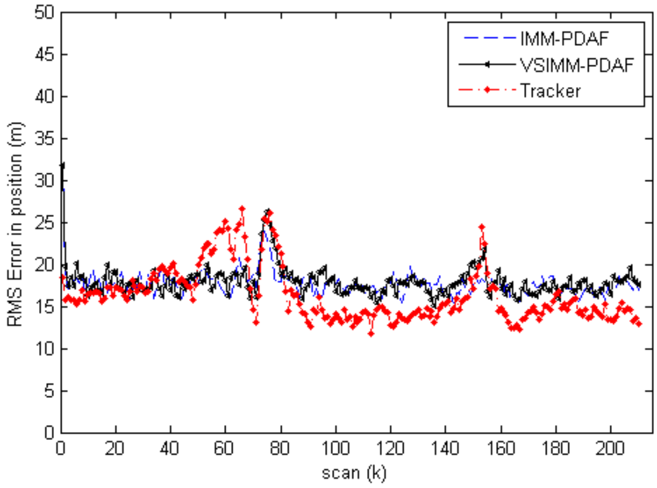

Figure 5.25: Trajectory 3: RMS error in position for HC

Table 5.12 compares the RMS error performances of the algorithms in high clutter density.

Table5.12: RMS errors in position and velocity for Trajectory 3 - HC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	17.7229	8.7124
VSIMM-PDAF	17.9303	7.9187
Tracker	16.3615	5.8593

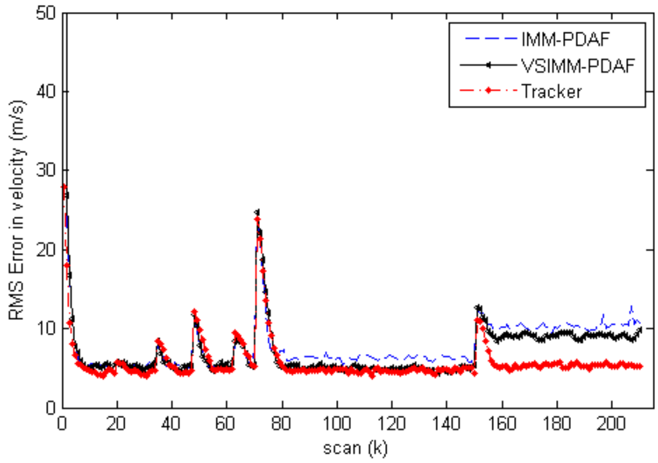

Figure 5.26: Trajectory 3: RMS error in velocity for HC

Observations for Trajectory 3

In general, the observations pertaining to Trajectory 2 are valid for Trajectory 3.

In specific terms, the RMS position error of the Tracker is affected most during offroad motion in which both a direction turn and velocity change occurred. The peaks in the RMS position error are due to this fact. This is even more evident in the heavy clutter scenario where the peaks have a wider structure. However in the RMS velocity error the peaks are present in all algorithms due to direction and velocity changes.

An interesting observation is that in the road segment JC the RMS velocity error of VSIMM-PDAF is now close to the IMM-PDAF in contrast to the Trajectory 2 where the RMS velocity error of VSIMM-PDAF is closer to the Tracker. This illustrates the fact that the velocity estimation of IMM and VSIMM are not reliable because the velocity characteristics of Trajectory 2 and Trajectory 3 are similar to each other in the road segment JC.

In the heavy clutter scenario, all the RMS errors in position and velocity increase, however the Tracker's performance is least effected among all. Also the performance improvement of the Tracker is more evident in the heavy clutter scenario.

5.4 TRAJECTORY 4

In this trajectory, there is an exit at the road segment 2 which is shown in Figure 5.27.

The velocity of the target is illustrated in Figure 5.28.

- The target for Trajectory 4 starts its motion in open field and arrives at the entry point B at scan $k = 61$,
- It passes to the onroad at point B and arrives at junction J at scan 161,

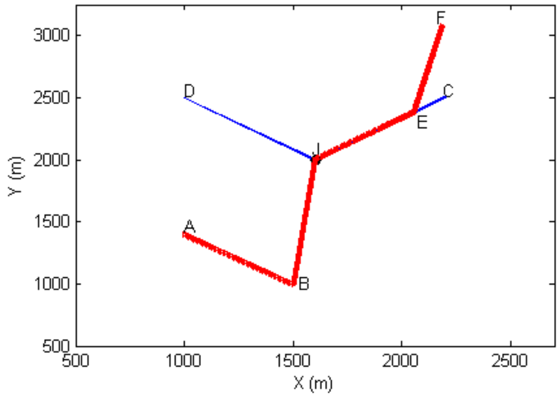

Figure 5.27: Trajectory 4: True target scenario

- On the road segment JC, the target changes its direction to the open field at point E by reducing its velocity at scan $k = 231$,
- It reduces its velocity and continues its motion at offroad for a while and stops at point F.

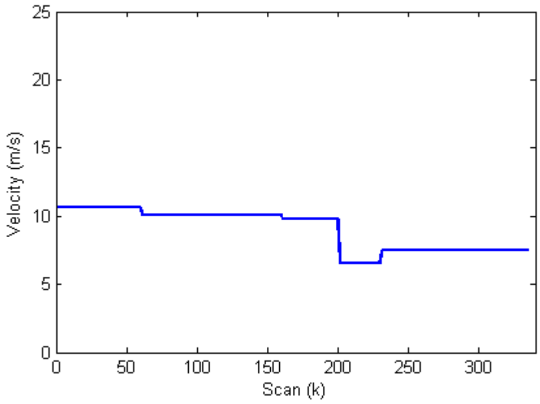

Figure 5.28: Trajectory 4: True velocity of the target

5.4.1 Light Clutter Density

The RMS error in position and the RMS error in velocity are demonstrated in Figure 5.29 and 5.30 respectively.

The overall RMS error values over the entire tracking period can be seen in Table 5.13.

5.4.2 Heavy Clutter Density

The RMS error in position and the RMS error in velocity performances of the algorithms are shown in Figure 5.31 and 5.32 respectively.

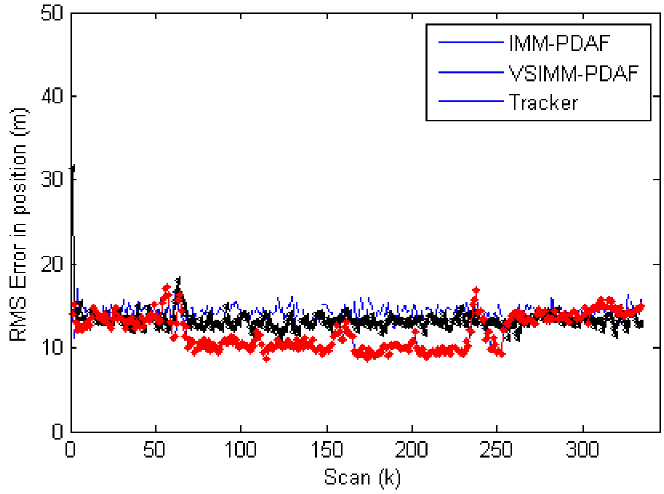

Figure 5.29: Trajectory 4: RMS error in position for LC

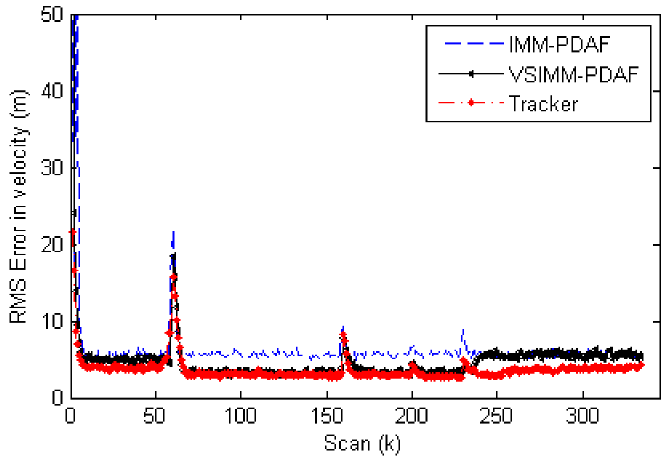

Figure 5.30: Trajectory 4: RMS error in velocity for LC

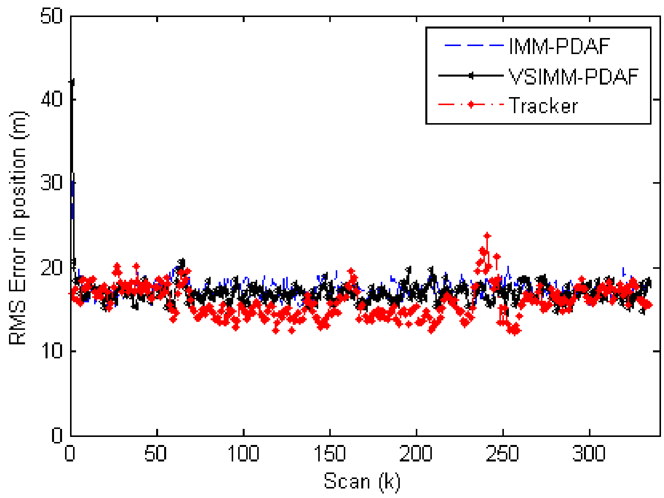

Figure 5.31: Trajectory 4: RMS error in position for HC

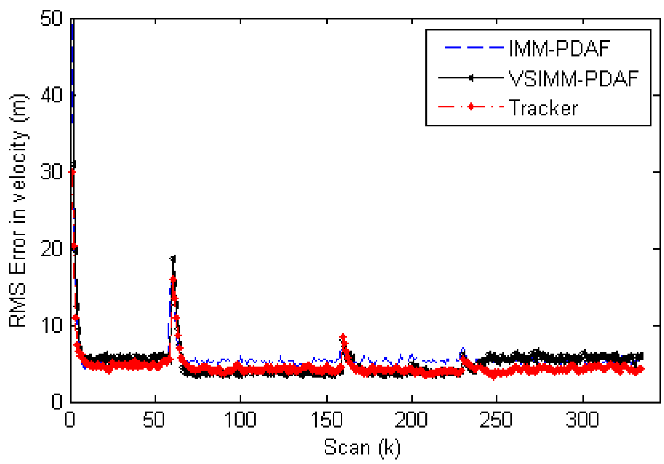

Figure 5.32: Trajectory 4: RMS error in velocity for HC

Table5.13: RMS errors in position and velocity for Trajectory 4 - LC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	14.3039	6.7448
VSIMM-PDAF	13.2978	5.7362
Tracker	11.7137	3.6795

The entire RMS error performances of the algorithms are given in Table 5.14.

Table5.14: RMS errors in position and velocity for Trajectory 4 - HC

Algorithms	RMS error in position (m)	RMS error in velocity (m/s)
IMM-PDAF	17.6528	5.8639
VSIMM-PDAF	17.0138	5.1379
Tracker	15.8803	4.6103

Observations for Trajectory 4

The observations pertaining to Trajectory 1 Motion 1 are valid for Trajectory 4 as well. In addition to them, the Tracker handles the offroad exit effectively. During the offroad motion in the last part of the scenario, performance of all algorithms are similar. During onroad motions the VSIMM-PDAF's performance is between the IMM-PDAF and the Tracker.

Since the VS-A-IMM tracker uses parallel branches to estimate the state, onroad entries and offroad exits with small angles between the target trajectory and the road may cause problems. This problem can be possibly eliminated by tracking the direction of the movement locally.

5.5 TRAJECTORY 5: Tunnel Approach I - Constant Velocity Assumption

The trajectory of the tunnel scenario is presented in Figure 5.33. The target starts at point A in open field and ends its offroad motion at point B on road segment BJ. On road segment JC, there is the tunnel information available for the Tracker. Points I and II indicate tunnel entrance point and exit point in the X-Y plane respectively. The target is invisible to the sensor and the Tracker assumes that the target moves at constant speed with the velocity lastly detected by the sensor. After passing through the tunnel, it stops at point C. The velocity of the true target is shown in Figure 5.34. The target starts onroad movement at scan $k = 51$ and arrives at junction at scan $k = 130$. Its velocity is kept constant on road segment JV, which means that the speed of the target in the tunnel is 10 m/s.

The RMS error both in position and velocity of the Trajectory 5 are depicted in Figure 5.35 and Figure 5.36.

Observations for Trajectory 5 Approach I

The Tracker presents expected results as observed in the previous trajectories when the target of interest moves in the uncertainties such as offroad/onroad transitions and junction. Because onroad model set,

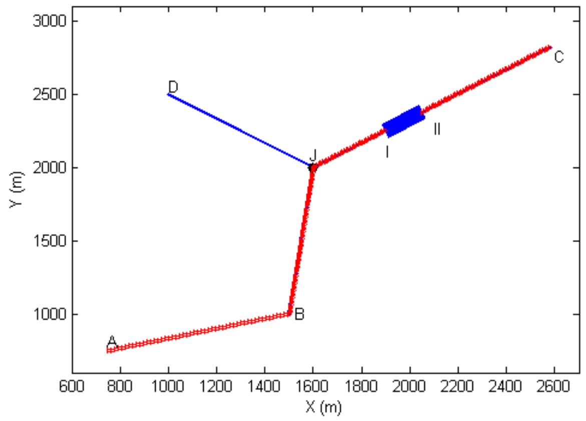

Figure 5.33: Trajectory 5: True target scenario

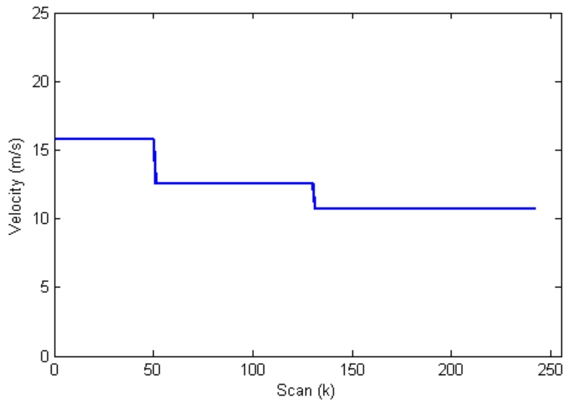

Figure 5.34: Trajectory 5: True velocity of the target

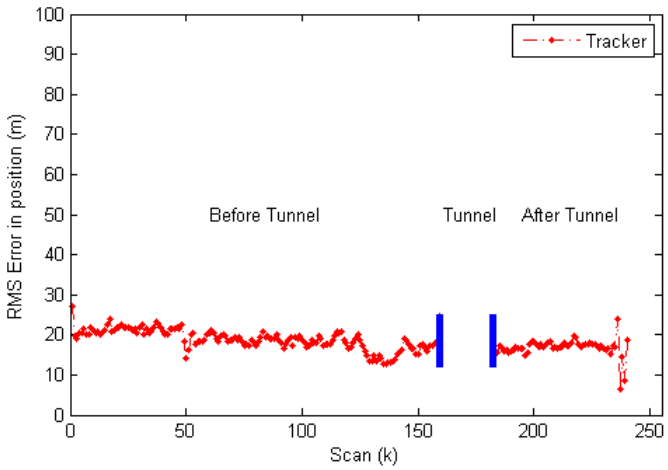

Figure 5.35: Trajectory 5: Approach I - RMS error in position

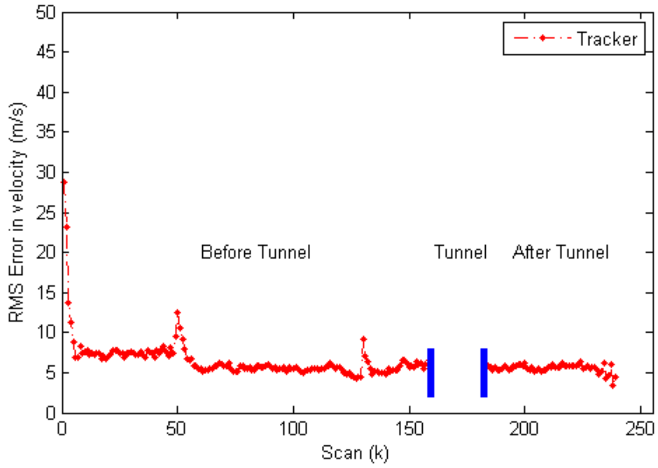

Figure 5.36: Trajectory 5: Approach I - RMS error in velocity

which has three models, provides a priori information regarding the road segments, this leads smaller error probabilities along the road. The Tracker's performance at the output of the tunnel results in good performances both in position and velocity as long as the motion of the target in a topographic constraint is known and deterministic.

The overall RMS error values in position and velocity are provided in Table 5.15.

Table5.15: Overall RMS errors in position and velocity for Trajectory 5: Approach I

Algorithm	RMS error in position (m)	RMS error in velocity (m/s)
Tracker	18.4593	5.5302

When target goes through the tunnel, there is a reasonable assumption made that the target moves at a constant speed while being invisible to the radar. Otherwise, the target loss would occur due to the driver's unprecedented action in the tunnel.

5.6 TRAJECTORY 5: Tunnel Approach II - Hide Model Proposed

5.6.1 Motion 1

The true velocity of the target is shown in Figure 5.37 with the trajectory given in 5.33. The target stops in the tunnel for 30 secs and then continues to move at 10 m/s and then exits tunnel with the same speed. The total duration in the tunnel is 50 secs.

The RMS error performances in position and velocity of the Trajectory 5 for Approach II - Motion 1 are depicted in Figure 5.38 and Figure 5.39 respectively.

Table 5.16 shows the average position and velocity errors over the recovery time.

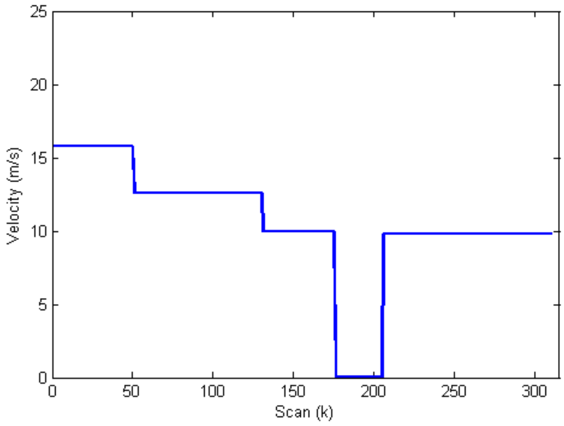

Figure 5.37: True velocity of the target

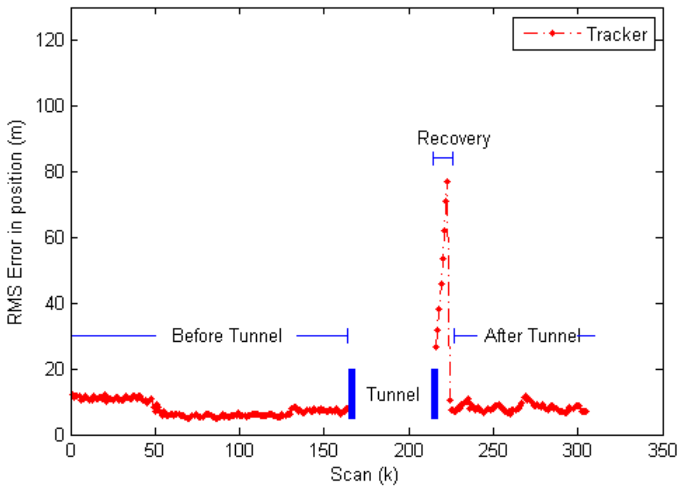

Figure 5.38: Trajectory 5: Approach II - Motion 1 - RMS error in position

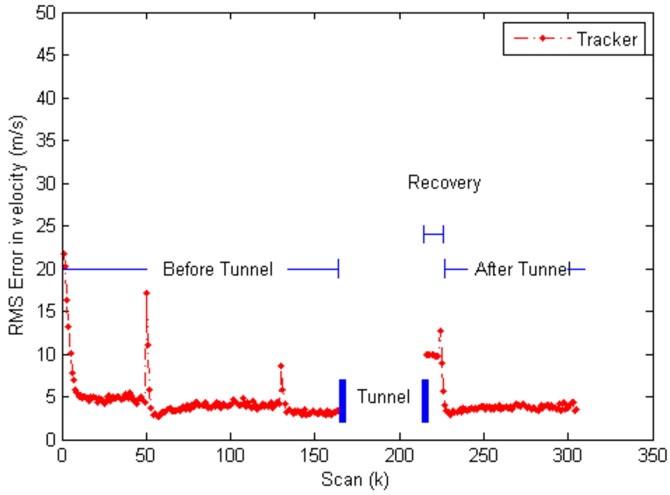

Figure 5.39: Trajectory 5: Approach II - Motion 1 - RMS error in velocity

Table5.16: RMS errors in position and velocity for Trajectory 5: Approach II - Motion 1 during the recovery time

Algorithm	RMS error in position (m)	RMS error in velocity (m/s)
Tracker	46.2389	10.1178

5.6.2 Motion 2

The true velocity of the target is shown in Figure 5.40 for the trajectory in Figure 5.33. The target accelerates along the tunnel movement in this scenario and jumps out of the tunnel at a speed of 25 m/s (or 90 km/h). This is usually the maximum speed limit in tunnels.

The RMS errors in position and velocity of the Trajectory 5 are provided in Figure 5.41 and Figure 5.42 respectively.

Average position and velocity errors over the recovery period are given in Table 5.17.

Table5.17: RMS errors in position and velocity for Trajectory 5: Approach II - Motion 2 over the recovery duration.

Algorithm	RMS error in position (m)	RMS error in velocity (m/s)
Tracker	37.3469	15.0314

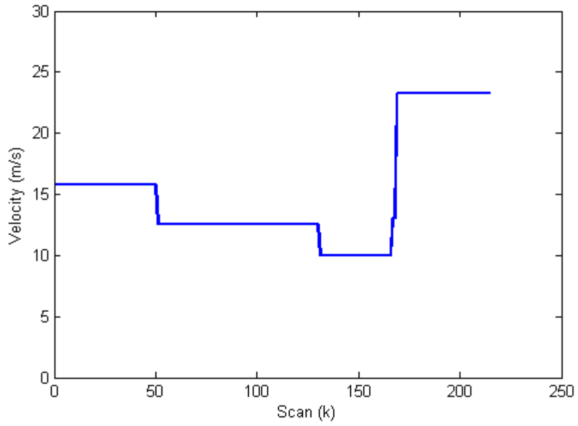

Figure 5.40: True velocity of the target

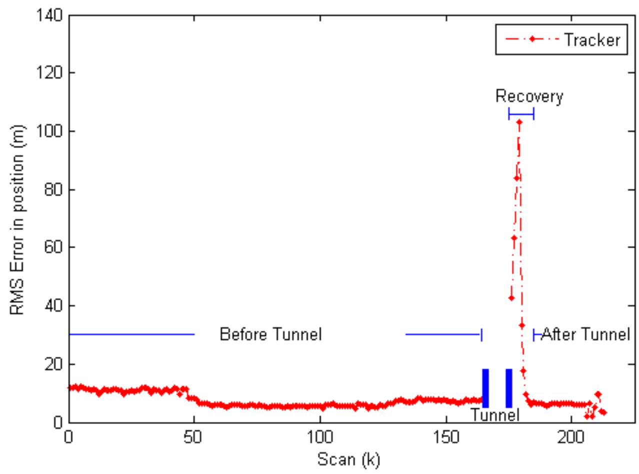

Figure 5.41: Trajectory 5: Approach II - Motion 2 - RMS error in position

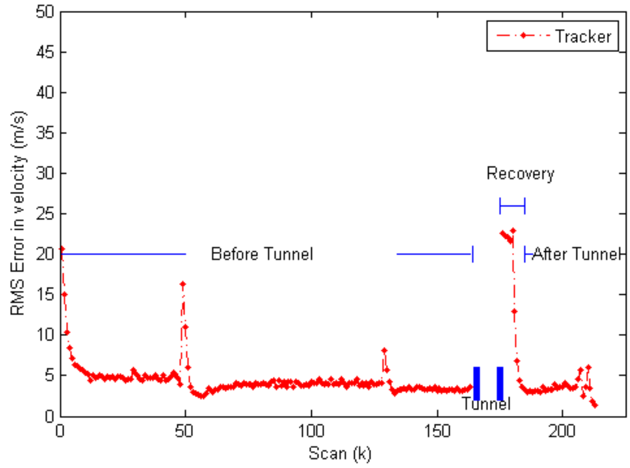

Figure 5.42: Trajectory 5: Approach II - Motion 2 - RMS error in position

Observations for Trajectory 5 Approach II for Both Motions

When the target rushes out of a topographic constraint with higher velocity, the Tracker has a higher peak in terms of the position error compared to the exit with lower velocity. However, the average recovery time has smaller error in the exit with higher speed than the exit with low speed.

Regarding the velocity error performance, both recovery time to catch the target and the peak value increase proportionally as the speed of the target increases.

The Approach II outperforms the Approach I since making an assumption would not be possible in realistic scenarios. When target moves evasively and misleads the tracker, this approach provides a practical mechanism to catch the target without taking into account how long it would take for the target to exit the topographic constraint (hence no track loss).

CHAPTER 6

CONCLUSIONS

A novel simple and elaborate method for tracking ground targets, referred to as Variable Structure-Autonomous-Interacting Multiple Model (VS-A-IMM) method, is developed from first principles of estimation and multiple model filtering. Multiple model estimation plays an important role for targets which do not follow the same pattern of motion. Each model output which correspond to a different dynamical model resulting state estimate are combined to obtain a final state estimate. Different combining strategies can be developed, the most common of which is the MMSE estimate. It is the most well known form for example to obtain the final estimate of the IMM algorithm. MAP estimation is another alternative which is much less known to the community.

Multiple model estimation opens up possibilities to combine estimators. In ground target tracking, grouping of models that relate to the same topographic region is a natural choice. These topographic regions are off-road areas and different road segments. Each region can then run its VSIMM algorithm. Obviously the final state estimate in each region is the MMSE estimate. The estimates in each region should then be combined. We have opted out an AMM approach in combining different regions because it is obvious that no mixing of states is needed once a target is confirmed it is off-road or on a road segment. Not only is this approach suitable to the nature of the problem but also it maintains a low computational complexity. Hence, a new theoretical multiple model estimation framework that models the target motion in a more realistic way is presented.

Grouping of models in off-road and road segments and selecting the appropriate output among the model outputs leads to a responsive target tracker with a low computational complexity.

A target tracker should support all the possible states of the target which may well include stopping and hiding. The target may appear intermittently due to obstructions. A stop model and a hide model are integrated into the tracker. Integrated PDAF approach is also utilized in the new algorithm to support intermittent detections of the same target. Target initialization after temporary loss of target is also an integral part of the tracker. This should not be confused with the complete loss of target in which case the target is deleted in the target database. The presence of dense background clutter, which obviously causes deterioration in target track quality, is inevitable in ground target tracking systems. Along with the use of methods that measures track quality, such as IPDAF, smart methods that would define the validation regions (gating) is an apparent necessity which has not attracted attention in the literature. It is observed that when effective validation regions are placed in the VS-A-IMM algorithm, position errors are reduced. Simulation scenarios are concentrated on measuring the position error which is the ultimate performance metric of the ground target tracker systems. It is observed that the simulation results confirm the theoretical predictions. The VS-A-IMM algorithm is more responsive than the standard IMM approaches. The reasons are that

- A priori information is used cleverly (groups of models are formed in geographic regions together with appropriate gating)
- Estimation framework matches the reality
- Almost least number of models are used in each group

The simulations are performed in dense clutter environments. The results show that the developed algorithm is effective in tracking targets when background clutter is present. The simulation results of stopping and hiding target cases indicate that the target is tracked effectively even if the target is stopped or is hidden for instance in a tunnel. In summary the ground target tracker developed in this thesis with the properties described below is an effective tracker compared to standard IMM approaches in terms of position error performance. The price paid for this is a minor increase in computational complexity.

- Theoretical framework for combined MMSE and MAP estimation
- Combined AMM and IMM tracker
- Variable structureness both in AMM and IMM parts
- Stop model and hide model
- Smart gating
- Track quality measure and track initiation after temporary loss

Among the future work planned to improve the VS-A-IMM tracker are the extension to multiple target cases, development of different combined estimators of MAP and MMSE, improvements of the stop model and the hide model. Countermeasures (e.g., jamming) used by targets is a source of track loss in ground target trackers. The VS-A-IMM tracker should be made immune to such jamming applications. The research along this line should have the concept of developing parallel hypotheses, the most probable of which should be selected as the output.

Multi-sensor approaches can also be developed to enhance the tracking quality of targets. The modular composite architecture of the VS-A-IMM algorithm lends itself to multi-sensor applications. For example different branches in the algorithm can be obtained from different sensors. Multiple sensor detection is important in the sense that the geographical diversity of the sensors compensate the weaknesses of single sensors. The IPDAF approach would play an important role to qualify the track formations and maintenance in each sensor. The IPDAF approach can be expanded to cover multi-sensors so that track observability in each sensor is measured and than utilized to fuse the state estimates.

REFERENCES

[1] Agate, C., Sullivan, K., Road Constrained Target Tracking and Identification Using a Particle Filter, Proceedings of SPIE: Signal and Data Processing of Small Targets, August 2003.

[2] Arulampalam, S., Gordon, N., Orton, M.,Ristic, B., A Variable Structure Multiple Model Particle filter for GMTI Tracking, 5th International Conference on Information Fusion, FUSION 2002.

[3] Bar-Shalom, Y., Chang, K.C, Blom, H., Multitarget Multisensor Tracking, Automatic Track Formation in Clutter with a Recursive Algorithm, Artech House, 1990.

[4] Bar-Shalom, Y., and X.R. Li, Estimation and Tracking: Principles, Techniques and Software, Artech House, 1993.

[5] Bar-Shalom, Y., Tse, E., Tracking in a Cluttered Environment with Probabilistic Data Association, Automatica, Vol. 11, September, 1975.

[6] Bar-Shalom, Y., Fortmann, T. E., Scheffe, M., Joint Probabilistic Data Association for Multiple Targets in Clutter, Information Sciences and Systems Conference, 1980.

[7] Blackman, S., Multiple Hypothesis Tracking For Multiple Target Tracking, IEEE Transactions Aerospace and Electronic Systems, vol. 19, Jan. 2004.

[8] Blackman, S., and Popoli, R., Design and Analysis of Modern Tracking Systems, Boston, MA: Artech House, 1999.

[9] Blom, H.A.P., and Bar-Shalom, Y., The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients, IEEE Transactions Automatic Control, Vol. 33, pp. 780-783, August 1988.

[10] Clark, J.M.C., Kountouriotis, P.A., Vinter, R.B., A Methodology for Incorporating the Doppler Blind Zone in Target Tracking Algorithms, 11th International Conference on Information Fusion, FUSION 2008.

[11] Dunn R.J., Bingham P.T., Fowler, C.A.B., Ground Moving Target Indicator Radar and the Transformation of US Warfighting, February 2004.

[12] Goldstein, J.S., Picciolo M.L., Rangaswamy, M., Griesbach, J.D., Detection of Dismounts Using a Synthetic Aperture Radar, IEEE International Radar Conference (9th), May 2010.

[13] Hall, D.L., Llinas , J., An Introduction to Multisensor Data Fusion, Proceedings of the IEEE, Vol.85, No. 1, January 1997.

[14] Helmick, R. E. and Watson, G. A., IMM-IPDAF for Track Formation on Maneuvering Targets in Cluttered Environments, SPIE Digital Library, Vol.2235, 2008.

[15] Hong-Quan, Q., Shao-Hong, L., The Model Set Multiple Hypotheses IMM Algorithm for Maneuvering Target Tracking, International Conference on Signal Processing Proceedings, ICSP, 2008.

[16] Houles, A., and Bar-Shalom, Y., Multisensor Tracking of a Maneuvering Target in Clutter, IEEE Transactions Aerospace Electronics Systems, vol.25 pp.176-189, 1989.

[17] Julier, S., Uhlmann, J., Durrant-Whyte, H.F., A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators, IEEE Transactions on Automatic Control, vol. 45, no. 3, pp. 477-482, 2000.

[18] Kalandros, M.K, Trailovic , L., Pao, L.Y., and Bar-Shalom, Y., Tutorial on Multisensor Management and Fusion Algorithms for Target Tracking, Proceeding of 2004 American Control Conference, 2004.

[19] Kalman, R.E., A New Approach to Linear Filtering and Prediction Problems, Transactions of the ASME, Journal of Basic Engineering, March 1960.

[20] Kay, S. M., Fundamentals of Statistical Signal Processing Volume I-Estimation Theory, Prentice Hall, 1993.

[21] Kirubarajan, T., Bar-Shalom, Y., Probabilistic Data Association Techniques for Target Tracking in Clutter, Proceedings of the IEEE, 92, pp. 536-556, 2004.

[22] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K.R., Kadar, I., Ground Target Tracking with Topography-based Variable Structure IMM Estimator, Proceedings of SPIE - The International Society for Optical Engineering, pp. 222-233.,1998.

[23] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K.R. Ground Target Tracking with Variable Structure IMM Estimator, IEEE Transactions on Aerospace and Electronic Systems, 36 (1), pp. 26 - 46, 2000.

[24] Koch, W., and Klemm, R., Ground Target Tracking with STAP Radar, IEE Proceedings, Radar Sonar and Navigation, 148, No. 3, 173 - 185, 2001.

[25] Kodagoda, K. R. S., et al., IMMPDAF Approach for Road-Boundary Tracking, IEEE Transactions on VT., Vol.56, 2, 2007.

[26] Lan, J., Li, X.R., Mu, C., Best Model Augmentation for Variable-Structure Multiple-Model Estimation, IEEE Transactions on Aerospace and Electronic Systems, 2011.

[27] Lancaster, J., Blackman,S., and Taniguchi, E., Joint IMM/MHT Tracking and Identification with Application to Ground Tracking, Proceedings of SPIE, 2005.

[28] Li, X.R., Bar-Shalom, Y., Design of an Interacting Multiple Model of an Air Traffic Control, IEEE Transactions on Control Systems Technology, 1(3), 186-194,September 1993.

[29] Li, X.R., and Bar-Shalom, Y., Multiple-Model Estimation with Variable Structure, IEEE Transactions Automatic Control, Vol. 41, pp. 478-493, April 1996.

[30] Li, X.R., Jilkov, V.P., A Survey of Maneuvering Target Tracking: Dynamic Models, In Signal and Data Processing of Small Targets 2000, Proceedings of SPIE, Vol. 4048, 2000.

[31] Li, X.R., Jilkov, V.P., Survey of Maneuvering Target Tracking PART I: Dynamic Models, IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 4, 2003.

[32] Li, X.R., Jilkov, V.P., A Survey of Maneuvering Target Tracking - Part III: Measurement Models, Proceedings of SPIE - The International Society for Optical Engineering, 4473, pp. 423-446, 2001.

[33] Li, X.R., Jilkov, V.P., A Survey of Maneuvering Target Tracking - Part V: Multiple Model Methods, IEEE Transactions on Aerospace and Electronic Systems vol. 41, no. 4, October 2005.

[34] Li, X.R., Jilkov, V.P., Ru, J., Multiple-Model Estimation with Variable Structure - Part VI: Expected-Mode Augmentation, IEEE Transactions on Aerospace and Electronic Systems, pp. 853 - 867, 2005.

[35] Li, X.R., Multiple-Model Estimation with Variable Structure - Part II: Model-Set adaptation, IEEE Transactions on Automatic Control, 45 (11), pp. 2047-2060, 2000

[36] Li, X.R., Zhang, Y.M., and Zhi, X.R., Multiple Model Estimation with Variable Structure - Part IV: Design and Evaluation of Model-Group Switching Algorithm, IEEE Transaction Aerospace and Electronic Systems, Jan. 1999.

[37] Li, X.R., Zhao, Z.L., Zhang, P. and He,C., Model - Set Design for Multiple Model Estimation - Part II: Examples, Proceedings of 2002 International Conference, on Information Fusion, FUSION 2002.

[38] Li, X.R., Zhi, and Y. M. Zhang, Multiple - Model Estimation with Variable Structure - Part III: Model - Group Switching Algorithm, IEEE Trans. Aerospace and Electronic Systems, Jan. 1999.

[39] Li, X.R., Model - Set Adaptation in Variable - Structure MM Estimation by Hypothesis Testing, American Control Conference, 1998.

[40] Li, X.R., Zhang, Y., Multiple - Model Estimation with Variable Structure - Part V. Likely - Model Set Algorithm, IEEE Transactions on Aerospace and Electronic Systems, 2000.

[41] Li, X.R., Bar-Shalom, Y., Multiple - Model Estimation with Variable Structure, IEEE Transactions on Automatic Control, 1996.

[42] Liu, Y., Li, X.R., Sequential Multiple - Model Detection of Target Maneuver Termination, 14th International Conference on Information Fusion, FUSION 2011.

[43] Maresca S., Greco M., Gini F. Verrazzani L., Radar Tracking of a Move - Stop - Move Maneuvering Target in Clutter, IEEE Radar Conference, 2008.

[44] Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J., Interacting Multiple Model Methods in Target Tracking: A Survey, IEEE Transactions on Aerospace and Electronic Systems, Vol. 34, No. 1, January 1998.

[45] Mertens, M., Ulmke, M., Ground moving target tracking with context information and a refined sensor model, 11th International Conference on Information Fusion, FUSION 2008.

[46] Musicki, D., Challa, S., Suvorova, S., Automatic Track Initiation of Maneuvering Target in Clutter, Proc. ASCC 2004, 2004.

[47] Musicki, D., Evans, R., and Stankovic, S., Integrated Probabilistic Data Association, IEEE Transactions on Automatic Control, Vol. 39, No.6, June,1994.

[48] Musicki, D., Evans, R., Joint Integrated Probabilistic Data Association - JIPDA, ISIF 2002.

[49] Pannetier, B., Benameur, K., Nimier, V., Rombaut, M., VS - IMM Using Road Map Information for a Ground Target Tracking, 8th International Conference on Information Fusion, FUSION 2005.

[50] Pannetier, B., Benameur, K., Nimier, V., Rombaut, M., Ground Moving Target Tracking with Road Constraint, Proceedings of SPIE - The International Society for Optical Engineering, 2004.

[51] Pitre, R.R., Jilkov, V.P., and Li, X.R., A Comparative Study of Multiple - Model Algorithms for Maneuvering Target Tracking, Proceedings of SPIE on Signal Processing, Sensor Fusion, and Target Recognition, vol. 5809, 2005.

[52] Ru, J., Rong Li, X., Jilkov, V.P., Bashi, A., Sequential Detection of Target Maneuvers, 8th International Conference on Information Fusion, FUSION 2005.

[53] Semerdijev, E., Mihaylova, L., Li, X.R., Variable and Fixed Structure Augmented IMM Algorithms Using Coordinated Model, 3rd International Conference on Information Fusion, FUSION 2000.

[54] Quanxin, D., Guowei, L., Ye, T., Hong, J., Adaptive Variable Structure Multiple Model Filter for High Maneuvering Target Tracking, International Conference on Computational and Information Sciences (ICCIS), 2010.

[55] Ru, J., Li, X.R., Variable - Structure Multiple - Model Approach to Fault Detection, Identification, and Estimation, IEEE Transactions on Control Systems Technology, 2008.

[56] Shea, P.J., Zadra, T., Klamer, D., Frangione, E., and Brouillard, R., Improved State Estimation Through Use of Roads in Ground Tracking, Proceedings of Signal and Data Processing of Small Targets, vol. 4048, SPIE, 2000.

[57] Skolnik M. I., Introduction to Radar Systems, McGraw - Hill, 1962.

[58] Streller, D., Road Map Assisted Ground Target Tracking, 11th International Conference on Information Fusion, FUSION 2008.

[59] Ulmke M., and Koch, W., Road - Map Assisted Ground Moving Target Tracking, IEEE Transactions on Aerospace and Electronic Systems, 42, No. 4, 2006.

[60] Wald, A., Sequential Analysis. New York: Wiley, 1947.

[61] Wang, X., Challa,S., and Evans, R. J., Gating Techniques for Maneuvering Target Tracking in Clutter, IEEE Transactions on Aerospace and Electronic Systems, Vol. 38, No. 3, July 2002.

[62] Wang, X., Challa,S., Evans, R., Maneuvering Target Tracking in Clutter Using VSIMM - PDA, Proceedings of the SPIE: Signal Processing, Sensor Fusion and Target Recognition XII, 2003.

[63] Welch G., Bishop G., An Introduction to the Kalman Filter, In Proceedings of the ACM SIGGRAPH Conference, ACM Press, August 2001.

[64] Zhang, S., Bar-Shalom, Y., Tracking Move - Stop - Move Targets with State - Dependent Mode Transition Probabilities, Proceedings of SPIE - The International Society for Optical Engineering, 2009.

[65] Zhang, S., Bar-Shalom, Y., Track Segment Association for Ground Moving Targets with Evasive Move - Stop - Move Maneuvers, Proceedings of SPIE - The International Society for Optical Engineering, 2010.

CURRICULUM VITAE

PERSONNEL INFORMATION

Surname, Name : Alat, Gökçen
Nationality : Turkish (TC)
Date and Place of Birth : 18.09.1973, Adana
Marital Status : Single

EDUCATION

Degree	Institution	Year of Graduation
MSc	Çankaya University Electrical and Communication Eng.	2004
MSc	Çankaya University Management	2002
BSc	Çukurova University Electrical and Electronics Eng.	1994

PUBLICATIONS

1. Alat, G., Leblebicioğlu, K., Manevralı ve Dur-Kalk Hareket Eden Karasal Bir Hedefin Yüksek Gürültülü Ortamda Takibi, TOK 2012, Niğde.

2. Alat, G., Leblebicioğlu, K., Baykal, B., Variable Structure Autonomous Interacting Multiple Model Algorithm (VS-A-IMM) for Ground Target Tracking, Submitted to Aerospace Science and Technology, Elsevier, Kasım 2012.

3. Alat, G., Leblebicioğlu, K., Topoğrafik Kısıtlar Nedeniyle Görünmeyen Karasal bir Hedefin Takibi, Submitted to SIU 2013, Kıbrıs.