RANSAC-Based Track Initiation for Kalman Filtering and Multi-Hypothesis Tracking Systems in Cluttered Range-Only Environments

by

Glenn Overmyer

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
July 20, 2026

Keywords: RANSAC, Kalman Filter, Multi-Hypothesis Tracking, Range-Only Radar, Track Initiation, Radar Tracking, Computer Vision Tracking

Approved by

Scott Martin, Chair, Professor of Mechanical Engineering
David Bevly, Bill and Lana McNair Distinguished Professor
Ehsan Taheri, Professor of Aerospace Engineering

Abstract

Track initiation in cluttered range-time surveillance environments is a challenging problem. Classical M-of-N initiation methods confirm tracks based on detection count alone within a static gate, with no geometric validation, making them susceptible to false track initialization as clutter density increases. This thesis proposes a RANSAC (Random Sample Consensus) based track incubation framework that utilizes the linear geometry of constant velocity targets in the range-time plane to differentiate genuine tracks from clutter before promotion. Candidate tracks are confirmed only when a sufficient number of detections support a linear hypothesis under random sampling, and promoted tracks are initialized using a Weighted Least Squares fit that provides range, range-rate, and covariance.

The incubation framework is evaluated in combination with two engines, a Global Nearest Neighbor Kalman Filter and a Track-Oriented Multiple Hypothesis Tracker, and benchmarked against M-of-N initialized counterparts across three simulated collision and debris scenarios under nominal, heavy clutter, and reduced detection probability conditions. While both M-of-N engines collapsed to negative MOTA under heavy clutter, the RANSAC-TOMHT engine maintained a positive MOTA (Multiple Object Tracking Accuracy). The RANSAC-GNNKF engine produced a MOTA that was orders of magnitude higher than M-of-N. This result demonstrates that geometric validation at initiation is beneficial for reliable tracking in high clutter environments. RANSAC engines outperformed M-of-N counterparts on MOTA, MOTP, and RMSE, at the cost of increased runtime. A Monte Carlo validation study confirmed that the WLS covariance is theoretically sound but subject to overconfidence due to inlier set contamination in clutter, an inherent limitation of RANSAC-based initiation.

Artificial Intelligence (AI) Use Disclosure Statement

In the preparation of this thesis, Claude (Anthropic) was used for spelling and grammar checking, as a thesaurus, and for LaTeX formatting and troubleshooting assistance. The author acknowledges full responsibility for the intellectual content of this work.

Accessibility Disclosure

In the preparation of this thesis, the following digital accessibility tools were used to ensure this document complies with federal requirements: Adobe Acrobat's accessibility checker was used to verify WCAG compliance, and the document passed this accessibility check. The author acknowledges full responsibility for the intellectual content of this work and has made a good faith effort to comply with digital accessibility requirements in publishing, wherein the nature of the content does not significantly change in order to do so. Furthermore, all content has been reviewed and revised to meet these requirements prior to final publication.

Acknowledgments

The author would like to thank Dr. Scott Martin for his guidance and support throughout this work. His insight into Least Squares initialization and its application to the track promotion problem was instrumental in shaping the theoretical foundation of this thesis.

The author would also like to thank Dr. David Bevly for encouraging a return to academia and providing the opportunity to pursue this work. Thanks are also due to Dr. Ehsan Taheri for his time and feedback as a member of the committee.

The author would like to thank labmate Jamison Hood for the original idea of applying RANSAC to the track initiation problem, which served as the starting point for this work.

Finally, the author would like to thank his parents and in-laws for their encouragement and support, and his wife Helena and son Augustine, who are the motivation behind it all.

Table of Contents

Abstract 2

Artificial Intelligence (AI) Use Disclosure Statement 3

Accessibility Disclosure 4

Acknowledgments 5

List of Tables 9

List of Figures 11

List of Abbreviations 16

Chapter 1 Introduction 18

1.1 Problem Statement 18

1.2 Proposed Approaches 19

1.3 Contributions 20

Chapter 2 Background and Prior Work 22

2.1 Weighted Least Squares 23

2.2 Kalman Filter 25

2.3 Data Association 26

2.3.1 Mahalanobis Gating 26

2.3.2 Hungarian Algorithm 28

2.3.3 Nearest Neighbor 30

2.3.4 Global Nearest Neighbor 30

2.3.5 Joint Probabilistic Data Association 31

2.3.6 Multi-Hypothesis Tracking 32

2.4 Track Initiation Methods 34

2.4.1 M-of-N Initiation 35

2.4.2 Two-Point Differencing 35

2.4.3 Hough Transform 35

2.4.4 RANSAC . . . 36
2.4.5 Conclusion . . . 37
Chapter 3 RANSAC Incubation Framework . . . 39
3.1 Detection Buffer . . . 40
3.2 Inlier Consensus and Track Promotion . . . 41
3.3 Parameter Selection . . . 43
3.4 Conclusion . . . 45
Chapter 4 RANSAC-GNN Kalman Filter Engine . . . 47
4.1 State Model . . . 47
4.2 RANSAC Handover and Track Initialization . . . 48
4.3 Association . . . 48
4.4 Kalman Filter Update . . . 50
4.5 Track Management . . . 51
4.6 Computational Complexity . . . 51
4.7 Conclusion . . . 52
Chapter 5 RANSAC-TOMHT Engine . . . 53
5.1 Hypothesis Generation . . . 53
5.2 Hypothesis Pruning and Track Management . . . 54
5.3 Computational Complexity . . . 55
5.4 Simulator Architecture . . . 56
5.5 Clutter Model . . . 58
5.6 Target Motion Model . . . 58
5.7 Scenario Definitions . . . 59
5.7.1 Scenario 1 — Simple Intercept . . . 59
5.7.2 Scenario 2 — Horizontal and Angled . . . 60
5.7.3 Scenario 3 — Symmetric Crossing . . . 61
5.7.4 WLS Validation Scenario — Collision with Debris . . . 62

Chapter 6 Results and Performance Comparison 64
6.1 Performance Metrics 64
6.1.1 Multiple Object Tracking Accuracy (MOTA) 64
6.1.2 Multiple Object Tracking Precision (MOTP) 64
6.1.3 Root Mean Square Error (RMSE) 65
6.1.4 Time to Initialization (TTI) 65
6.1.5 Run Time 66
6.2 Scenario 1 Results — Simple Intercept 66
6.2.1 Nominal Performance 66
6.2.2 Clutter Sensitivity 70
6.2.3 Dropout Sensitivity 73
6.3 Scenario 2 Results — Horizontal and Angled 77
6.3.1 Nominal Performance 77
6.3.2 Clutter Sensitivity 80
6.3.3 Dropout Sensitivity 83
6.4 Scenario 3 Results — Symmetric Crossing 87
6.4.1 Nominal Performance 87
6.4.2 Clutter Sensitivity 91
6.4.3 Dropout Sensitivity 93
6.5 WLS Initialization Validation 96
6.5.1 Monte Carlo Study Design 97
6.5.2 Covariance Deflation Results 100
Chapter 7 Conclusions and Future Work 102
7.1 Conclusions 102
7.2 Future Work 103
References 106

List of Tables

5.1 Fixed engine parameters used across all scenarios and conditions. 57

6.1 Tracking Performance for Scenario 1 — Simple Intercept (Nominal Conditions: $\lambda = 0.10$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. . . . 69

6.2 Tracking Performance for Scenario 1 — Simple Intercept (Clutter Break Conditions: $\lambda = 0.30$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. Large negative MOTA scores correspond to conditions where the false track count significantly exceeded the total ground truth frame count. 73

6.3 Tracking Performance for Scenario 1 — Simple Intercept (Dropout Break Conditions: $\lambda = 0.10$, $p_d = 0.70$). Results are mean ± std across 1000 Monte Carlo trials. 76

6.4 Tracking Performance for Scenario 2 — Horizontal and Angled (Nominal Conditions: $\lambda = 0.10$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. 80

6.5 Tracking Performance for Scenario 2 — Horizontal and Angled (Clutter Break Conditions: $\lambda = 0.30$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. Large negative MOTA scores correspond to conditions where the false track count significantly exceeded the total ground truth frame count. 83

6.6 Tracking Performance for Scenario 2 — Horizontal and Angled (Dropout Break Conditions: $\lambda = 0.10$, $p_d = 0.70$). Results are mean ± std across 1000 Monte Carlo trials. 86

6.7 Tracking Performance for Scenario 3 — Symmetric Crossing (Nominal Conditions: $\lambda = 0.10$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. . . . 90

6.8 Tracking Performance for Scenario 3 — Symmetric Crossing (Clutter Break Conditions: $\lambda = 0.30$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. Large negative MOTA scores correspond to conditions where the false track count significantly exceeded the total ground truth frame count. 93
6.9 Tracking Performance for Scenario 3 — Symmetric Crossing (Dropout Break Conditions: $\lambda = 0.10$, $p_d = 0.70$). Results are mean ± std across 1000 Monte Carlo trials. 96
6.10 Mean inverse observation geometry matrix $(\mathbf{H}^T\mathbf{H})^{-1}$ averaged across all 13,000 promoted tracks. The Standard RANSAC $(\mathbf{H}^T\mathbf{H})^{-1}$ is systematically larger than truth, reflecting a modest inflation of the theoretical covariance from inlier set contamination. The Oracle WLS $(\mathbf{H}^T\mathbf{H})^{-1}$ matches truth exactly, confirming that the resulting overconfidence is driven by inlier set quality rather than any deficiency in the WLS formulation. 99
6.11 WLS initialization covariance deflation from 13,000 Monte Carlo track promotions across two evaluation windows of the collision with debris scenario. Deflation factor is the ratio of actual to theoretical covariance diagonal. A value greater than 1.0 indicates filter overconfidence. 101

List of Figures

2.1 WLS line fit to RANSAC inliers in range-time space. The fitted line provides the initial state estimate at track birth. Vertical residuals illustrate the distances minimized by the estimator. 24

2.2 Kalman filter predict-correct cycle [11]. 26

2.3 Euclidean versus Mahalanobis gating in range-time space. The Euclidean gate (left) admits detections within a fixed radius regardless of track dynamics. The Mahalanobis gate (right) is oriented by the innovation covariance, tilting to follow the target trajectory and admitting detection D which the Euclidean gate rejects. 28

2.4 Hough transform applied to two crossing lines. The accumulator space shows two distinct bright spots corresponding to the detected line parameters [22]. . . 36

2.5 RANSAC line fitting in the presence of outliers. Inliers (blue) support the fitted model while outliers (red) are excluded from parameter estimation. 37

3.1 Sequential RANSAC pool depletion. Starting from a cluttered detection pool, tracks are extracted iteratively. Each promoted track's inliers are removed from the pool before the next search, preventing confirmed structures from biasing subsequent hypotheses. 42

5.1 Scenario 1 — Simple Intercept. A horizontal target at 250m is struck by an angled closing target at $t = 60$s, spawning six debris children. 60

5.2 Scenario 2 — Horizontal and Angled. Two horizontal flyby targets at 380m and 120m remain active throughout while an angled target strikes a third horizontal target at 250m at $t = 60$s, spawning six debris children. 61

5.3 Scenario 3 — Symmetric Crossing. Two horizontal flyby targets bracket the scene while two targets approach symmetrically from opposite ends of the range extent, crossing at 250m at $t = 60$s and spawning six debris children. The symmetric crossing geometry produces maximum association ambiguity at the collision point. 62

5.4 Ground truth targets (green) and clutter detections (black) for the two Monte Carlo evaluation windows. Window 1 shows the pre-collision regime with two targets. Window 2 shows the post-collision debris fan with ten simultaneously active targets. 63

6.1 Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-GNNKF engine. 67

6.2 Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-GNNKF engine. 68

6.3 Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-TOMHT engine. 68

6.4 Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-TOMHT engine. 69

6.5 Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-GNNKF engine. 70

6.6 Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-GNNKF engine. 71

6.7 Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-TOMHT engine. 72

6.8 Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-TOMHT engine. 72

6.9 Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-GNNKF engine. 74

6.10 Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-GNNKF engine. 75
6.11 Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-TOMHT engine. 75
6.12 Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-TOMHT engine. 76
6.13 Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-GNNKF engine. 78
6.14 Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-GNNKF engine. 78
6.15 Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-TOMHT engine. 79
6.16 Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-TOMHT engine. 80
6.17 Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-GNNKF engine. 81
6.18 Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-GNNKF engine. 82
6.19 Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-TOMHT engine. 82
6.20 Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-TOMHT engine. 83
6.21 Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-GNNKF engine. 84
6.22 Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-GNNKF engine. 85

6.23 Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-TOMHT engine. 85
6.24 Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-TOMHT engine. 86
6.25 Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-GNNKF engine. 88
6.26 Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-GNNKF engine. 89
6.27 Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-TOMHT engine. 89
6.28 Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-TOMHT engine. 90
6.29 Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-GNNKF engine. 91
6.30 Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-GNNKF engine. 92
6.31 Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-TOMHT engine. 92
6.32 Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-TOMHT engine. 93
6.33 Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-GNNKF engine. 94
6.34 Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-GNNKF engine. 95
6.35 Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-TOMHT engine. 95

6.36 Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-TOMHT engine. 96
6.37 RANSAC line fits (red) for a representative Monte Carlo run in both evaluation windows. Window 1 shows two clean tracks in heavy clutter. Window 2 shows the RANSAC incubator successfully recovering multiple debris trajectories from the post-collision detection pool. 98
6.38 Theoretical versus actual 99% confidence covariance ellipses for WLS and Oracle WLS initialization across 13,000 Monte Carlo track promotions. The theoretical ellipse lies inside the actual ellipse in the WLS panel, reflecting filter overconfidence due to clutter contamination of the RANSAC inlier set. The Oracle WLS panel shows near-perfect calibration when true inliers are known. 100

List of Abbreviations

GNN Global Nearest Neighbor

GNNKF Global Nearest Neighbor Kalman Filter

GOSPA Generalized Optimal Sub-Pattern Assignment

JPDA Joint Probabilistic Data Association

KF Kalman Filter

LLR Log-Likelihood Ratio

MHT Multi-Hypothesis Tracking

MOMHT Measurement-Oriented Multi-Hypothesis Tracking

MOTA Multiple Object Tracking Accuracy

MOTP Multiple Object Tracking Precision

NN Nearest Neighbor

OSPA Optimal Sub-Pattern Assignment

PDA Probabilistic Data Association

RANSAC Random Sample Consensus

R-RANSAC Recursive RANSAC

RMSE Root Mean Square Error

TOMHT Track-Oriented Multi-Hypothesis Tracking

TPD Two-Point Differencing

TTI Time to Initialization

WLS Weighted Least Squares

Chapter 1

Introduction

In dense clutter environments, radar range detection pools are dominated by false alarms that substantially outnumber true target returns, making reliable track initiation a fundamental challenge. This research addresses the problem of track initiation in these highly cluttered environments, where sensors produce a single range measurement per scan with no angular information. Classical track initiation methods struggle in environments where false alarm rates approach or exceed target detection rates, motivating a more robust approach to accumulating and evaluating detections prior to track confirmation. The tracking engines proposed here build upon the foundational recursive estimation framework of R-RANSAC [1], classical multi-hypothesis tracking theory [2], and the combination of Kalman filtering with Hungarian algorithm data association demonstrated in the computer vision domain by Bewley et al. [3]. These techniques are adapted to the degraded radar tracking problem. Although developed in a radar context, the proposed framework is applicable to any domain where targets produce approximately linear trajectories in a detection space including the computer vision applications from which some of its components are drawn.

1.1 Problem Statement

Range-time environments present a fundamental tracking challenge. Targets trace paths through the range-time plane as a function of their dynamics, while clutter returns are distributed across all range bins at every scan, making it difficult to distinguish genuine target detections from false alarms. As clutter density increases, classical track initiation methods based on M-of-N confirmation thresholds become unreliable, since random clutter detections falling within the confirmation gate on M out of N consecutive scans are sufficient to promote a false track [1]. This problem is compounded in post-collision debris scenarios, where multiple targets appear simultaneously at similar ranges and must be initiated quickly before they diverge.

Range-based tracking has a long history in radar applications [4] and remains an active area of research in modern sensor systems [5], [6]. However, the specific challenge of robust multi-target track initiation in heavily cluttered 1D range-time environments has received limited attention in the literature. This thesis addresses that gap by proposing a RANSAC-based track incubation framework that exploits the linear geometry of constant velocity targets in the range-time plane to discriminate genuine tracks from clutter before committing to a confirmed track.

1.2 Proposed Approaches

This thesis proposes a RANSAC-based track incubation framework as an alternative to classical M-of-N initiation for multi-target tracking in cluttered range-time environments. Rather than confirming a track based on detection count alone, the incubator searches the sliding window detection pool for groups of detections that are consistent with a linear trajectory in the range-time plane, similar to the recursive RANSAC framework of Niedfeldt and Beard [1] adapted to this specific range-time tracking problem. A candidate track is promoted only when a sufficient number of detections support a common line hypothesis under random sampling, providing geometric validation that M-of-N initiation cannot offer. Confirmed tracks are initialized using a Weighted Least Squares fit over the inlier detection set producing an initial state estimate that includes range, range-rate, and covariance.

The incubation framework is evaluated in combination with two tracking engines. The first pairs the RANSAC incubator with a Global Nearest Neighbor Kalman Filter [3], providing a computationally efficient baseline that makes hard data associations at each scan using the Hungarian algorithm. The second pairs the incubator with a Track-Oriented Multiple Hypothesis Tracker [2], which maintains multiple association hypotheses across several scans before committing to the most likely track. Both engines are benchmarked against M-of-N initialized counterparts across three simulated collision and debris scenarios under nominal, heavy clutter, and reduced detection probability conditions.

1.3 Contributions

This thesis makes the following contributions to the track initiation and tracking literature for cluttered range-only radar environments:

1. **RANSAC-incubated Global Nearest Neighbor Kalman Filter.** A novel track initiation architecture is presented in which geometrically validated RANSAC inlier sets [7] seed a Kalman filter using Hungarian algorithm data association [3]. The global optimal assignment is critical in high-density scenarios; greedy nearest-neighbor association starves competing tracks of detections when other tracks are close in proximity. Hungarian assignment resolves this by jointly optimizing across all track-detection pairs, minimizing total assignment cost for that frame. To the author's knowledge, no prior work combines RANSAC incubation with Hungarian-assigned GNNKF.

2. **RANSAC-incubated Track-Oriented Multi-Hypothesis Tracker.** A novel architecture is presented in which RANSAC [7] serves as the track incubation stage for a TOMHT engine [2]. Candidate trajectories are promoted only upon meeting a minimum inlier consensus threshold, ensuring that only geometrically validated tracks enter the hypothesis generation stage. By bounding the number of tracks that generate hypotheses at birth, the RANSAC incubation stage reduces the hypothesis space that must be managed at each scan, complementing standard pruning strategies. To the author's knowledge, no prior work has employed RANSAC as an incubator for TOMHT.

3. **WLS-based track initialization from RANSAC inlier geometry.** Upon RANSAC hypothesis confirmation, a Weighted Least Squares estimator is applied to the validated inlier set to derive the initial state estimate and covariance at track birth. The WLS formulation, following Li and He [8], produces an initial covariance $\mathbf{P}_{kin}$ that reflects the actual uncertainty in the batch estimate given the sensor noise, providing a statistically principled filter initialization without requiring prior noise characterization or heuristic covariance tuning.

4. **Sequential greedy pool depletion.** A greedy sequential depletion strategy is proposed in which confirmed inliers are removed from the residual detection pool before each successive RANSAC search, enabling multiple simultaneous track initiations from a single scan window without mutual interference.

5. **Simulation study in a high-clutter debris environment.** Performance of both engines is evaluated in a purpose-built range-only simulator featuring crossing and colliding targets with debris children under Poisson clutter, characterizing initiation TTI, tracking accuracy, and sensitivity to clutter density and detection probability across operationally relevant scenarios. Performance is additionally benchmarked against non-RANSAC-initialized counterparts to isolate the contribution of the incubation stage.

Chapter 2

Background and Prior Work

A range-only radar sensor produces a single range profile per scan. This creates a vector of detection reports at discrete range bins with no angular information. Each scan represents a snapshot of detected returns across all range bins at one instant in time. As successive scans are collected, they form a two-dimensional range-time record which can be represented by a horizontal axis corresponding to scan time and the vertical axis to range. Assuming the relative velocity between the radar and the target is constant, the target traces a straight line with slope proportional to the target's range-rate and intercept determined by its range at the start of the observation interval. This linear geometry is exploited by the RANSAC incubation framework developed in Chapter 3.

In the degraded environments considered in this thesis, the range profile is contaminated by two principal sources of interference. False detections arise from environmental clutter and noise. This produces a dense background of returns with no coherent geometric structure is present [9]. Another challenge is a low and variable probability of detection p_d. A target may be absent from several consecutive scans producing gaps in what would otherwise be a continuous linear trajectory. The combination of high false alarm rate and low p_d defines the degraded domain addressed in this thesis, and represents the fundamental challenge for any track initiation method operating in this space. Classical methods that evaluate detections on a scan-by-scan basis are poorly suited to this problem motivating the multi-scan accumulation strategy developed in Chapter 3.

Estimating a target's kinematic state from these degraded detections is therefore a two part problem. First, once a batch of validated detections has been accumulated for a target, an initial state estimate and covariance must be extracted from that batch before tracking can begin. Second, once a track is confirmed and new detections continue to arrive scan by scan, that initial estimate must be propagated and updated recursively without reprocessing the full measurement history at every step. Weighted Least Squares (WLS), developed in the following section, addresses the first problem, providing a statistically principled initial state and covariance from a batch of detections. The Kalman filter, developed in Section 2.2, addresses the second, propagating that initial estimate forward scan by scan as new measurements arrive. Together these two tools bridge the gap between the RANSAC incubation framework of Chapter 3, which collects a batch of detections as belonging to a common target, and the recursive tracking engines of Chapters 4 and 5, which maintain that track after initialization.

2.1 Weighted Least Squares

Weighted Least Squares (WLS) is a batch estimation method that fits a linear model to a set of observations by minimizing a weighted sum of squared residuals. Unlike ordinary least squares, which treats all observations equally, WLS assigns a weight to each measurement that reflects its reliability. This makes WLS particularly well suited to sensor fusion and filter initialization problems where measurement quality is known or can be characterized [8]. Given a batch of N measurements $\mathbf{y} \in \mathbb{R}^N$ related to an unknown parameter vector θ by the linear model:

$$\mathbf{y} = \mathbf{H}\theta + \epsilon \tag{2.1}$$

where $\mathbf{H} \in \mathbb{R}^{N \times p}$ is the observation matrix and ϵ is a noise vector with covariance $\sigma_r^2 \mathbf{I}$, the WLS estimator minimizes the weighted residual:

$$\hat{\theta} = \arg\min_{\theta} (\mathbf{y} - \mathbf{H}\theta)^T \mathbf{W} (\mathbf{y} - \mathbf{H}\theta) \tag{2.2}$$

where $\mathbf{W}$ is a positive definite weighting matrix. The closed-form solution is:

$$\hat{\theta} = \left(\mathbf{H}^T \mathbf{W} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{W} \mathbf{y} \tag{2.3}$$

with associated estimation covariance:

$$\mathbf{P} = \left(\mathbf{H}^T \mathbf{W} \mathbf{H}\right)^{-1} \tag{2.4}$$

When $\mathbf{W} = \sigma_r^{-2}\mathbf{I}$, the WLS estimator reduces to ordinary least squares and the covariance $\mathbf{P}$ reflects the actual uncertainty in the batch estimate given the sensor noise. WLS formulation naturally accommodates non-uniform weights for varying measurement quality. This work uses uniform weights, since the simulated detections all carry the same noise level.

Figure 2.1 illustrates the WLS line fit applied to a set of RANSAC inliers in range-time space. The vertical residuals show the distances being minimized by the estimator. The fitted line provides both the initial range and range rate estimates projected to the current frame.

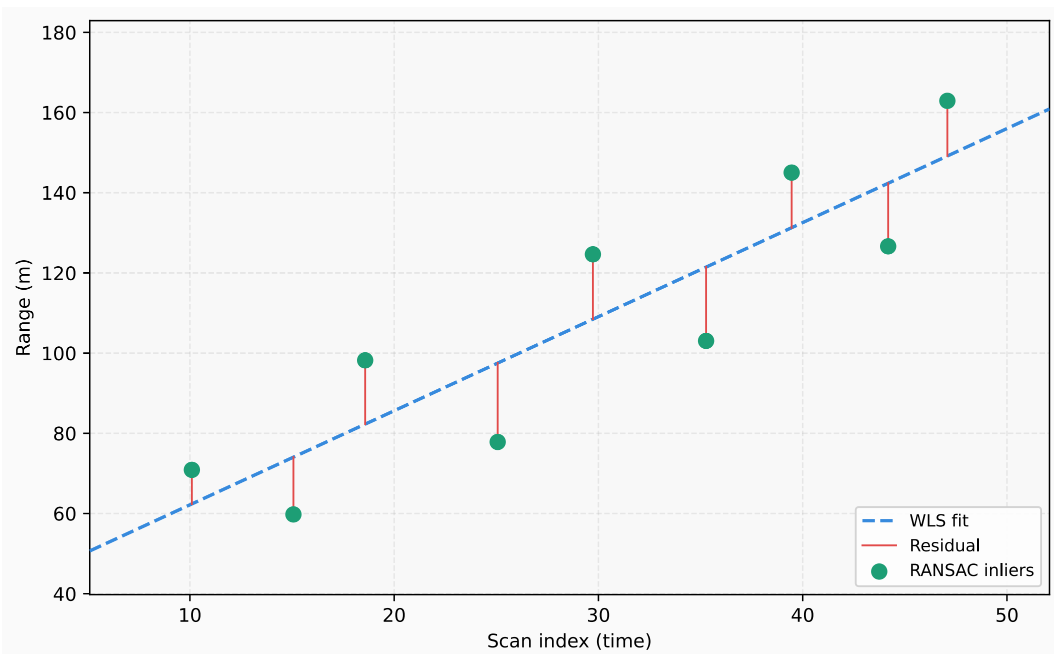

Figure 2.1: WLS line fit to RANSAC inliers in range-time space. The fitted line provides the initial state estimate at track birth. Vertical residuals illustrate the distances minimized by the estimator.

The relevance of WLS to filter initialization is substantial. Li and He [8] demonstrated that initializing a Kalman filter using a WLS batch estimator produces a superior initial state estimate and covariance compared to simpler approaches such as two-point differencing. Crucially, the WLS covariance $\mathbf{P}$ provides a statistically principled initialization of the filter covariance that reflects the uncertainty in the batch estimate. This property is directly exploited in the RANSAC incubation framework developed in Chapter 3, where WLS is applied to the validated inlier set to derive both the initial state estimate and initial covariance at track birth.

2.2 Kalman Filter

The Kalman filter, introduced by Kalman [10], is a recursive minimum mean square error estimator for linear systems subject to Gaussian process and measurement noise. At each scan, the filter maintains a state estimate and associated covariance that are propagated forward in time using the system dynamics model and corrected upon receiving a new measurement. The resulting predicted state estimate and its covariance are the quantities used to generate predicted measurements and innovation covariances for gating and data association, as described in Section 2.3.1. The two-step predict and correct cycle is illustrated in Figure 2.2, and the full discrete-time Kalman filter equations, process noise model, and initialization procedure used in both tracking engines are developed in detail in Chapters 4 and 5.

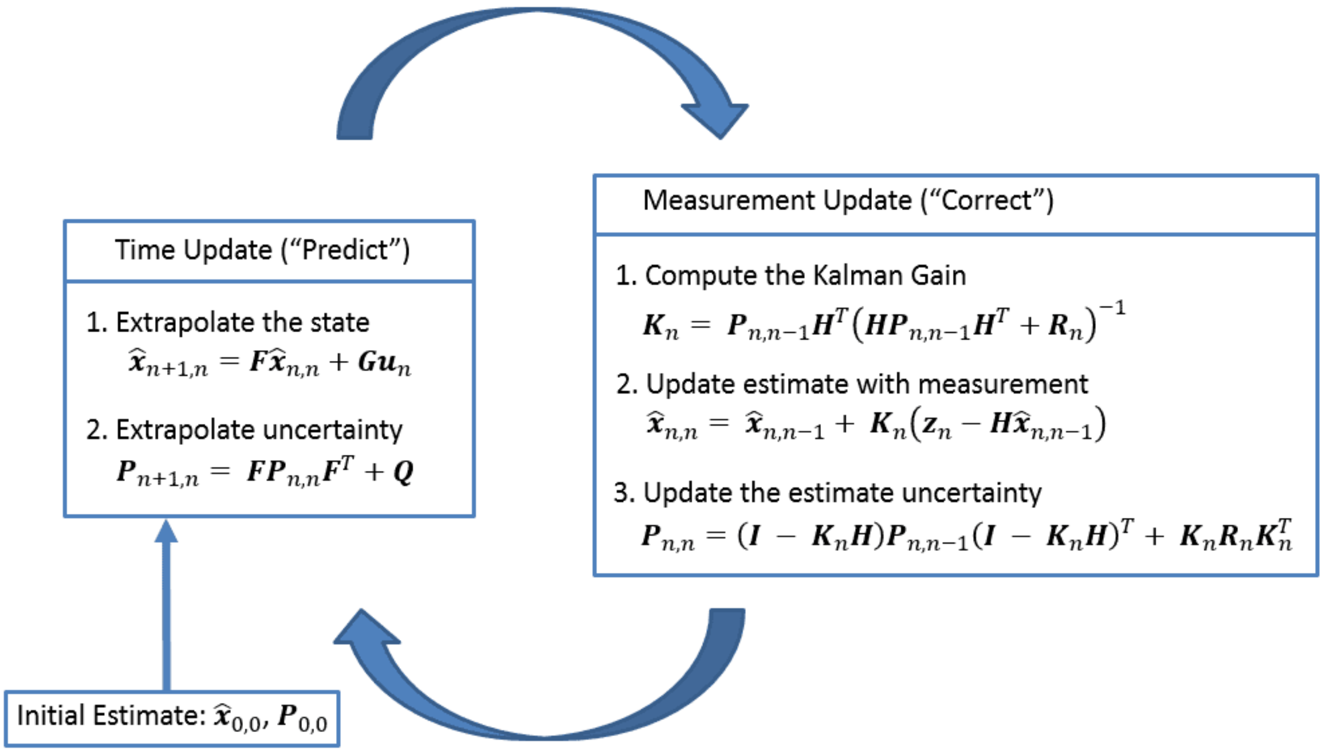

Figure 2.2: Kalman filter predict-correct cycle [11].

2.3 Data Association

Data association in the context of this thesis is the process of determining which detections, if any, belong to pre-established tracks. In multi-target tracking scenarios, this requires both a mechanism for rejecting improbable pairings and an assignment strategy for resolving the remaining candidates. The following subsections describe the gating procedure used to restrict candidate associations, followed by three data association methods of increasing sophistication that appear in the tracking literature and motivate the design choices made in this thesis.

2.3.1 Mahalanobis Gating

Prior to solving the assignment problem, improbable track-detection pairings are eliminated through a gating procedure. Gating reduces both computational cost and assignment risk by restricting potential associations to detections that fall within a statistically justified neighborhood of each track's predicted state [9], [12]. The Mahalanobis distance between track i and detection j is defined as:

$$d_{i,j}^2 = \tilde{y}_{i,j}^T S_i^{-1} \tilde{y}_{i,j} \tag{2.5}$$

where $\tilde{y}_{i,j} = z_j - \mathbf{C}\hat{\mathbf{x}}_{i,k|k-1}$ is the innovation and $S_i = \mathbf{C}\mathbf{P}_{i,k|k-1}\mathbf{C}^T + R$ is the innovation covariance, with z_j the range measurement of detection j, $\hat{\mathbf{x}}_{i,k|k-1}$ the predicted state of track i at scan k, $\mathbf{C}$ the observation matrix, and R the measurement noise variance. Under the assumption that the measurement noise is Gaussian, $d_{i,j}^2$ follows a chi-squared distribution with degrees of freedom equal to the measurement dimension. A detection is admitted to the gate for track i only if:

$$d_{i,j}^2 \leq \chi_{n_z,\alpha}^2 \tag{2.6}$$

where n_z is the measurement dimension and $\chi_{n_z,\alpha}^2$ is the chi-squared threshold corresponding to confidence level α. Although the sensor produces a scalar range measurement, the gate is evaluated in the joint range-time innovation space, giving $n_z = 2$. The 99% confidence gate threshold is therefore $\chi_{2,0.99}^2 = 9.21$ [13]. Track-detection pairs outside this gate are considered extremely unlikely and excluded from the association step.

Figure 2.3 illustrates the difference between Euclidean and Mahalanobis gating in range-time space. The Euclidean gate is a fixed-radius circle that treats range and time errors as equally likely in all directions, admitting detections based solely on raw distance regardless of track dynamics. The Mahalanobis gate is an ellipse oriented by the innovation covariance S_i, which accounts for the target's range-rate as a correlation between range and time errors, tilting the gate to follow the track's trajectory. In the figure detection D lies along the trajectory axis and is correctly admitted by the Mahalanobis gate despite falling outside the Euclidean gate. Detections A, B, and C are rejected as inconsistent with the track by the Mahalanobis gate. This demonstrates that gate shape, not just size, helps determine which detections are plausible associations.

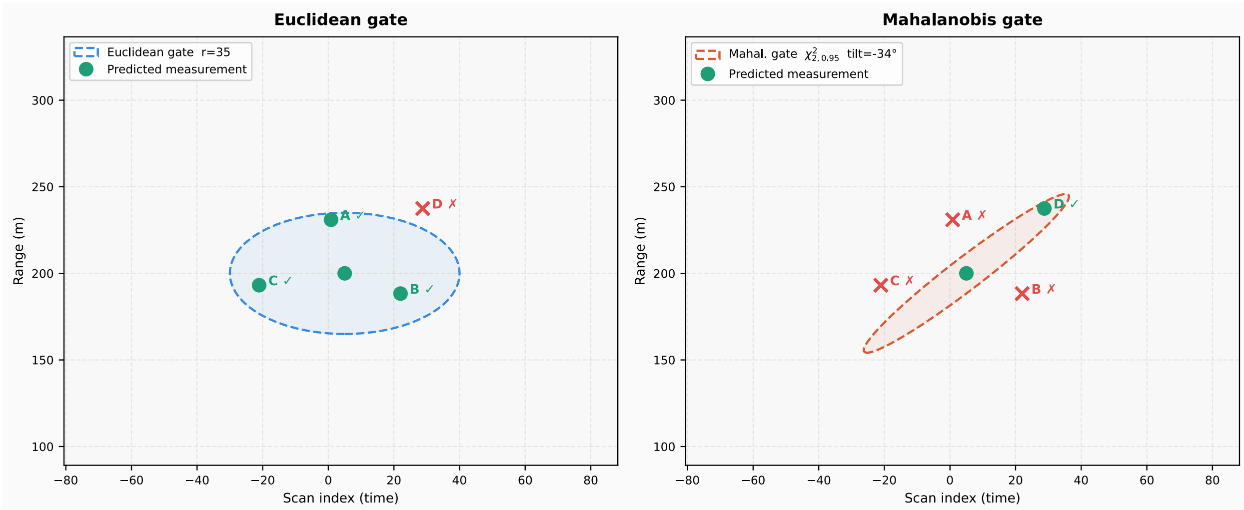

Figure 2.3: Euclidean versus Mahalanobis gating in range-time space. The Euclidean gate (left) admits detections within a fixed radius regardless of track dynamics. The Mahalanobis gate (right) is oriented by the innovation covariance, tilting to follow the target trajectory and admitting detection D which the Euclidean gate rejects.

2.3.2 Hungarian Algorithm

The Hungarian algorithm, introduced by Kuhn [14], solves the linear assignment problem optimally, finding the one-to-one assignment between tracks and detections that minimizes total cost. Given a cost matrix where entry $\Omega_{i,j}$ represents the cost of assigning detection j to track i, the algorithm finds:

$$\min_{\chi} \sum_i \sum_j \Omega_{i,j} \chi_{i,j} \tag{2.7}$$

where $\chi_{i,j} \in \{0,1\}$ enforces a hard one-to-one assignment. Gating is incorporated directly into the cost matrix by assigning infinite cost to any track-detection pair that fails the Mahalanobis gate described in Section 2.3.1. This prevents geometrically improbable assignments from being selected. In this work the assignment is solved using an implementation of the Hungarian algorithm [15], which produces the globally optimal solution to the same problem.

To illustrate how this works, consider a tracking scenario with three active tracks (T_1, T_2, T_3) and three new radar detections (D_1, D_2, D_3). Let the maximum allowable cost be set to $\gamma = 3.0$. The initial cost matrix $_{init}$ is:

$$_{init} = \begin{pmatrix} 1.5 & 4.2 & 6.0 \\ 5.1 & 2.2 & 1.1 \\ 3.8 & 7.0 & 4.5 \end{pmatrix} \tag{2.8}$$

Applying the validation gate, any pairing where $\Omega_{i,j} > \gamma$ fails the innovation threshold criteria and is overwritten with an infinite cost (∞). For example, entry $\Omega_{1,1} = 1.5 \leq 3.0$ survives the gate, while $\Omega_{1,2} = 4.2 > 3.0$ is rejected. Because every candidate pairing for track T_3 exceeds the threshold ($\Omega_{3,1}, \Omega_{3,2}, \Omega_{3,3} > 3.0$), its entire row is gated out. This yields the gated cost matrix :

$$= \begin{pmatrix} 1.5 & \infty & \infty \\ \infty & 2.2 & 1.1 \\ \infty & \infty & \infty \end{pmatrix} \tag{2.9}$$

Solving this minimization problem via the Hungarian algorithm results in the optimal binary assignment matrix :

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.10}$$

The unique optimal pairings are $T_1 \rightarrow D_1$ and $T_2 \rightarrow D_3$, yielding a total minimized global cost of:

$$\text{Total Cost} = \sum_i \sum_j \Omega_{i,j} \chi_{i,j} = 1.5 + 1.1 = 2.6 \tag{2.11}$$

This scenario demonstrates the key operational features of the gated assignment framework:

- **Track Coasting:** Because the third row contains exclusively ∞ entries, track T_3 receives no assignment ($\sum_j \chi_{3,j} = 0$). The tracking architecture automatically coasts T_3 using its kinematic state prediction model.

- **Clutter Rejection:** Detection D_2 remains completely unassigned ($\sum_i \chi_{i,2} = 0$). It is rejected by the active tracking loop (and routed back to the track initialization pool).

2.3.3 Nearest Neighbor

The Nearest Neighbor (NN) algorithm is the simplest approach to data association. At each scan, each track is independently assigned to the detection within its gate that minimizes the distance between the predicted position and the measurement [12], [16]. No joint optimization is performed. Each assignment decision is made greedily and independently of all other tracks.

While computationally efficient, NN is suboptimal in multi-target scenarios. When two tracks compete for the same detection, NN awards the detection to whichever track has the smaller Mahalanobis distance, potentially leaving the other track unupdated regardless of the global consequence of that decision. In high density environments this greedy behavior can cause track starvation and premature termination of valid tracks. Global Nearest Neighbor association, described in the following subsection, addresses this limitation by solving the assignment problem jointly across all tracks and detections.

2.3.4 Global Nearest Neighbor

Global Nearest Neighbor (GNN) extends the Nearest Neighbor (NN) approach by solving the assignment problem globally rather than greedily. Rather than assigning each track independently to its closest detection, GNN constructs a gated cost matrix as described in Sections 2.3.1 and 2.3.2 and applies the Hungarian algorithm to find the one-to-one assignment that minimizes total cost across all track-detection pairs simultaneously per frame [9], [14], [17].

This global optimality is particularly important in high-density scenarios where tracks are proximate. A greedy NN approach may award a shared detection to the wrong track, starving another track of its update and potentially causing premature termination. GNN resolves such conflicts by jointly optimizing across all pairs ensuring that all tracks compete on equal footing with established ones.

The known limitation of GNN is its hard commitment to a single assignment at each scan. When two tracks compete for the same detection in an ambiguous scenario, GNN makes an irrevocable decision based only on current scan information. This motivates the softer and deferred decision strategies of Joint Probabilistic Data Association (JPDA) and Multi-Hypothesis Theory (MHT) described in the following subsections.

2.3.5 Joint Probabilistic Data Association

Joint Probabilistic Data Association (JPDA) was introduced by Fortmann et al. [12] as an extension of the single-target Probabilistic Data Association (PDA) filter of Bar-Shalom and Tse [9]. It addresses the hard-commitment limitation of NN and GNN by computing the posterior probability that each detection originated from each track jointly across all targets in a cluster. Rather than selecting a single detection as the update source, JPDA computes a weighted average of all gated detections where the weights reflect the posterior probability that each detection originated from the track being updated [9], [12]. This produces a soft update that accounts for association uncertainty without committing to a single hypothesis.

JPDA offers improved robustness over GNN in ambiguous multi-target scenarios, particularly when targets are closely spaced and share common detections. However, its computational complexity grows with the number of targets and detections within a cluster making JPDA impractical in dense environments. A known failure mode is track coalescence, where two closely spaced tracks begin pulling toward the same weighted average of shared detections causing their state estimates to converge. Recent work by Rezatofighi et al. [18] addressed the complexity limitation by reformulating the joint association probability computation as an integer linear program and approximating it using only the m-best solutions, making JPDA more viable in high-density applications. JPDA is not employed in the tracking engines developed in this thesis, but is identified as a direction for future work in Chapter 7.2.

2.3.6 Multi-Hypothesis Tracking

Multi-Hypothesis Tracking (MHT), introduced by Reid [17], addresses the fundamental limitation shared by NN, GNN, and JPDA: the requirement to resolve data association ambiguity at the scan in which it arises. Rather than committing to a single assignment or computing a weighted average of competing hypotheses at each scan, MHT explicitly iterates through multiple association hypotheses and defers resolution until sufficient evidence has accumulated across scans to decide on the best branch. This deferred decision framework is particularly well suited to cluttered environments where momentary ambiguity is common but resolves naturally over a short observation horizon.

Reid's original formulation, now referred to as measurement-oriented MHT (MOMHT), maintains a global hypothesis tree in which each node represents a complete assignment of all detections across all scans to all tracks [17]. At each scan, every existing hypothesis spawns a family of children, one for each way the new detections could be explained: assigned to an existing track, treated as a new target, or dismissed as a false alarm. Each hypothesis is scored by its joint probability, which accumulates evidence across scans. The optimal assignment is recovered by identifying the highest scoring branch of the tree. While theoretically optimal, in practice MOMHT becomes unmanageable quickly. Each new scan multiplies the number of active hypotheses by the number of gated detections per track. With multiple targets in clutter the total hypothesis count grows unwieldy within just a few scans [2], [17].

The track-oriented formulation of MHT (TOMHT), developed and popularized by Blackman [2], restructures the hypothesis representation to make the framework computationally practical. Rather than maintaining a single global hypothesis tree over all targets simultaneously, TOMHT maintains a separate hypothesis tree for each individual track. Each track maintains a set of hypotheses $\{\mathcal{H}_{i,l}\}$, where each hypothesis $\mathcal{H}_{i,l}$ records the full association history of track i from birth through the current scan. When a new scan arrives, every surviving hypothesis branches once for each detection within its gate, plus one missed detection branch, accounting for the possibility that the target was simply not detected that scan. This process repeats each scan, so the tree grows with each new set of detections. After N scans a single hypothesis will have branched into a large family of descendants, each representing a different association history, which is precisely why pruning is required to keep the hypothesis count manageable. Each branch carries a Kalman filter propagated under its specific association history and is scored by the accumulated log-likelihood of its measurement sequence. This per-track structure decouples the branching problem across tracks, reducing the burden from global to local and enabling the framework to scale to multi-target scenarios [2], [19]. Frank et al. [20] provide a graphical model reformulation of TOMHT that makes the probabilistic structure of the per track hypotheses explicit, offering a useful framework for reasoning about track-to-measurement relationships. The score of each hypothesis is updated recursively as [2]:

$$L(k) = L(k-1) + \Delta L(k) \tag{2.12}$$

where the score increment is:

$$\Delta L(k) = \begin{cases} \ln(1-\hat{P}_D) & \text{no update on scan } k \\ \Delta L_u(k) & \text{track update on scan } k \end{cases} \tag{2.13}$$

and the update increment $\Delta L_u(k)$ is a function of the innovation, its covariance, the probability of detection $\hat{P}_D$, and the clutter density λ [2], [19]:

$$\Delta L_u(k) = \ln(\hat{P}_D) + \ln \mathcal{N}(\tilde{y}_{i,k};\, 0,\, S_{i,k}) - \ln(\lambda) \tag{2.14}$$

where $\mathcal{N}(\tilde{y}_{i,k};\, 0,\, S_{i,k}) = \frac{1}{\sqrt{2\pi S_{i,k}}} \exp\left(-\frac{\tilde{y}_{i,k}^2}{2S_{i,k}}\right)$, with $\tilde{y}_{i,k} = z_j - \mathbf{C}\hat{\mathbf{x}}_{i,k|k-1}$ and $S_{i,k} = \mathbf{C}\mathbf{P}_{i,k|k-1}\mathbf{C}^T + R$ as defined in Section 2.3.1. Each hypothesis maintains its own Kalman filter state and covariance, and is updated according to its specific association sequence.

Without pruning, the number of hypotheses per track grows without bound as the tree expands across scans. Two strategies are traditionally used to keep the hypothesis set tractable.

The first is score thresholding: hypotheses whose accumulated log-likelihood falls more than a fixed margin Δ_Λ below the best hypothesis for the same track are pruned, on the grounds that their probability of being the correct association sequence has become negligible [2], [19]:

$$\text{Prune } \mathcal{H}_{i,l} \text{ if } L^* - L(\mathcal{H}_{i,l}) > \Delta_\Lambda \tag{2.15}$$

where $\mathcal{H}_{i,l}$ denotes the l-th hypothesis maintained for track i, and $L^* = \max_l L(\mathcal{H}_{i,l})$ is the score of the best surviving hypothesis for track i. The second strategy is N-scan pruning [2], [17]. The tree branches at every scan, accumulating hypotheses across a window of N scans. At the end of the window, the highest-scoring hypothesis is selected and all competing branches are pruned. Branching then resumes from the surviving hypothesis at the next scan, with the tree depth reset to one. This bounds the hypothesis tree to a fixed depth of N scans at all times, regardless of detection density. The cost is that the commitment is irrevocable, if the best hypothesis at scan N turns out to be wrong, those competing branches are gone. The parameter N therefore controls a direct tradeoff. Larger N gives more evidence before committing at the cost of a larger hypothesis tree. Smaller N keeps computation cheap but risks premature commitment in difficult scenarios.

The TOMHT engine developed in Chapter 5 adopts this per-track branching and pruning structure directly, with the RANSAC incubation stage of Chapter 3 serving as the track birth mechanism. Because RANSAC geometrically validates candidate tracks before they enter the hypothesis generation stage, the hypothesis space that must be managed at each scan is bounded from birth, complementing the score thresholding and N-scan pruning strategies described here.

2.4 Track Initiation Methods

A range of methods exist for creating a new track from a pool of detections, each striking a different balance between computational cost and robustness to clutter. The following sections review four such methods, M-of-N, two-point differencing, the Hough transform, and RANSAC. Each method is evaluated for its suitability to the degraded range-only tracking problem described earlier in this chapter, motivating the RANSAC-based incubation framework developed in Chapter 3.

2.4.1 M-of-N Initiation

M-of-N initiation is a widely employed track initiation strategy in which a potential track is confirmed when M detections are observed within a sliding window of N consecutive frames [13], [21]. While effective in low-clutter environments with high probability of detection, this approach exhibits two characteristic failure modes as clutter density increases: false track initiation when the confirmation threshold M is set too low, and missed track initiation when M is set too high. In practice, selecting an appropriate threshold becomes impractical when false alarm rates approach or exceed the target detection rate [1].

2.4.2 Two-Point Differencing

Two-point differencing (TPD) is a classical track initiation technique in which two sequential position measurements are used to estimate the initial state of a target. Given measurements z_{k-1} and z_k separated by a time interval T, the initial range and range-rate estimates are formed as:

$$\hat{x} = \begin{bmatrix} z_k \\ \frac{z_k - z_{k-1}}{T} \end{bmatrix} \tag{2.16}$$

While computationally simple, TPD is sensitive to false alarms. If either seed point is a false detection, the resulting state estimate is corrupted, actively degrading filter initialization. In dense clutter environments this is a significant limitation, as there is no mechanism to distinguish a valid target detection from a false alarm [8].

2.4.3 Hough Transform

The Hough transform, formalized by Duda and Hart [22], detects linear structures by transforming each detection point into a sinusoidal curve in a discretized parameter space. Points lying on the same line in the detection plane produce curves that intersect at a common point in parameter space, allowing line detection through a voting accumulator procedure, as illustrated in Figure 2.4.

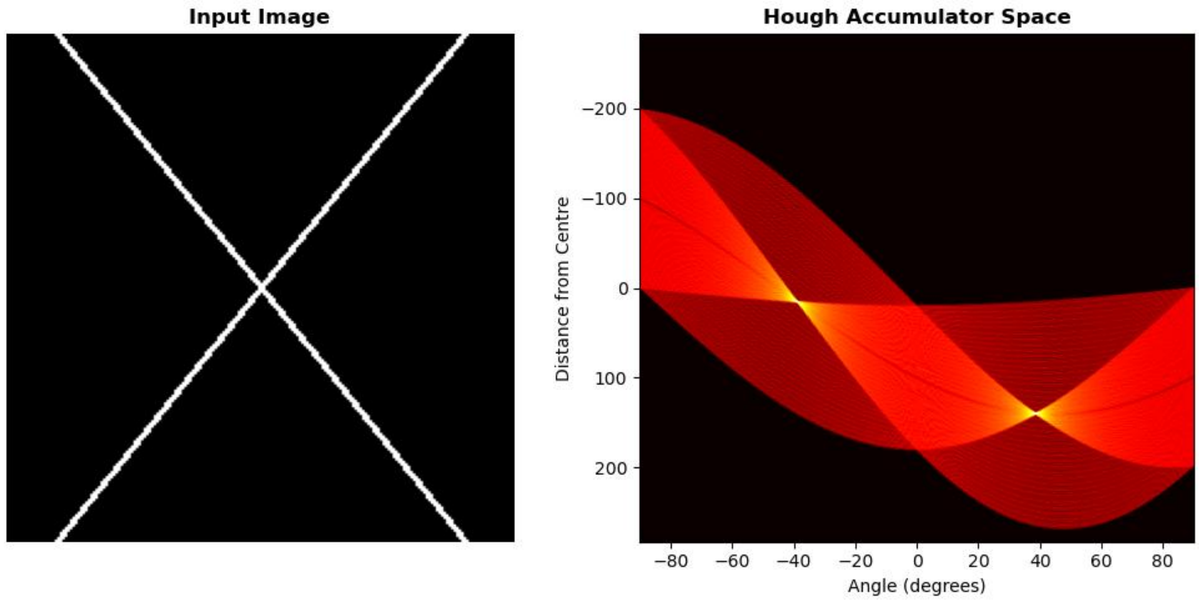

Figure 2.4: Hough transform applied to two crossing lines. The accumulator space shows two distinct bright spots corresponding to the detected line parameters [22].

While computationally efficient for isolated line detection, the Hough transform exhibits resolution limitations that are problematic in multi-target tracking scenarios. As noted by Duda and Hart [22], the method depends on the choice of bin size, bins that are too coarse can merge distinct target trajectories, while bins that are too fine increase computational cost. This tradeoff becomes particularly problematic when multiple targets occupy similar velocity and range bins simultaneously.

2.4.4 RANSAC

Random Sample Consensus (RANSAC), introduced by Fischler and Bolles [7], is a model fitting algorithm designed to estimate parameters from a dataset containing a significant proportion of outliers. The algorithm operates by repeatedly sampling a minimal subset of points, and fits a model to that subset. The algorithm then evaluates the number of remaining points consistent with the fitted model within a predefined tolerance, noting inliers. If the inlier count exceeds a minimum threshold, the hypothesis is considered a strong candidate and is extracted. This process repeats for a fixed number of iterations, retaining the hypothesis with the greatest inlier support [7]. Figure 2.5 illustrates a two parameter case (linear).

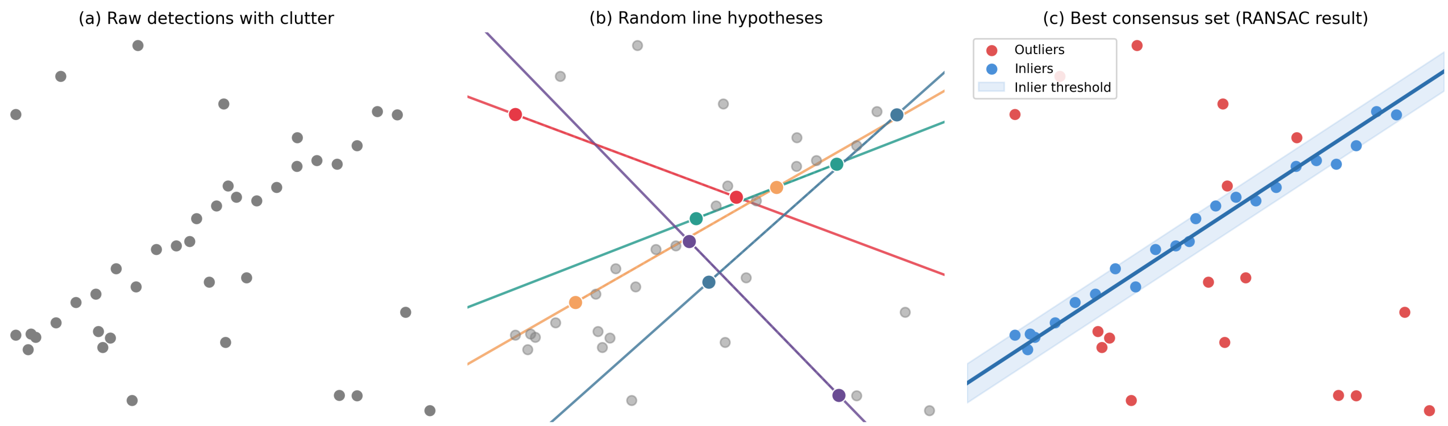

Figure 2.5: RANSAC line fitting in the presence of outliers. Inliers (blue) support the fitted model while outliers (red) are excluded from parameter estimation.

Under a constant velocity motion model, target returns in a range-only radar trace approximately linear trajectories in the range-time plane [4]. RANSAC is particularly well suited to this problem for two reasons. First, random clutter returns are geometrically unlikely to produce sufficient collinear support to satisfy the inlier threshold, providing natural suppression of false initiations without perfect parameter tuning. Second, unlike the Hough transform which discretizes parameter space and suffers from resolution limitations when targets are closely spaced [22], RANSAC operates directly on the data and can resolve independent linear structures in close proximity, making it more suitable for multi-target scenarios in dense environments [23].

2.4.5 Conclusion

This chapter covered the foundational concepts behind the tracking engines developed in this thesis, including range-only radar, Weighted Least Squares and Kalman filtering, data association, and track initiation methods. RANSAC was identified as the best fit for the cluttered, multi-target environments considered here, avoiding the single-point fragility of TPD and the resolution limits of the Hough transform. The RANSAC-based incubation framework built on these foundations is developed next, in Chapter 3.

Chapter 3

RANSAC Incubation Framework

Track initiation in dense clutter environments presents fundamental challenges that classical methods struggle to address. The M-of-N logic-based initiator, while computationally efficient, exhibits a characteristic threshold dilemma in high false alarm environments, a low confirmation threshold M produces excessive false track initiations as clutter returns satisfy the density requirement, while a high threshold causes genuine target tracks to be missed entirely [1], [17]. In cluttered range-only environments where false alarm rates substantially exceed target detection rates, this tradeoff becomes increasingly difficult.

The Hough transform addresses the batch processing requirement but introduces resolution limitations in parameter space that are problematic when multiple targets occupy similar range-rate bins simultaneously. RANSAC-based approaches represent a robust method of line finding, originally developed in the computer vision community [7] and subsequently adapted for dynamic target tracking. Niedfeldt and Beard [1], [24] demonstrated that RANSAC could be extended to dynamic target tracking through the recursive R-RANSAC framework, coupling hypothesis generation with a Kalman filter to track an unknown number of targets in clutter. In this thesis, RANSAC is used differently, as a one shot track initiation mechanism that operates on a sliding detection buffer and hands off a fully initialized state estimate to a separate tracking engine.

The framework proposed in this thesis positions RANSAC as a track initiation engine operating on detections that have not been claimed by active tracking engines. At each scan, confirmed tracks first compete for detections through the data association stage. Remaining unmatched detections are accumulated in a sliding window buffer, allowing detections to accumulate across multiple scans. RANSAC then searches this detection pool for linear structures consistent with constant velocity motion, promoting confirmed hypotheses as newly initiated tracks. Confirmed inliers are removed from the pool prior to subsequent searches, providing implicit data separation between simultaneously present targets and preventing structured trajectories from biasing future hypotheses. This also has the additional benefit of reducing the total amount of guesses in the RANSAC algorithm. Although developed and validated in a range-only radar context, the framework is theoretically sensor agnostic and applicable to any domain where targets produce approximately linear trajectories in a detection space, including computer vision applications.

3.1 Detection Buffer

The detection buffer $\mathcal{W}$ is a sliding window accumulator that collects unassigned detections across consecutive scans. At each scan k, detections that were not claimed by any active tracking engine are appended to the buffer. Detections older than the window length N_w are simultaneously expired, maintaining the buffer as a rolling collection of recent unassigned observations:

$$\mathcal{W}_k = \left\{ z_j \,:\, k - N_w \leq t_j \leq k,\ z_j \notin \mathcal{A}_k \right\} \tag{3.1}$$

where t_j denotes the scan index of detection z_j and $\mathcal{A}_k$ denotes the set of detections claimed by active tracks at scan k through the data association stage.

The sliding window serves two purposes. First, it provides depth sufficient for RANSAC to observe enough collinear detections to confirm a track, which is particularly important in degraded environments where a target may only be detected on a fraction of scans within the window. Second, by expiring old detections, the buffer remains bounded in size and prevents stale observations from corrupting hypothesis generation for targets that have since changed state or left the region of interest.

The window length N_w represents a fundamental tradeoff between start-up Time to Initialization (TTI) and robustness. A longer window accumulates more target evidence, improving detection probability in heavy clutter at the cost of increased initiation delay, and potential use of old information, thus increasing the chance of false tracks. A shorter window reduces start-up TTI and ensures fresh detections but may be insufficient to satisfy N_{min} for targets with low probability of detection. Parameter selection for N_w is discussed further in Section 3.3.

3.2 Inlier Consensus and Track Promotion

Upon completion of the RANSAC hypothesis search, the hypothesis with the greatest inlier support is evaluated against a minimum inlier count threshold N_{min}. A candidate track is promoted only when the inlier count satisfies:

$$|\mathcal{S}| \geq N_{min} \tag{3.2}$$

where $\mathcal{S}$ denotes the set of detections within the inlier distance threshold τ of the fitted line hypothesis. This density gate provides natural suppression of spurious hypotheses, as random clutter is geometrically unlikely to produce sufficient collinear support to satisfy N_{min}. Upon promotion, the inlier set $\mathcal{S}$ is removed from the detection pool $\mathcal{W}$, yielding a depleted pool:

$$\mathcal{W} \leftarrow \mathcal{W} \setminus \mathcal{S} \tag{3.3}$$

This greedy sequential extraction, similar to the sequential RANSAC approach of Vincent and Laganière [25], is repeated on the depleted pool until no further hypotheses satisfy the promotion threshold or the maximum number of iterations is reached. The motivation for pool depletion is that confirmed track inliers constitute structured trajectory and represent noise to any subsequent searches. Retaining them would bias future RANSAC iterations toward already confirmed structures, increasing both false hypothesis rates and computational load. The complete sequential extraction procedure is illustrated in Figure 3.1.

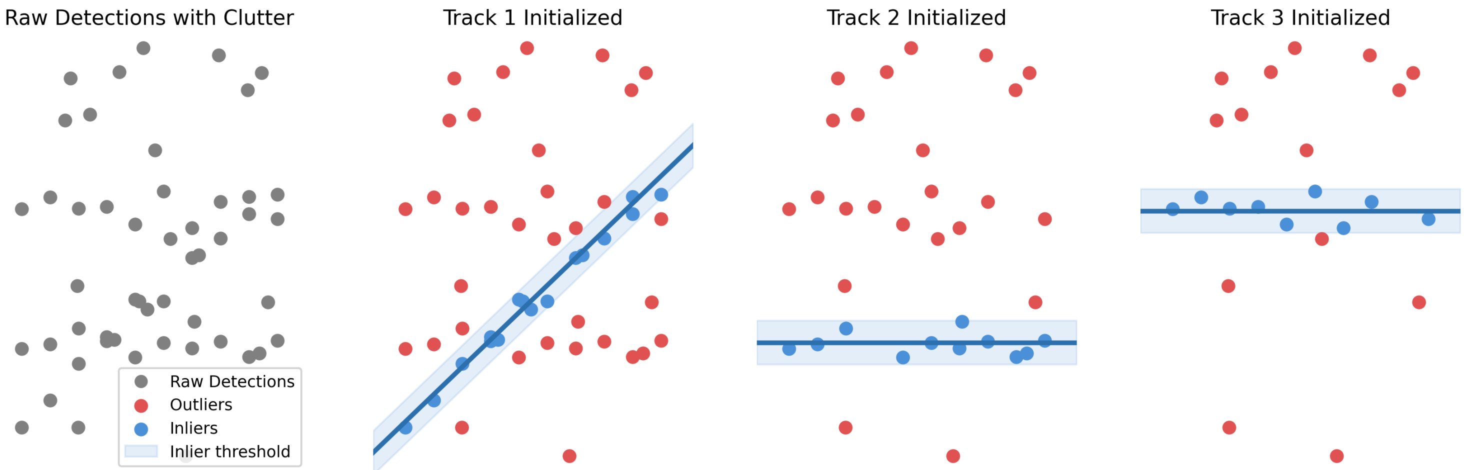

Figure 3.1: Sequential RANSAC pool depletion. Starting from a cluttered detection pool, tracks are extracted iteratively. Each promoted track's inliers are removed from the pool before the next search, preventing confirmed structures from biasing subsequent hypotheses.

Upon confirmation, the inlier set $\mathcal{S}$ is passed to a Weighted Least Squares estimator as described in Section 2.1 to derive the initial state estimate and covariance at track birth. The batch measurement vector and observation matrix are constructed from the N inlier detections, with time expressed relative to the current frame t_{now}:

$$\mathbf{y} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & (t_1 - t_{now}) \\ 1 & (t_2 - t_{now}) \\ \vdots & \vdots \\ 1 & (t_N - t_{now}) \end{bmatrix} \tag{3.4}$$

The weighting matrix $\mathbf{W} = \sigma_r^{-2}\mathbf{I}_N$ treats all inlier measurements as equally reliable. If per detection quality metrics were available, $\mathbf{W}$ could be constructed to reflect them, potentially improving initialization accuracy.

$$\mathbf{W} = \frac{1}{\sigma_r^2}\mathbf{I}_{N\times N} \tag{3.5}$$

The WLS solution yields the initial kinematic state estimate and its covariance:

$$\hat{\mathbf{x}}_{kin} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y} \tag{3.6}$$

$$\mathbf{P}_{kin} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1} \tag{3.7}$$

where $\hat{\mathbf{x}}_{kin} = [r_{now}, \dot{r}]^T$ gives the estimated range at the current frame and the estimated range-rate, and $\mathbf{P}_{kin}$ reflects the actual uncertainty in the batch estimate given the sensor noise [8]. These quantities are used directly as the initial state and covariance for the promoted track:

$$\hat{\mathbf{x}}_{init} = \hat{\mathbf{x}}_{kin}, \quad \mathbf{P}_{init} = \mathbf{P}_{kin} \tag{3.8}$$

3.3 Parameter Selection

Number of Iterations K

The required number of RANSAC iterations can be bounded analytically. Let p denote the inlier ratio of the detection pool, that is, the fraction of detections belonging to a true target rather than clutter. The probability that a randomly sampled pair of points are both inliers is p^2. The probability that a given iteration draws at least one clutter point, that is, fails to find a clean pair, is therefore:

$$P_{fail} = 1 - p^2 \tag{3.9}$$

Since each iteration is independent, the probability that all K iterations fail to find a clean pair is:

$$P_{all\_fail} = (1 - p^2)^K \tag{3.10}$$

The probability that at least one iteration finds a clean pair is the complement:

$$P_{success} = 1 - (1 - p^2)^K \tag{3.11}$$

Setting $P_{success} = P^*$ and solving for K:

$$P^* = 1 - (1 - p^2)^K \tag{3.12}$$

$$(1 - p^2)^K = 1 - P^* \tag{3.13}$$

$$K \log(1 - p^2) = \log(1 - P^*) \tag{3.14}$$

$$K = \frac{\log(1 - P^*)}{\log(1 - p^2)} \tag{3.15}$$

yielding the minimum number of iterations required to achieve a desired success probability P^* [7]:

$$K \geq \frac{\log(1 - P^*)}{\log(1 - p^2)} \tag{3.16}$$

In practice the inlier ratio p is not known a priori, but a worst case estimate of p can be derived from the expected clutter density and target detection probability, yielding a conservative upper bound on the required number of iterations. Setting $P^* = 0.99$ ensures that a clean hypothesis is found with high confidence. As the inlier ratio decreases, the required number of iterations grows rapidly. This inverse relationship means that K should be set conservatively to account for worst-case clutter conditions, where the fraction of true target detections in the pool is smallest. In practice, K should be set as large as computational resources permit, as additional iterations improve the probability of finding a valid hypothesis at negligible marginal cost once the analytical bound is satisfied.

Window Length N_w

The window length controls how many scans of evidence are available to RANSAC at each initiation attempt. A window that is too short may not accumulate sufficient collinear detections to satisfy N_{min}, particularly for targets with low probability of detection. A window that is too long introduces stale detections that may no longer be geometrically consistent with the current target state, and increases initiation TTI on start-up. In practice, N_w should be set large enough that a target with the expected probability of detection will produce at least N_{min} detections within the window with high probability.

Minimum Inlier Count N_{min}

The minimum inlier count is the primary gate against false track promotion. A value that is too low allows spurious collinear clutter alignments to generate false tracks, while a value that is too high increases initiation TTI and causes missed initiations for targets with sparse detections. N_{min} should be set below the expected number of target detections within a window of length N_w, given the target's probability of detection p_d. A reasonable starting point is:

$$N_{min} \leq p_d \cdot N_w \tag{3.17}$$

ensuring that a target present for the full window duration is likely to produce enough detections to satisfy the desired minimum promotion threshold.

Inlier Distance Threshold τ

The inlier threshold τ defines the perpendicular distance from a detection to the fitted line within which a detection is counted as an inlier. Setting τ too tight causes genuine target detections to be rejected due to measurement noise, reducing the effective inlier count. Setting τ too loose admits false clutter detections as inliers, inflating hypothesis support and increasing false track rates. Since τ governs the geometric tolerance of the line fit, it should be set according to the measurement noise characteristics of the sensor, independent of the inlier count threshold. A value of one to three standard deviations of the range measurement noise is a practical starting point.

3.4 Conclusion

This chapter presented the RANSAC-based track incubation framework used in this thesis. Detections that are not claimed by an active track are collected in a sliding window buffer. RANSAC then searches this buffer for groups of detections that line up along a constant velocity path.

When enough detections support the same line, that group is promoted to a new track, and Weighted Least Squares is used to turn it into an initial state estimate and covariance. Removing each confirmed group from the buffer before the next search lets multiple tracks be found from the same pool without interfering with each other. The parameter choices discussed in this chapter control the tradeoff between how quickly a track is confirmed and how confident that confirmation is. The two tracking engines built on top of this framework, RANSAC-GNNKF and RANSAC-TOMHT, are covered next in the following chapters.

Chapter 4

RANSAC-GNN Kalman Filter Engine

This chapter presents the first of the two proposed tracking engines. The RANSAC-GNNKF engine is designed for computational efficiency. It combines the RANSAC incubator developed in Chapter 3 with a Global Nearest Neighbor Kalman filter for data association. The result is a tracker capable of operating in real time while maintaining robust track initiation in high clutter environments. Section 4.1 defines the state model and process noise formulation. Section 4.2 describes the WLS-based track handover procedure. Section 4.3 presents the GNN association framework. Section 4.4 develops the recursive Kalman filter update equations, and Section 4.5 describes the track lifecycle and termination logic.

4.1 State Model

Under the constant velocity assumption, the motion of a target in range-only radar space is fully described by its range and range-rate. This two-state kinematic model is consistent with the linear trajectory assumption made by the RANSAC incubation stage in Chapter 3, where targets are expected to trace straight lines in the range-time plane. The state vector for each tracked target is defined as:

$$\mathbf{x} = \begin{bmatrix} r \\ \dot{r} \end{bmatrix} \tag{4.1}$$

where r denotes range in meters and $\dot{r}$ denotes range-rate in meters per second. The discrete-time state transition matrix under the constant velocity model is:

$$\mathbf{A}_d = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{4.2}$$

where Δt is the radar scan interval. The observation matrix mapping the state to the range measurement is:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.3}$$

Although the constant velocity model assumes range-rate is constant between scans, a small process noise is applied to prevent the filter from becoming overconfident when the CV assumption is mildly violated in practice. The process noise covariance is [13]:

$$\mathbf{Q}_d = q_p \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \tag{4.4}$$

where q_p is the model noise tuning parameter that governs the magnitude of velocity uncertainty injected at each scan interval. The measurement noise is modeled as a scalar range variance:

$$R = \sigma_r^2 \tag{4.5}$$

where σ_r is the sensor range noise standard deviation. Both q_p and σ_r are fixed parameters set prior to simulation and held constant across all scenarios evaluated in this work.

4.2 RANSAC Handover and Track Initialization

Upon promotion from the RANSAC incubation buffer, the inlier set $\mathcal{S}$ satisfying $|\mathcal{S}| \geq N_{min}$ is passed to the WLS estimator described in Section 3. The WLS procedure yields the initial state estimate $\hat{\mathbf{x}}_{init} = \hat{\mathbf{x}}_{kin}$ and initial covariance $\mathbf{P}_{init} = \mathbf{P}_{kin}$, both projected to the current frame t_{now}. This provides a warm, statistically principled filter initialization that avoids the cold start common to filter-based track initiation methods.

4.3 Association

At each scan, the predicted measurement for each active track is computed and compared against all detections in the current scan. The innovation between track i and detection j is computed as:

$$\hat{y}_{i,j} = z_j - \mathbf{C}\hat{\mathbf{x}}_{i,k|k-1} \tag{4.6}$$

with associated innovation covariance:

$$S_i = \mathbf{C}\mathbf{P}_{i,k|k-1}\mathbf{C}^T + R \tag{4.7}$$

The Mahalanobis distance $d^2_{i,j}$ is then computed as defined in Section 2.3.1. Candidate associations are gated admitting only pairings within the 99% confidence gate ($\chi^2_{2,0.99} = 9.21$) [13]:

$$\Omega_{i,j} = \begin{cases} d^2_{i,j} & \text{if } d^2_{i,j} \leq 9.21 \\ \infty & \text{otherwise} \end{cases} \tag{4.8}$$

Track-detection pairs that fall outside the Mahalanobis gate are assigned infinite cost in , preventing them from being selected regardless of the optimization. The remaining gated pairs are passed to the Hungarian algorithm described in Section 2.3.2, which finds the globally optimal one-to-one assignment by minimizing total cost across all valid pairs simultaneously. The GNN assignment is obtained by solving:

$$\min_{\chi} \sum_i \sum_j \Omega_{i,j} \chi_{i,j} \tag{4.9}$$

where the binary decision variable $\chi_{i,j}$ is defined as:

$$\chi_{i,j} = \begin{cases} 1 & \text{if track } i \text{ is assigned to detection } j \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

with the constraint $\chi_{i,j} \in \{0,1\}$ enforcing a hard one-to-one assignment. GNN makes a hard commitment to a single assignment at each scan based only on current-scan information. The RANSAC-TOMHT engine described in Chapter 5 addresses this limitation by deferring association decisions across multiple scans.

4.4 Kalman Filter Update

At each scan the filter propagates the prior state estimate and covariance forward in time using the system dynamics model, then corrects the prediction using the assigned detection if one exists. The time update is:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_d \hat{\mathbf{x}}_{k-1|k-1} \tag{4.11}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_d \mathbf{P}_{k-1|k-1} \mathbf{A}_d^T + \mathbf{Q}_d \tag{4.12}$$

When a matched detection $z_{matched}$ exists, the innovation and its covariance are computed as:

$$\tilde{y}_k = z_{matched} - \mathbf{C}\hat{\mathbf{x}}_{k|k-1} \tag{4.13}$$

$$S_k = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + R \tag{4.14}$$

The Kalman gain is computed as:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T S_k^{-1} \tag{4.15}$$

The measurement update is then applied:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{y}_k \tag{4.16}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k \mathbf{C}\right) \mathbf{P}_{k|k-1} \tag{4.17}$$

When no detection is assigned, the track coasts on the predicted state with no measurement correction. The covariance continues to grow each scan during coasting, reflecting increasing uncertainty until a detection is assigned at a future scan or the track is terminated.

4.5 Track Management

Once a track is promoted by the RANSAC incubation stage it remains active until it accumulates too many consecutive missed detections. At each scan, a miss counter N_{miss} is incremented when no detection is assigned to the track and reset to zero when a detection is successfully matched. Track termination is governed by a termination threshold T_{kill}:

$$\text{Status}_i = \begin{cases} \text{Active} & \text{if } N_{miss,i} < T_{kill} \\ \text{Terminated} & \text{if } N_{miss,i} \geq T_{kill} \end{cases} \tag{4.18}$$

The miss counter evolves as:

$$N_{miss,k} = \begin{cases} 0 & \text{if } \sum_j \chi_{i,j} = 1 \\ N_{miss,k-1} + 1 & \text{if } \sum_j \chi_{i,j} = 0 \end{cases} \tag{4.19}$$

T_{kill} controls the tolerance for missed detections before a track is dropped. A low value terminates tracks quickly, reducing false track persistence at the cost of dropping valid tracks that experience temporary detection gaps due to low probability of detection. A high value keeps tracks alive through longer gaps but risks retaining false tracks that have drifted from any real target. Since each scan corresponds to a fixed time interval Δt, T_{kill} equates to the maximum allowable coast time before a track is terminated. During coasting the Kalman filter prediction covariance grows with each missed scan, expanding the Mahalanobis gate and increasing the risk of false associations. Keeping T_{kill} as small as practical limits this covariance growth and keeps the gate tight.

4.6 Computational Complexity

The RANSAC-GNNKF engine is the computationally lighter of the two engines proposed in this thesis. Each active track maintains a single state estimate and covariance, so the per-scan cost scales with the number of active tracks. The dominant cost at each scan is the Hungarian assignment, which operates on a cost matrix of size $N_{tracks} \times N_{detections}$ and grows with both quantities. The primary parameter controlling computational load is T_{kill}. A smaller value terminates tracks sooner, reducing the number of active tracks and the size of the assignment problem at each scan. A runtime comparison against the RANSAC-TOMHT engine is provided in Chapter 6.

4.7 Conclusion

This chapter presented the RANSAC-GNNKF engine, which pairs the RANSAC incubation framework with a Global Nearest Neighbor Kalman filter for real-time tracking. For the near constant velocity trajectories considered in this work, a higher-order motion model would introduce unnecessary complexity and degrade performance by injecting uncertainty that is not present in the scenario. A more capable motion model may be warranted if the framework is extended to maneuvering targets in future work. Detections are assigned to tracks each scan using Mahalanobis gating and the Hungarian algorithm, giving a hard, globally optimal assignment at low computational cost. This makes RANSAC-GNNKF the lighter weight of the two engines developed in this thesis, at the cost of committing to associations immediately rather than deferring them across scans. The RANSAC-TOMHT engine, developed in the next chapter addresses this limitation directly.

Chapter 5

RANSAC-TOMHT Engine

This chapter presents the second of the two proposed tracking engines. The RANSAC-TOMHT engine differs from the RANSAC-GNNKF engine by replacing the hard single-assignment data association step with a track-oriented multi-hypothesis tracker. Rather than committing to a single detection assignment at each scan, the TOMHT engine maintains a tree of association hypotheses per track, deferring the commitment decision until sufficient evidence has accumulated across N scans. This is intended to reduce false associations in ambiguous scenarios where targets cross or operate in close proximity, at the cost of greater computational overhead. The RANSAC incubation stage and WLS initialization procedure are shared with the GNNKF engine and are not repeated here.

5.1 Hypothesis Generation

Upon promotion from the RANSAC incubation stage, a track enters the TOMHT engine with a single hypothesis initialized from the WLS state estimate and covariance as described in Section 3. At each subsequent scan, every surviving hypothesis splits into multiple children, one for each detection that falls within its Mahalanobis gate, plus one additional branch representing the case where the target was not detected that scan. Each child hypothesis maintains its own Kalman filter state and covariance, and is updated according to the specific sequence of detections assigned to it since track birth. Each hypothesis is scored by its accumulated log-likelihood ratio (LLR), updated at each scan as defined in Section 2.3.6. For a detection branch assigned measurement z_j [2], [19]:

$$L(k) = L(k-1) + \ln(\hat{P}_D) + \ln \mathcal{N}\big(\tilde{y}_{i,k};\, 0,\, S_{i,k}\big) - \ln(\lambda) \tag{5.1}$$

where $\mathcal{N}(\tilde{y}_{i,k};\, 0,\, S_{i,k}) = \frac{1}{\sqrt{2\pi S_{i,k}}} \exp\left(-\frac{\tilde{y}_{i,k}^2}{2S_{i,k}}\right)$, with $\tilde{y}_{i,k} = z_j - \mathbf{C}\hat{\mathbf{x}}_{i,k|k-1}$ and $S_{i,k} = \mathbf{C}\mathbf{P}_{i,k|k-1}\mathbf{C}^T + R$. For a missed detection branch:

$$L(k) = L(k-1) + \ln(1 - \hat{P}_D) \tag{5.2}$$

Since $\ln(1-\hat{P}_D)$ is always negative, a hypothesis that consistently misses detections accumulates a worsening score relative to hypotheses that receive matched detections. The LLR score is used solely to rank competing hypotheses and select the best branch at the N-scan commit point. Track termination is governed independently by a miss counter on the best hypothesis, as described in Section 5.2.

5.2 Hypothesis Pruning and Track Management

Without pruning, the number of hypotheses per track grows without bound as the tree expands across scans. The RANSAC-TOMHT engine uses N-scan pruning to keep the hypothesis set tractable, as introduced in Section 2.3.6. At each scan the hypothesis tree branches on all gated detections plus one missed detection branch. After N consecutive scans the highest-scoring hypothesis is selected as the committed root:

$$L^* = \max_l \; L_l(k) \tag{5.3}$$

All competing branches are pruned and branching resumes from L^* at the next scan with the tree depth reset to one. The track state and covariance are updated from the committed hypothesis at each commit point:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^*, \quad \mathbf{P}_k = \mathbf{P}_k^* \tag{5.4}$$

This bounds the hypothesis tree to a fixed depth of N scans at all times regardless of detection density, providing a hard computational guarantee at the cost of occasionally committing to an incorrect association before enough evidence has accumulated to resolve the ambiguity.

The LLR score is used solely to rank competing hypotheses and select the best branch at the N-scan commit point. It does not directly govern track termination. Track termination follows the same logic as the RANSAC-GNNKF engine described in Section 4.5. The miss counter on the best hypothesis increments by one each scan a detection is not assigned and resets to zero when one is. When the miss counter reaches T_{kill}, the track is terminated. Since T_{kill} is identical between both engines, any difference in track persistence observed in the results reflects the deferred association decisions of the TOMHT framework rather than differences in termination logic.

The RANSAC incubation stage complements the N-scan pruning strategy by bounding the number of tracks that enter the hypothesis generation stage, limiting the total hypothesis space at track initialization.

5.3 Computational Complexity

The RANSAC-TOMHT engine is the more computationally expensive of the two proposed engines. Unlike the RANSAC-GNNKF engine which maintains a single state estimate per track, the TOMHT engine maintains a tree of hypotheses per track that grows with each scan until the N-scan commit point. Three parameters directly control the computational load of the engine.

The N-scan window is the primary lever. Each scan within the window multiplies the hypothesis count by the number of gated detections plus one, so a smaller N produces a shallower tree and proportionally lower cost. A larger N allows more evidence to accumulate before committing, improving association accuracy in ambiguous scenarios at the cost of a deeper tree.

The Mahalanobis gate threshold $\chi^2_{2,0.99}$ [13] controls how many detections are admitted per hypothesis per scan. A tighter gate reduces the branching factor directly and fewer gated detections means fewer child hypotheses spawned per branch. A looser gate admits more detections and produces a wider tree. Since the gate threshold also affects association quality, it should be tuned with both accuracy and computational cost in mind.

The termination threshold T_{kill} controls how long tracks remain active. A smaller T_{kill} terminates tracks sooner, reducing the number of active tracks competing for detections and the total hypothesis count across the scene. A larger T_{kill} keeps tracks alive through longer detection gaps at the cost of maintaining more active hypothesis trees simultaneously.

The RANSAC incubation stage cost is identical between both engines since it operates on the same detection buffer independently of the tracking engine. A detailed runtime comparison between both engines is provided in Chapter 6.

5.4 Simulator Architecture

All scenarios are implemented as discrete-time simulations operating at an update rate of 1Hz. At each scan the simulator generates target detections, clutter returns, and debris detections independently, producing a single unlabeled detection pool presented to the tracker. Ground truth target ranges are recorded separately at each scan for evaluation. Collision events are modeled as instantaneous. Both colliding targets persist through the collision frame and a fixed number of debris objects are spawned at the collision range, each assigned a constant velocity drawn from a uniform distribution. All targets and debris share a common probability of detection p_d applied independently at each scan.

Targets are the things of interest, debris are the tracks born of the collision of targets, and clutter refers to noise or background detections unrelated to any true object. Target and debris detections are generated by applying zero-mean Gaussian noise with standard deviation $\sigma_r = 2.5$m to the true target range, with each detection occurring independently with probability p_d. The range extent is $[0, 500]$m divided into 100 range bins of 5m each. All measurements are reported as range values.

The four tracking engines are evaluated on identical measurement sequences for each scenario condition. Engine parameters are held fixed across all scenarios and conditions and are summarized in Table 5.1.

Table 5.1: Fixed engine parameters used across all scenarios and conditions.

Parameter	Description	Value
σ_r	Range noise standard deviation	2.5m
K	RANSAC iterations	1000
N_{min}	Minimum inliers for promotion	22
τ	RANSAC inlier threshold	4m
N_w	Sliding window length	30 scans
T_{kill}	Miss counter termination threshold	5 scans
$\chi^2_{2,0.99}$	Mahalanobis gate threshold	9.21
q_p	Process noise scalar	0.001
M	M-of-N required hits	20
N	M-of-N window length	30 scans
N_{scan}	TOMHT hypothesis tree depth	3 scans

To characterize run to run variability and provide statistically grounded performance estimates, a Monte Carlo study was conducted over all three primary evaluation scenarios. For each scenario and condition combination, 1000 independent trials were generated by varying the random seed controlling measurement noise and clutter realization, while holding all target trajectories, engine parameters, and scenario geometry fixed. The simulator and RANSAC sampling process were seeded independently at each trial to ensure uncorrelated randomness across measurement generation and hypothesis search. All quantitative results reported in Chapter 6 represent the mean and standard deviation computed across these 1000 trials. Tracking output figures show a single representative run selected from the nominal seed for each condition.

5.5 Clutter Model

Clutter is modeled as a uniform Poisson process over the full range at each scan [12]. The number of clutter returns per scan is drawn from a Poisson distribution with mean $\lambda \cdot N_{bins}$, where λ is the expected false alarm rate per bin per scan and $N_{bins} = 100$ is the number of range bins. Each clutter return is placed at a range drawn uniformly from $[0, 500]$m. Clutter returns carry no label and are indistinguishable from target detections in the measurement pool presented to the tracker.

The clutter density λ is varied in the sensitivity analysis to characterize engine performance across operating conditions. The nominal condition uses $\lambda = 0.10$, producing an average of 10 false alarms per scan. The sensitivity sweep covers $\lambda \in \{0.10, 0.20, 0.30\}$, reaching an average of 30 false alarms per scan at the upper end. This range was determined from a preliminary sweep on Scenario 1 that identified $\lambda = 0.30$ as the point at which false track promotion begins to increase significantly for the RANSAC-based engines.

5.6 Target Motion Model

All targets follow a constant range rate model in range space. A target at initial range r_0 with range-rate $\dot{r}$ produces a true range at time t of:

$$r(t) = r_0 + \dot{r} \cdot t \tag{5.5}$$

This linear trajectory in the range-time plane is the geometric structure exploited by the RANSAC incubation framework described in Chapter 3. Targets that exit the maximum and minimum ranges $[0, 500]$m are removed from the scene.

The detection probability p_d is varied in the sensitivity analysis alongside clutter density. The nominal condition uses $p_d = 0.90$. The sensitivity sweep covers $p_d \in \{0.90, 0.80, 0.70\}$, with $p_d = 0.70$ identified from the preliminary sweep as the point at which the RANSAC incubator begins missing targets due to insufficient detections accumulating within the sliding window to satisfy N_{min}.

5.7 Scenario Definitions

Three scenarios are evaluated in this thesis, each presenting a different pre-collision tracking geometry of increasing difficulty. All three include a primary target that is struck in a collision and an interceptor that closes in and causes the collision. Horizontal and Angled and Symmetric Crossing also include persistent flyby targets that remain present throughout the scenario. Symmetric Crossing differs slightly in that neither of its two colliding targets is stationary, so the primary and interceptor roles are assigned arbitrarily to maintain consistency with the other scenarios.

Figures throughout this chapter use a consistent color scheme. Blue denotes the primary target, red the interceptor, and purple any flyby targets, with debris shown in orange and clutter in gray. Dashed lines indicate ground truth, while scattered points of the same color are the corresponding noisy detections generated from that truth. This mapping is held fixed across every scenario and condition.

5.7.1 Scenario 1 — Simple Intercept

This scenario presents the simplest tracking geometry evaluated in this thesis. A stationary horizontal target sits at a constant range of 250m while an angled target closes at 4.0m/s from low range. The two targets meet at 250m at $t = 60$s, at which point six debris objects are spawned at the collision point with range-rates drawn uniformly from $[-5, 5]$m/s. The debris fan expands outward from the collision point for the remainder of the 120 second scenario. Clutter is generated at the nominal density of $\lambda = 0.10$ false alarms per bin per scan, producing an average of 10 false alarms per scan. This scenario serves as the baseline evaluation, isolating tracker performance on a single collision event with a modest clutter background. The scenario geometry is illustrated in Figure 5.1.

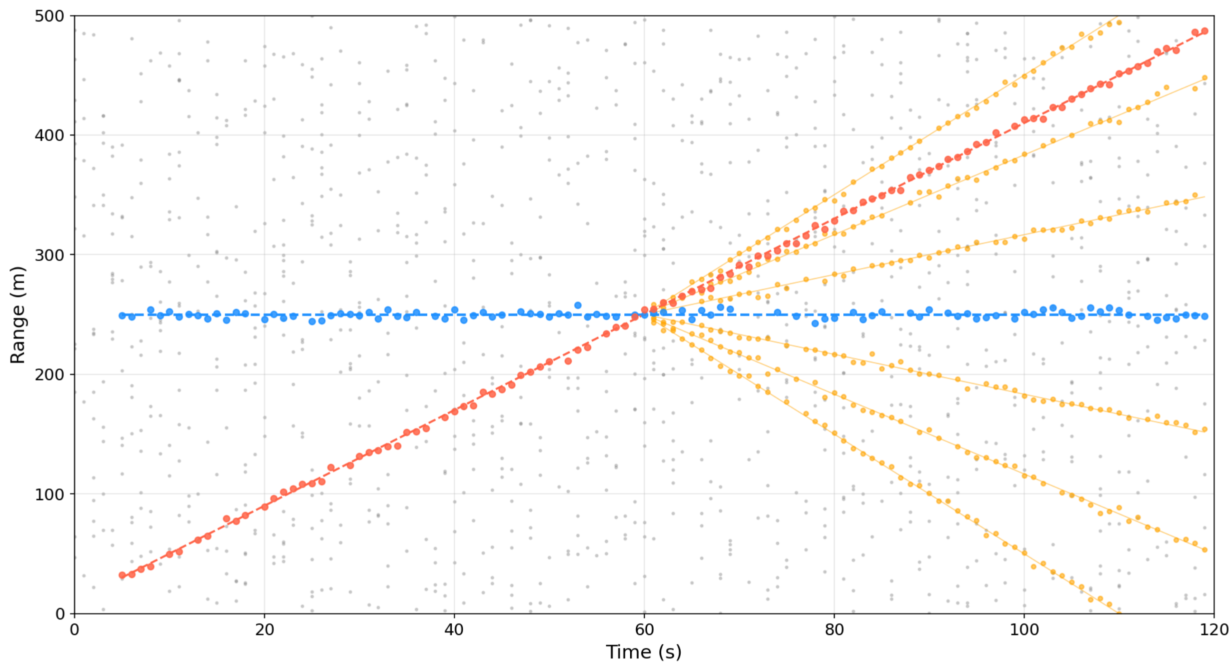

Figure 5.1: Scenario 1 — Simple Intercept. A horizontal target at 250m is struck by an angled closing target at $t = 60$s, spawning six debris children.

5.7.2 Scenario 2 — Horizontal and Angled

This scenario increases the target count and tracking complexity relative to Scenario 1. Four targets are present simultaneously: two horizontal flyby targets at constant ranges of 380m and 120m, a third horizontal target at 250m, and an angled target closing at 3.5m/s from high range. The angled target strikes the horizontal target at 250m at $t = 60$s, at which point six debris objects are spawned with range-rates drawn uniformly from $[-5, 5]$m/s. The two flyby targets remain active throughout the scenario, requiring the tracker to maintain multiple simultaneous confirmed tracks while managing the post-collision debris fan. The scenario runs for 120 seconds and is illustrated in Figure 5.2.

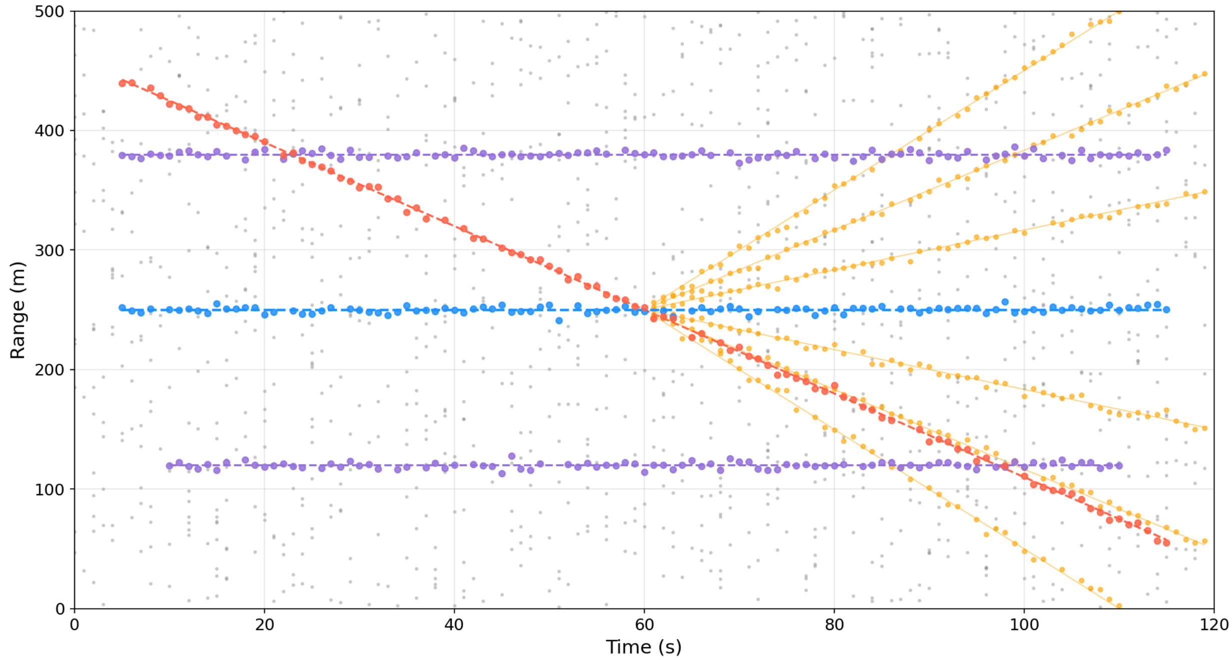

Figure 5.2: Scenario 2 — Horizontal and Angled. Two horizontal flyby targets at 380m and 120m remain active throughout while an angled target strikes a third horizontal target at 250m at $t = 60$s, spawning six debris children.

5.7.3 Scenario 3 — Symmetric Crossing

This scenario presents the most geometrically challenging pre-collision configuration of the three. Four targets are present: two horizontal flyby targets at constant ranges of 380m and 120m, and two colliding targets approaching each other symmetrically from opposite ends of the range extent. The first closes at 3.0m/s from high range and the second opens at 3.0m/s from low range, meeting at 250m at $t = 60$s, at which point six debris objects are spawned at the collision point with range-rates drawn uniformly from $[-5, 5]$m/s, and both targets continue on their original trajectories. The symmetric geometry produces a direct crossing event in the range-time plane where both colliding trajectories occupy the same range at the collision frame, creating maximum association ambiguity at the crossing point. The scenario runs for 120 seconds and is illustrated in Figure 5.3.

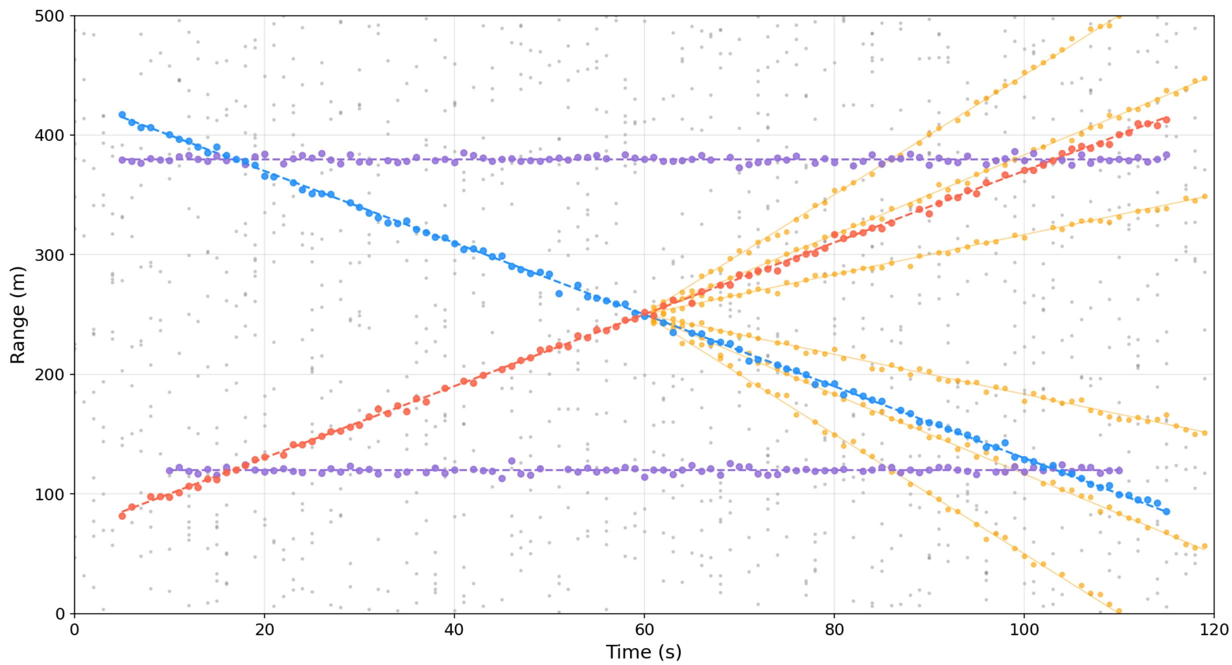

Figure 5.3: Scenario 3 — Symmetric Crossing. Two horizontal flyby targets bracket the scene while two targets approach symmetrically from opposite ends of the range extent, crossing at 250m at $t = 60$s and spawning six debris children. The symmetric crossing geometry produces maximum association ambiguity at the collision point.

5.7.4 WLS Validation Scenario — Collision with Debris

This scenario is used exclusively for the WLS initialization covariance validation study described in Section 6.5. A stationary target at range $r = 0$m and an incoming target at range $r = 100$m close at constant velocity and collide at $t = 5.0$s, at which point both are removed from the scene and ten debris objects are spawned at the collision point with range-rates drawn uniformly from $[-40, 40]$m/s. The debris fan expands outward from the collision point for the remainder of the scenario, producing a dense set of diverging linear trajectories in the range-time plane.

The scenario runs at 200Hz for 10 seconds, producing 2000 total frames. Clutter is modeled as a Poisson process with a mean rate of 30 returns per scan, uniformly distributed over $[-250, 250]$m. Range measurements are corrupted by zero-mean Gaussian noise with standard deviation $\sigma_r = 0.5$m. All targets are detected with probability $p_d = 1.0$.

Two evaluation windows are extracted from the scenario for analysis. Window 1 covers frames 1 to 100, capturing the pre-collision regime with two clean constant velocity targets in heavy clutter. Window 2 covers frames 1201 to 1300, capturing the post-collision debris field with ten simultaneously active targets and diverse range-rates expanding through the same clutter environment. Together the two windows characterize WLS initialization performance across both simple and highly complex tracking conditions. The ground truth targets and clutter field for both windows are illustrated in Figure 5.4.

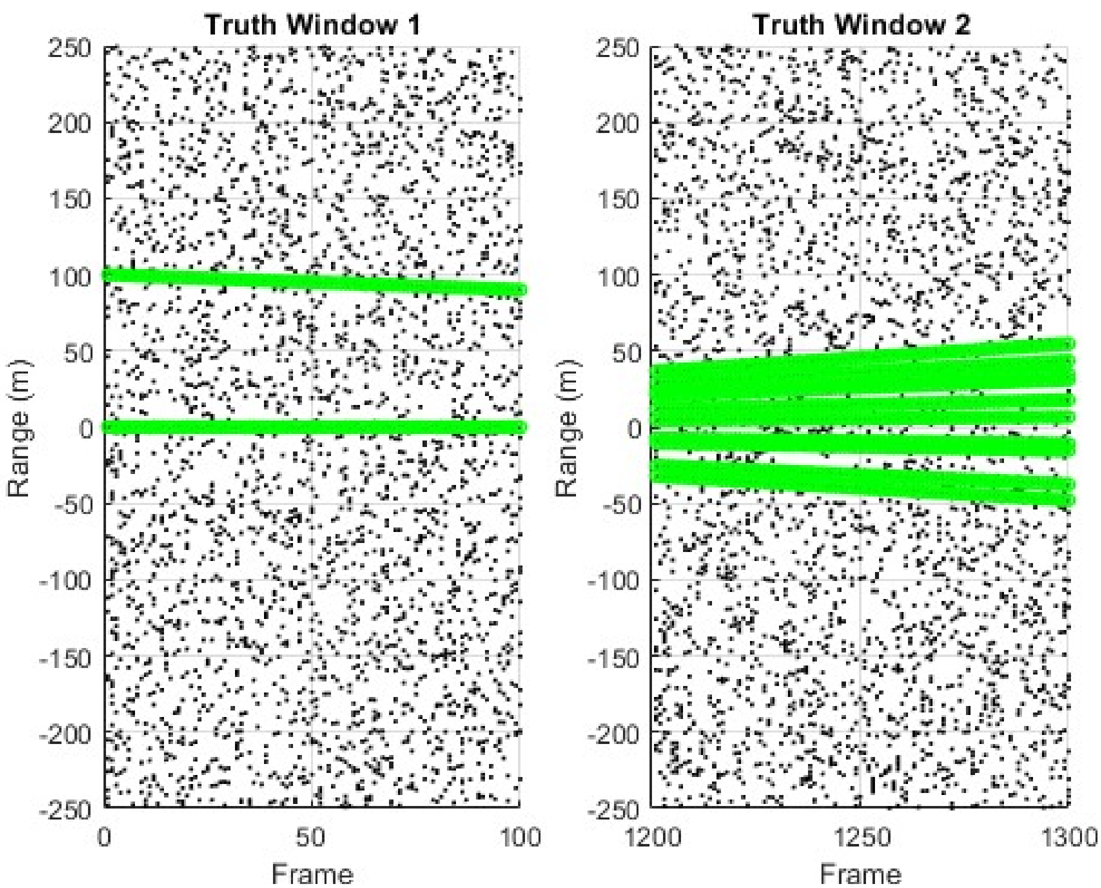

Figure 5.4: Ground truth targets (green) and clutter detections (black) for the two Monte Carlo evaluation windows. Window 1 shows the pre-collision regime with two targets. Window 2 shows the post-collision debris fan with ten simultaneously active targets.

Chapter 6

Results and Performance Comparison

6.1 Performance Metrics

Evaluation of multi-object tracking systems requires metrics that capture both the accuracy of individual track estimates and the correctness of the overall track set, including false tracks, missed targets, and assignment errors. The metrics used in this work are drawn from the CLEAR MOT framework introduced by Bernardin and Stiefelhagen [26] and standard state estimation error analysis.

6.1.1 Multiple Object Tracking Accuracy (MOTA)

Multiple Object Tracking Accuracy (MOTA) is a composite metric that summarizes the overall tracking quality by penalizing three types of errors across all scans [26]:

$$\text{MOTA} = 1 - \frac{\sum_k \left(\text{FN}_k + \text{FP}_k + \text{IDSW}_k\right)}{\sum_k \text{GT}_k} \tag{6.1}$$

where FN_k is the number of false negatives at scan k, FP_k is the number of false positives, IDSW_k is the number of identity switches, and GT_k is the number of ground truth targets present. MOTA ranges from $-\infty$ to 1, with higher values indicating better performance. A MOTA of 1 indicates perfect tracking with no missed targets, no false tracks, and no identity switches.

6.1.2 Multiple Object Tracking Precision (MOTP)

Multiple Object Tracking Precision (MOTP) measures the average localization accuracy of correctly matched track-to-target assignments [26]:

$$\text{MOTP} = \frac{\sum_{k,i} d_{k,i}}{\sum_k c_k} \tag{6.2}$$

where $d_{k,i}$ is the distance between the matched track and its assigned ground truth target at scan k, and c_k is the number of matches at scan k. In this work $d_{k,i}$ is the absolute range error between the estimated and true target range. MOTP captures how accurately the filter estimates target state, independently of whether the correct number of tracks is maintained.

6.1.3 Root Mean Square Error (RMSE)

Detection level Root Mean Square Error (RMSE) measures the state estimation accuracy of confirmed tracks matched to ground truth trajectories. It is computed over all matched track-target detection pairs across all scans:

$$\text{RMSE} = \sqrt{\frac{\sum_{t,i} \left(d_{t,i}\right)^2}{\sum_t c_t}} \tag{6.3}$$

where $d_{t,i}$ is the range error between matched track i and its assigned ground truth target at scan t, and c_t is the number of matched pairs at scan t. While numerically related to MOTP, RMSE is retained as a complementary metric because its squared error formulation penalizes large deviations more heavily than small ones, making it more sensitive to edge cases where the filter briefly diverges or locks onto the wrong target. This makes it a useful diagnostic alongside MOTP for identifying scenarios where average performance is acceptable but occasional large errors are present.

6.1.4 Time to Initialization (TTI)

Time to Initialization (TTI) is defined as the number of scans elapsed between a target's first appearance in the detection pool and the first scan at which any detection associated with a confirmed track matches that target within the 5m range threshold, including detections accumulated during the incubation window prior to promotion. Lower TTI indicates faster response to new targets, which is particularly important in scenarios where targets are present for a short duration or undergo rapid state changes shortly after entering the scene. Mean and standard deviation of TTI are reported for each scenario and supports direct comparison between the RANSAC and M-of-N initiators.

6.1.5 Run Time

Run time is reported as the total time required to process all scans in a scenario, measured in seconds. This provides a direct comparison of computational cost between the RANSAC-GNNKF and RANSAC-TOMHT engines and their M-of-N counterparts. All timing measurements are collected on the same hardware under identical conditions to ensure a fair comparison.

6.2 Scenario 1 Results — Simple Intercept

This section presents the tracking performance of all four engines on the Simple Intercept scenario. Three conditions are evaluated for this scenario, each varying a single environmental factor from its nominal value while holding everything else fixed. The nominal condition uses a clutter density of $\lambda = 0.10$ and a probability of detection of $p_d = 0.90$. The clutter sensitivity condition raises clutter density to $\lambda = 0.30$ while holding $p_d = 0.90$, stressing the false track suppression capability of each initiator. The dropout sensitivity condition instead lowers probability of detection to $p_d = 0.70$ while holding $\lambda = 0.10$, stressing each engine's ability to maintain a track through missed detections.

6.2.1 Nominal Performance

Figures 6.1 through 6.4 show the tracking output for all four engines under nominal conditions ($\lambda = 0.10$, $p_d = 0.90$), and Table 6.1 summarizes the quantitative results. RANSAC-GNNKF leads with a mean MOTA of $86.81 \pm 3.11\%$ and near-zero initiation TTI of 0.28 ± 0.28 scans, meaning confirmed tracks were matched to targets on or near the same scan they first appeared in the detection pool. MofN-GNNKF scores $43.22 \pm 13.43\%$ MOTA despite sharing the same GNNKF engine, a gap of roughly 44 percentage points caused by false track proliferation. Under nominal clutter the M-of-N threshold is low enough to confirm real targets, but it also confirms enough spurious clutter alignments to drive up the false positive count significantly.

RANSAC-TOMHT scores $83.68 \pm 2.23\%$ with a mean initiation TTI of 2.87 ± 0.93 scans, about 3 percentage points behind RANSAC-GNNKF. The added TTI comes from the TOMHT engine claiming detections across all open hypothesis branches before the N-scan commit, which leaves fewer detections available in the RANSAC pool for new track discovery. MOTP and RMSE are consistently lower for both RANSAC engines, consistent with the WLS initialization producing a better initial state estimate than the zero-velocity assumption used by the M-of-N initiators.

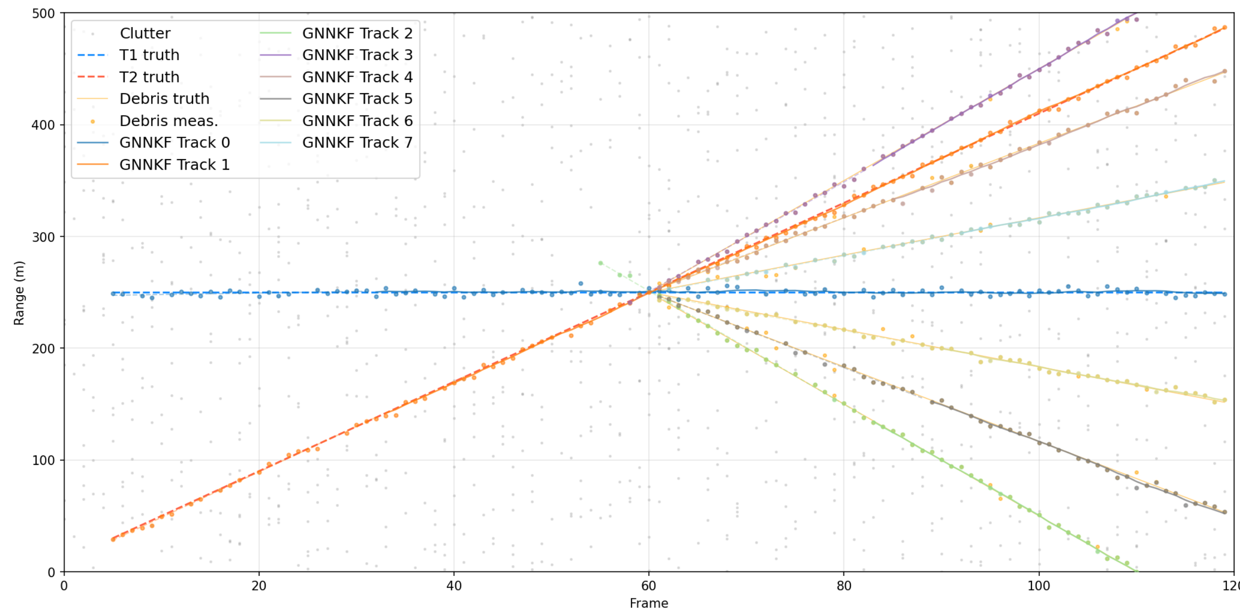

Figure 6.1: Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-GNNKF engine.

RANSAC-GNNKF cleanly resolves all tracks, matching both pre-collision targets and every debris child with continuous estimates.

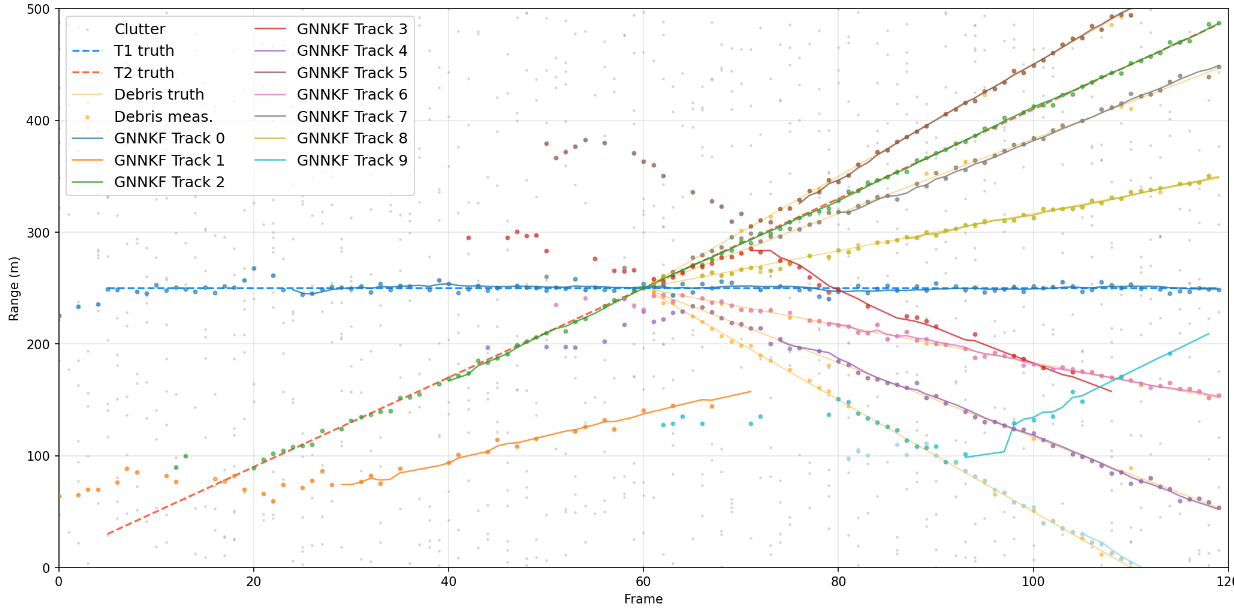

Figure 6.2: Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-GNNKF engine.

MofN-GNNKF shows several spurious track segments not aligned with any true target. These artifacts reflect the false track proliferation driving the engine's substantially lower MOTA.

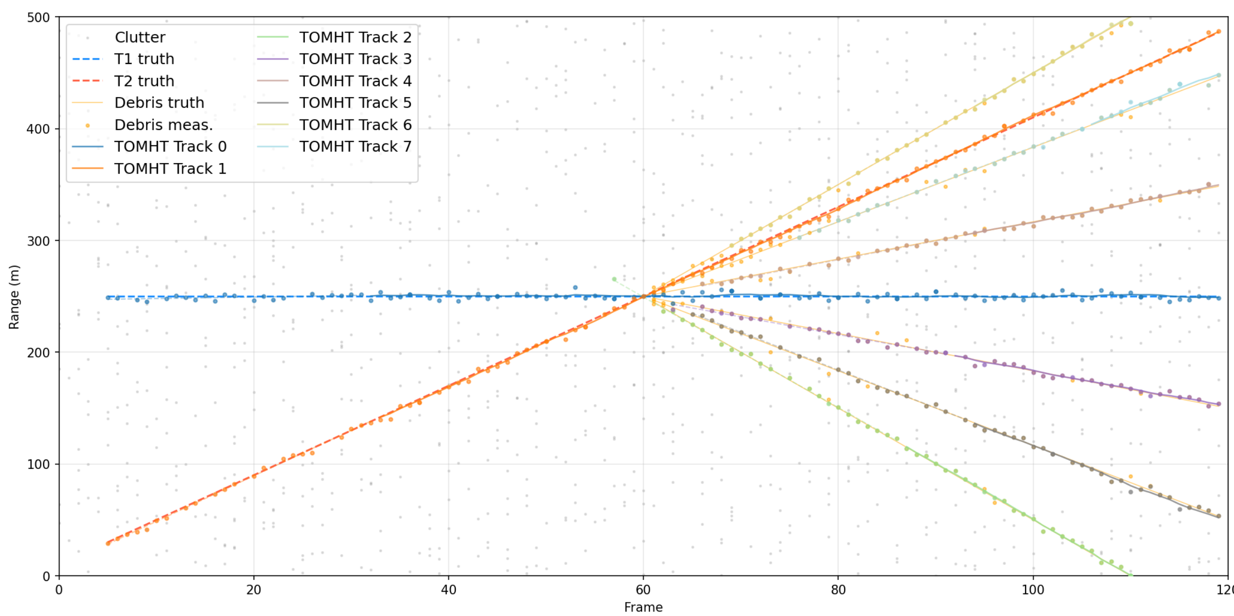

Figure 6.3: Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-TOMHT engine.

RANSAC-TOMHT produces clean target and debris tracks with no visible false branches. Its modestly higher TTI reflects detections held across open hypothesis branches prior to the N-scan commit.

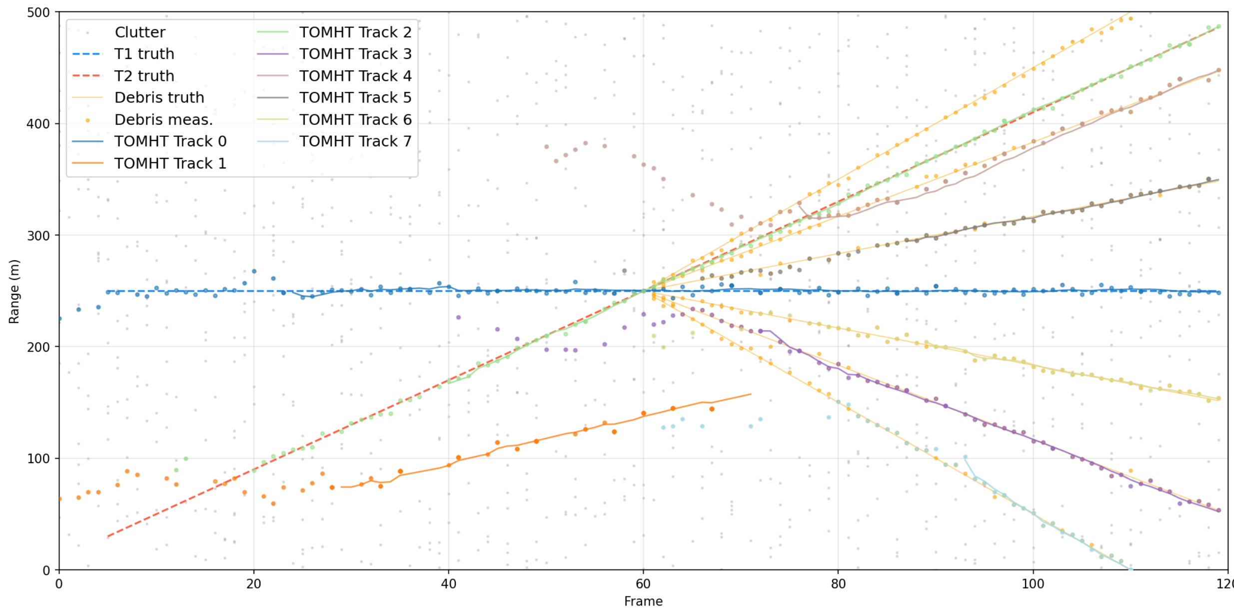

Figure 6.4: Scenario 1 — Simple Intercept, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-TOMHT engine.

MofN-TOMHT exhibits fragmentation similar to its GNNKF counterpart, including multiple false track segments. This engine records the highest TTI of all four in this condition.

Table 6.1: Tracking Performance for Scenario 1 — Simple Intercept (Nominal Conditions: $\lambda = 0.10$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	86.81 ± 3.11	1.1797 ± 0.1112	1.5365 ± 0.1255	0.28 ± 0.28	3.02 ± 0.10
MofN-GNNKF	43.22 ± 13.43	1.3756 ± 0.1350	1.7766 ± 0.1425	2.39 ± 1.48	0.10 ± 0.02
RANSAC-TOMHT	83.68 ± 2.23	1.1888 ± 0.1053	1.5488 ± 0.1151	2.87 ± 0.93	3.02 ± 0.11
MofN-TOMHT	47.19 ± 12.43	1.3692 ± 0.1229	1.7704 ± 0.1286	5.04 ± 2.07	0.29 ± 0.05

6.2.2 Clutter Sensitivity

Figures 6.5 through 6.8 show the tracking output under heavy clutter conditions ($\lambda = 0.30$, $p_d = 0.90$), and Table 6.2 summarizes the quantitative results. This is the most important condition for evaluating the RANSAC incubation framework, as both M-of-N engines collapse to deep negative MOTA while the RANSAC engines prove significantly more resilient. A negative MOTA means the false track count exceeded the total ground truth frame count, indicating the initiator has lost the ability to discriminate targets from clutter entirely. RANSAC-GNNKF scores $-2.96 \pm 20.98\%$ and RANSAC-TOMHT scores $21.51 \pm 21.24\%$, with RANSAC-TOMHT pulling ahead, which may reflect its deferred commitment allowing it to avoid some false associations that GNNKF commits to immediately. Both RANSAC engines see their MOTA drop significantly from nominal, driven by increased false track promotion as clutter density approaches the point where random collinear alignments begin satisfying N_{min}. Runtime increases substantially for both RANSAC engines at this clutter level, reflecting the larger detection pool the incubator must search each scan. MOTP and RMSE remain lower for both RANSAC engines despite the heavy clutter, consistent with the pattern established under nominal conditions.

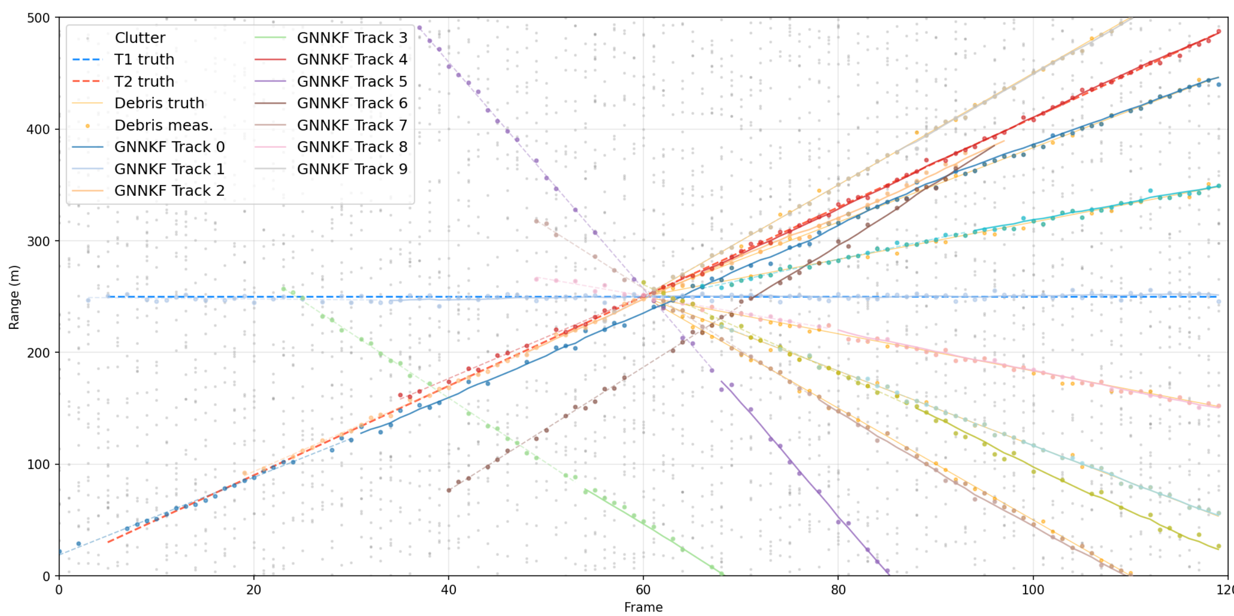

Figure 6.5: Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-GNNKF engine.

RANSAC-GNNKF still tracks most targets and debris children despite the heavier clutter background. Its MOTA drops sharply, reflecting the point at which false track creation outweighs correct confirmations for the RANSAC-based initiator.

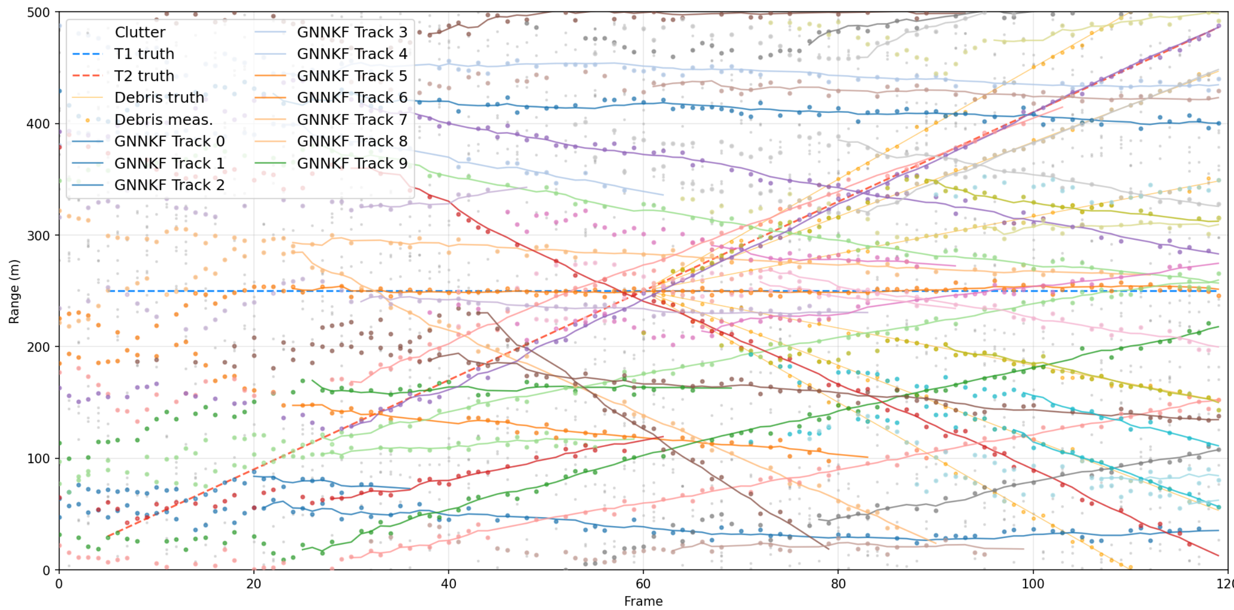

Figure 6.6: Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-GNNKF engine.

MofN-GNNKF collapses under heavy clutter, with dozens of spurious, overlapping false tracks burying the true target trajectories. This produces a false track count that vastly exceeds the total ground truth frame count.

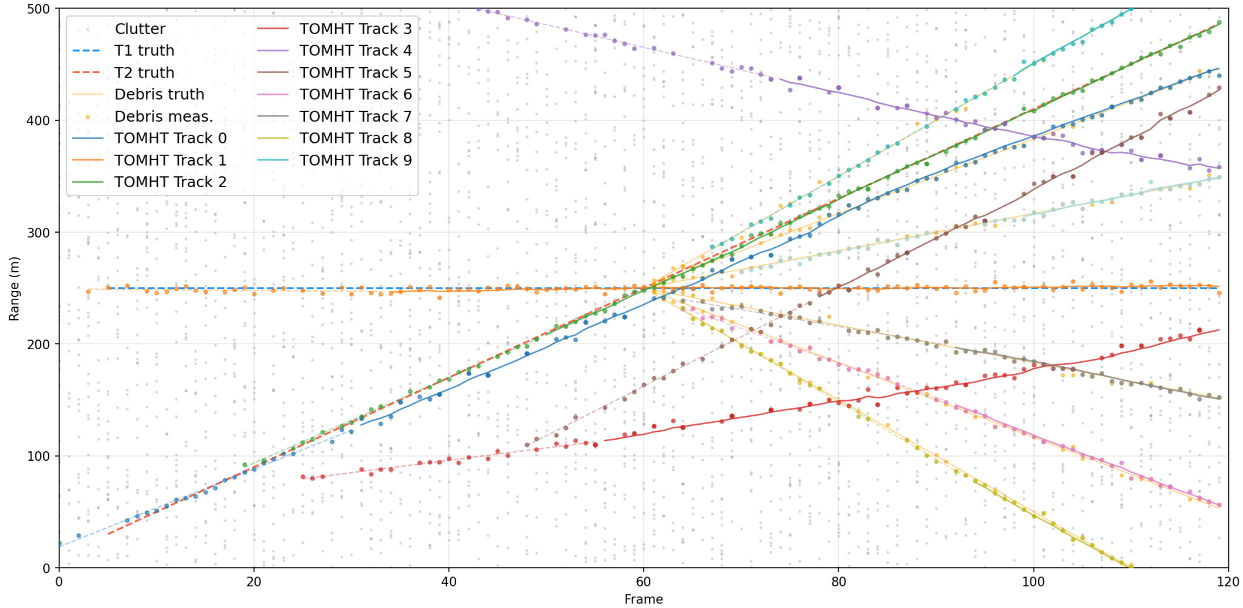

Figure 6.7: Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-TOMHT engine.

RANSAC-TOMHT remains largely clean under the same heavy clutter conditions, with tracks staying close to their truth lines. Its MOTA is the best of all four engines in this condition, the deferred commitment of TOMHT avoids some false associations that GNNKF accepts immediately.

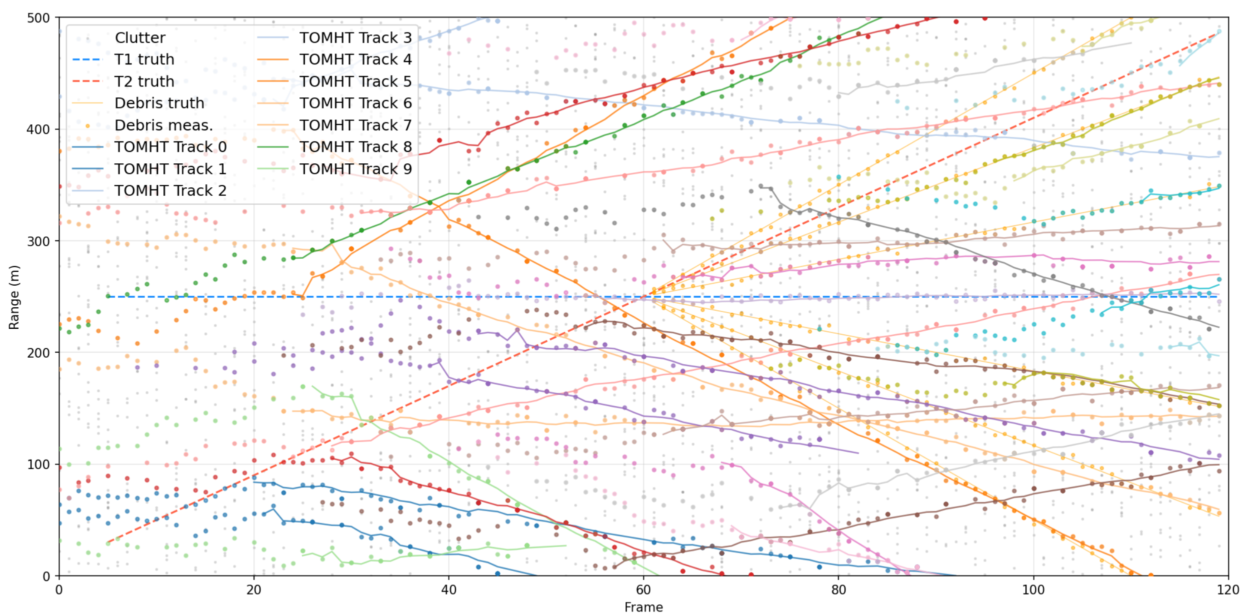

Figure 6.8: Scenario 1 — Simple Intercept, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-TOMHT engine.

MofN-TOMHT also collapses under heavy clutter, with numerous spurious tracks across the plot.

Table 6.2: Tracking Performance for Scenario 1 — Simple Intercept (Clutter Break Conditions: $\lambda = 0.30$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. Large negative MOTA scores correspond to conditions where the false track count significantly exceeded the total ground truth frame count.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	-2.96 ± 20.98	1.4885 ± 0.1647	1.8865 ± 0.1748	0.45 ± 0.46	8.25 ± 0.65
MofN-GNNKF	-432.39 ± 26.68	1.7272 ± 0.1390	2.1473 ± 0.1381	0.84 ± 0.59	0.87 ± 0.22
RANSAC-TOMHT	21.51 ± 21.24	1.4846 ± 0.1663	1.8814 ± 0.1783	2.88 ± 1.29	8.13 ± 0.74
MofN-TOMHT	-244.42 ± 35.15	1.6921 ± 0.1821	2.1338 ± 0.1756	2.53 ± 1.20	2.42 ± 0.64

6.2.3 Dropout Sensitivity

Figures 6.9 through 6.12 show the tracking output under dropout conditions ($\lambda = 0.10$, $p_d = 0.70$), and Table 6.3 summarizes the quantitative results. RANSAC-GNNKF scores 67.50 ± 8.36% MOTA and RANSAC-TOMHT scores 63.33 ± 8.70%, both holding up significantly better than the M-of-N engines under sparse detections. MofN-GNNKF drops to 24.90 ± 14.49% and MofN-TOMHT falls to 25.51 ± 14.06%, reflecting a compounding failure mode specific to the dropout condition. The M-of-N initiator promotes tracks with a zero velocity assumption, meaning the KF prediction after promotion is immediately wrong for any moving target. The Mahalanobis gate is centered at the wrong range, so the next detection may fall outside it and increment the miss counter rather than updating the filter. With $p_d = 0.70$ the tracker is already operating with only 7 out of every 10 detections on average, leaving little margin for gate misses before the miss counter reaches $T_{kill} = 5$ and the track is terminated. RANSAC engines are less affected by this failure mode because WLS provides a velocity estimate at promotion, keeping the gate better aligned with the target from promotion onward. MOTP and RMSE remain lower for both RANSAC engines across this condition, consistent with the velocity initialization advantage persisting through the tracking phase.

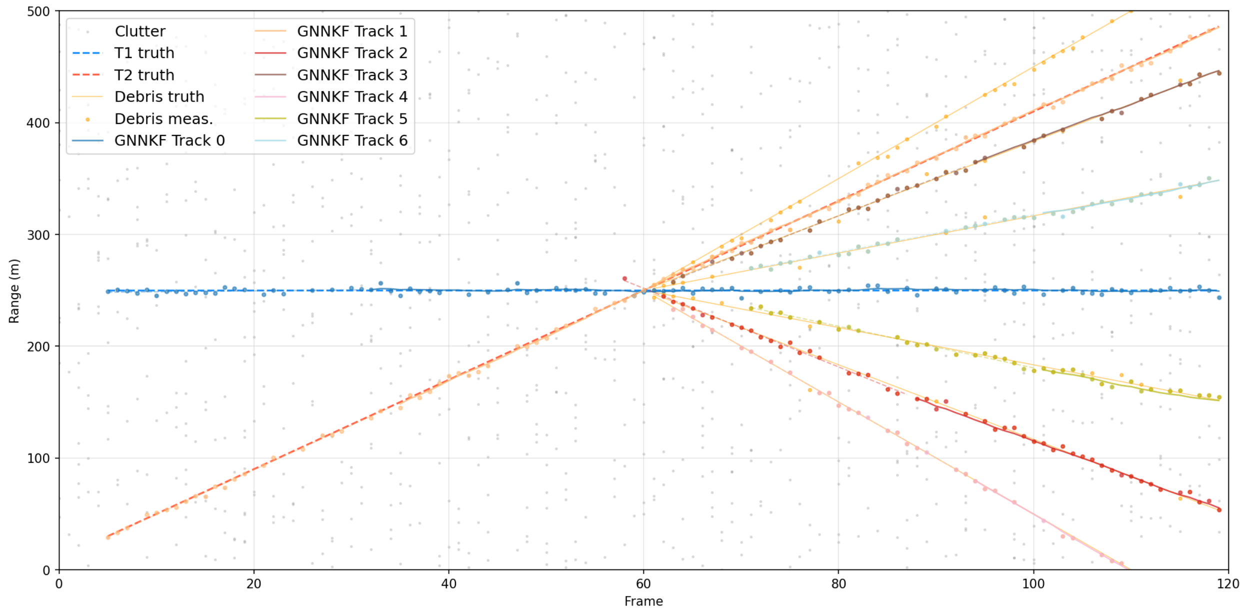

Figure 6.9: Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-GNNKF engine.

RANSAC-GNNKF continues to track all branches cleanly even with sparse detections, closely matching every measurement and truth line. It holds up well, aided by the velocity estimate WLS provides at promotion.

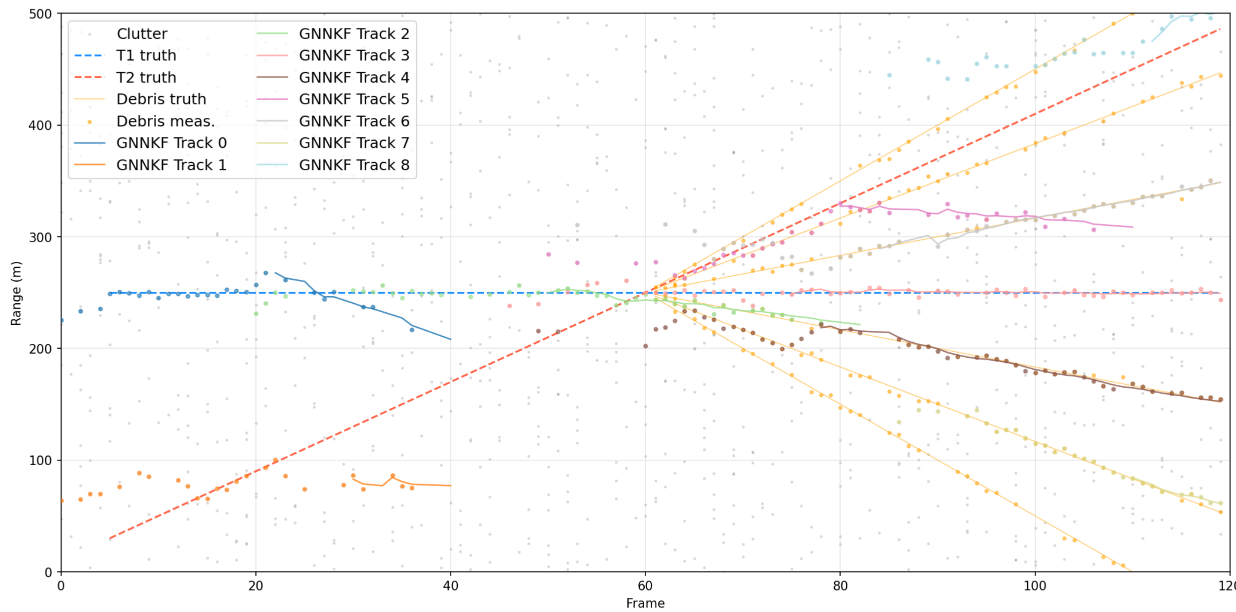

Figure 6.10: Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-GNNKF engine.

MofN-GNNKF shows fragmented and broken tracks, with several debris branches only partially covered by disconnected segments. Its MOTA drops, reflecting the zero-velocity initialization failure mode showing under reduced detection probability.

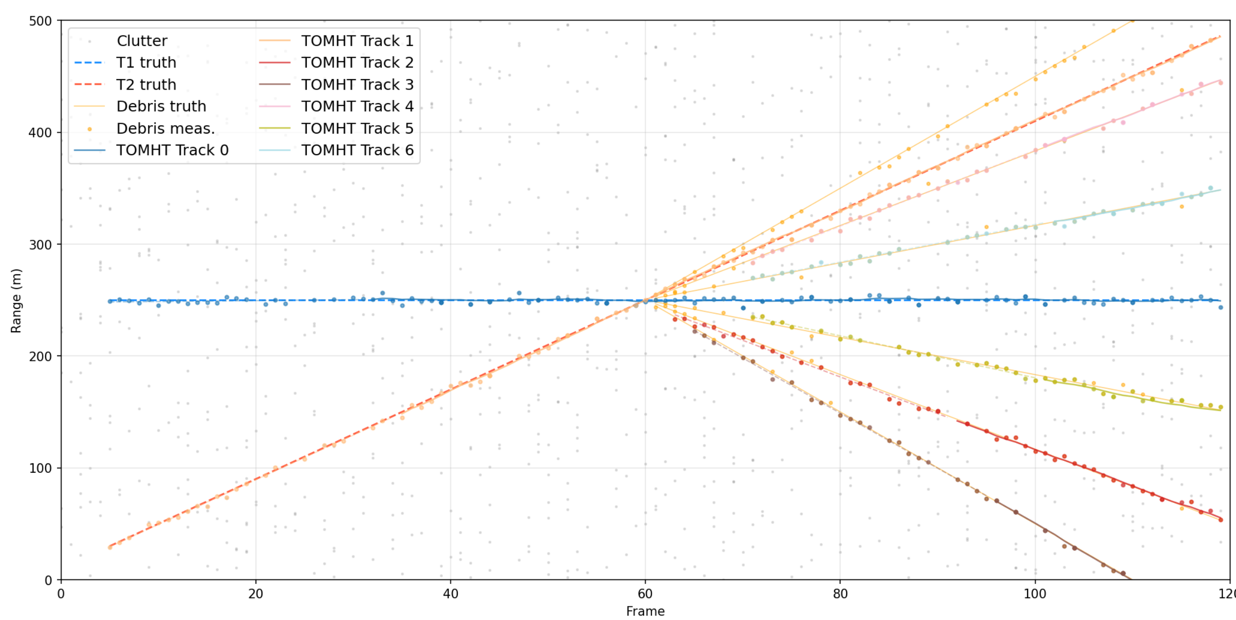

Figure 6.11: Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-TOMHT engine.

RANSAC-TOMHT maintains clean track histories despite the reduced detection probability. Its MOTA is close to RANSAC-GNNKF's, both engines benefiting from the same WLS-derived velocity estimate at promotion.

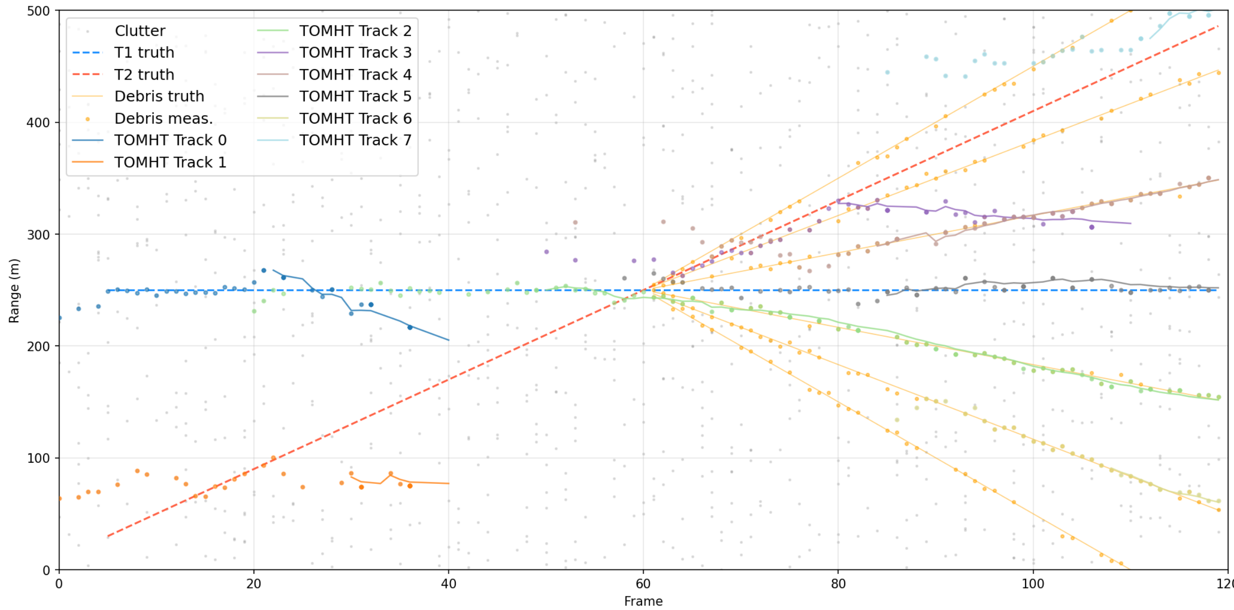

Figure 6.12: Scenario 1 — Simple Intercept, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-TOMHT engine.

MofN-TOMHT exhibits the same fragmented track behavior as MofN-GNNKF under dropout. It also suffers from the zero-velocity failure mode.

Table 6.3: Tracking Performance for Scenario 1 — Simple Intercept (Dropout Break Conditions: $\lambda = 0.10$, $p_d = 0.70$). Results are mean ± std across 1000 Monte Carlo trials.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	67.50 ± 8.36	1.3179 ± 0.1340	1.6801 ± 0.1475	4.11 ± 2.97	2.63 ± 0.09
MofN-GNNKF	24.90 ± 14.49	1.5824 ± 0.1630	1.9899 ± 0.1677	4.65 ± 2.90	0.04 ± 0.01
RANSAC-TOMHT	63.33 ± 8.70	1.3389 ± 0.1420	1.7039 ± 0.1524	6.75 ± 3.41	2.58 ± 0.10
MofN-TOMHT	25.51 ± 14.06	1.5866 ± 0.1653	1.9941 ± 0.1681	6.65 ± 3.27	0.11 ± 0.02

6.3 Scenario 2 Results — Horizontal and Angled

This section presents the tracking performance of all four engines on the Horizontal and Angled scenario. Three conditions are evaluated for this scenario, each varying a single environmental factor from its nominal value while holding everything else fixed. The nominal condition uses a clutter density of $\lambda = 0.10$ and a probability of detection of $p_d = 0.90$. The clutter sensitivity condition raises clutter density to $\lambda = 0.30$ while holding $p_d = 0.90$, stressing the false track suppression capability of each initiator. The dropout sensitivity condition instead lowers probability of detection to $p_d = 0.70$ while holding $\lambda = 0.10$, stressing each engine's ability to maintain a track through missed detections.

6.3.1 Nominal Performance

Figures 6.13 through 6.16 show the tracking output for all four engines under nominal conditions ($\lambda = 0.10$, $p_d = 0.90$), and Table 6.4 summarizes the quantitative results. Scenario 2 adds two persistent flyby targets at 380m and 120m, increasing the number of simultaneously active tracks relative to Scenario 1 and raising the association burden throughout the scenario. RANSAC-GNNKF leads at $83.16 \pm 3.38\%$ MOTA, a small drop from its Scenario 1 nominal performance consistent with the added tracking complexity. MofN-GNNKF scores $51.18 \pm 9.64\%$, notably higher than its Scenario 1 nominal result, likely because the two persistent flyby targets are easy to confirm and maintain, boosting the true positive count relative to the false positive count. RANSAC-TOMHT scores $77.12 \pm 2.35\%$ and MofN-TOMHT scores $48.83 \pm 9.52\%$, both trailing their respective GNNKF counterparts. MOTP and RMSE remain lower for both RANSAC engines, consistent with the pattern established in Scenario 1.

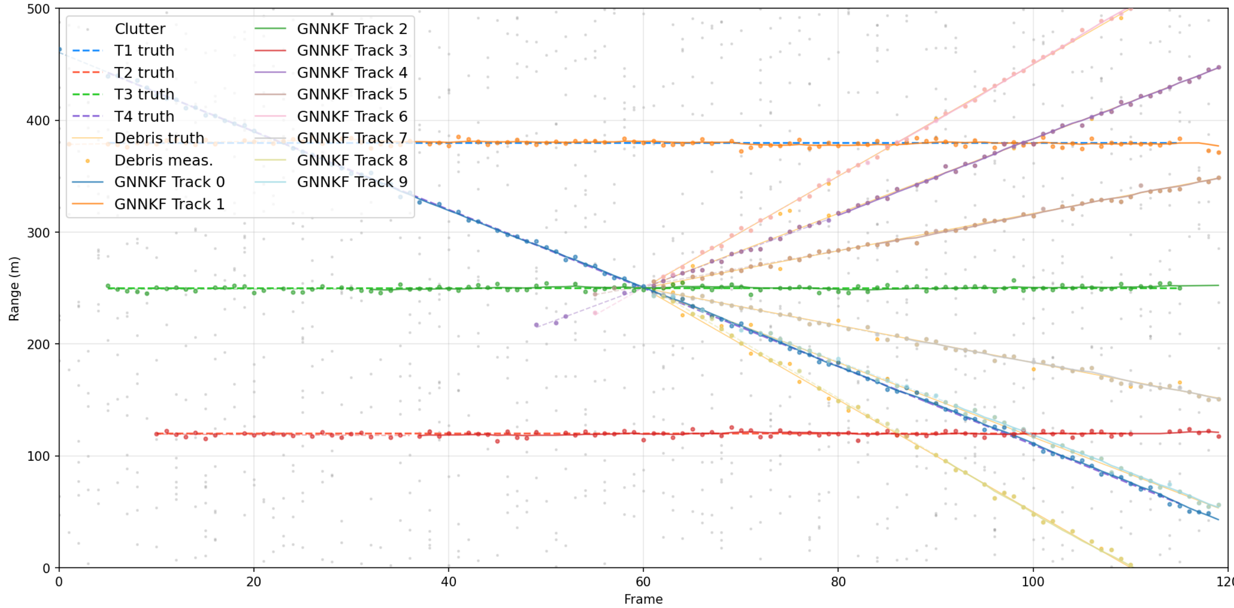

Figure 6.13: Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-GNNKF engine.

RANSAC-GNNKF cleanly tracks all targets and debris children, with estimates overlaying truth.

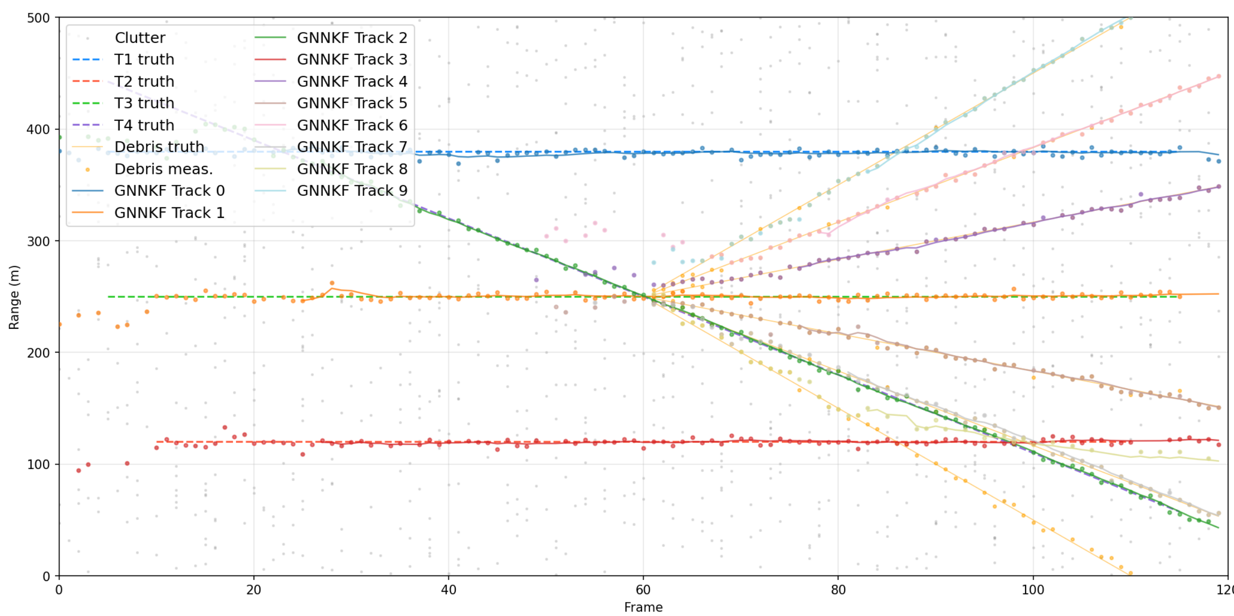

Figure 6.14: Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-GNNKF engine.

MofN-GNNKF mostly follows its measurements but shows minor errors. Its MOTA is higher than Scenario 1, likely boosted by the two easily-confirmed flyby targets.

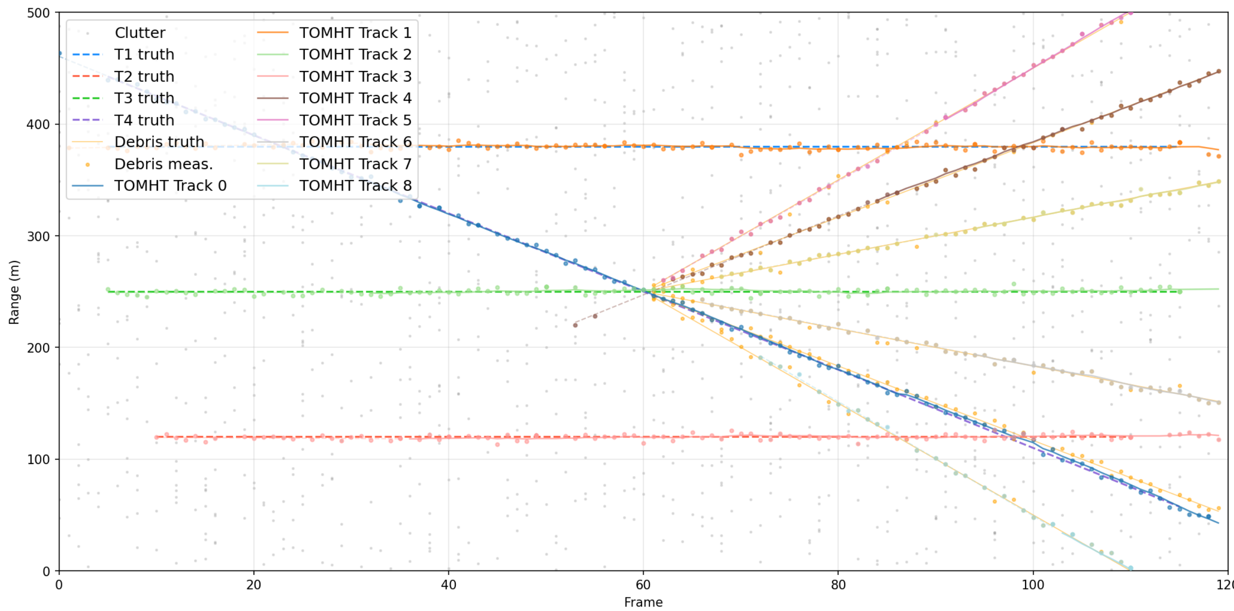

Figure 6.15: Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-TOMHT engine.

RANSAC-TOMHT overlays all tracks with small deviations from truth. Its MOTA trails RANSAC-GNNKF, due to maintaining hypotheses across more targets.

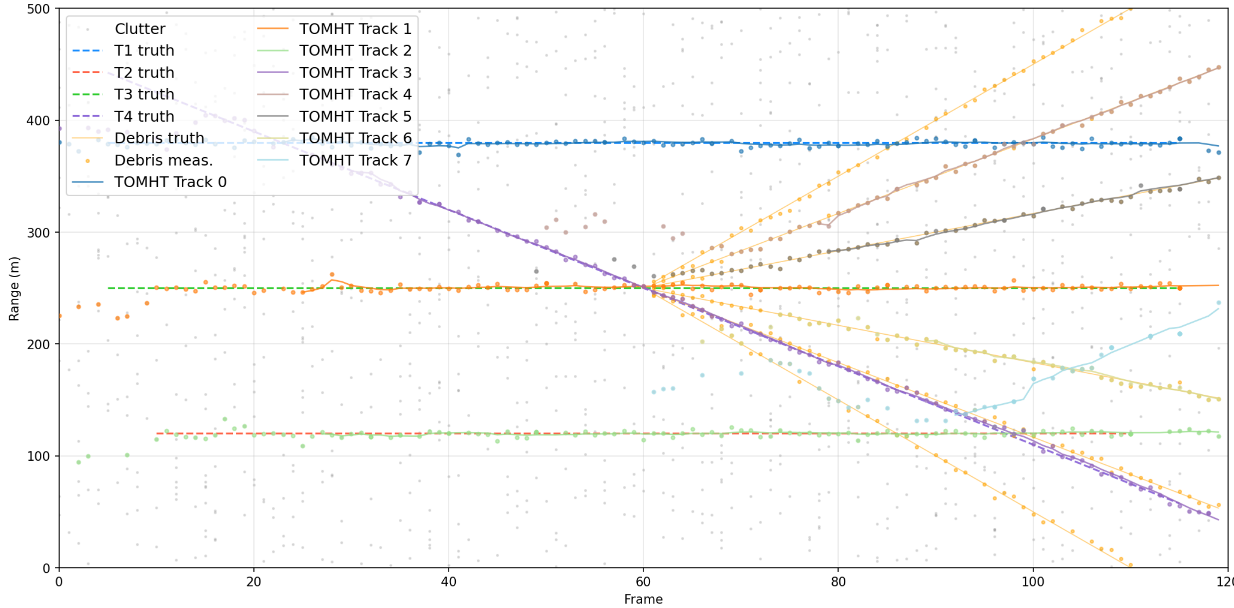

Figure 6.16: Scenario 2 — Horizontal and Angled, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-TOMHT engine.

MofN-TOMHT mostly follows the true trajectories, though one track branches sharply from truth. Its MOTA also benefits from the same flyby targets.

Table 6.4: Tracking Performance for Scenario 2 — Horizontal and Angled (Nominal Conditions: $\lambda = 0.10$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	83.16 ± 3.38	1.1456 ± 0.0943	1.4956 ± 0.1057	0.39 ± 0.41	2.69 ± 0.08
MofN-GNNKF	51.18 ± 9.64	1.3105 ± 0.1078	1.7040 ± 0.1162	2.12 ± 1.18	0.09 ± 0.01
RANSAC-TOMHT	77.12 ± 2.35	1.1515 ± 0.0890	1.4998 ± 0.0978	2.76 ± 1.56	2.56 ± 0.08
MofN-TOMHT	48.83 ± 9.52	1.2948 ± 0.1061	1.6844 ± 0.1159	4.65 ± 2.06	0.28 ± 0.03

6.3.2 Clutter Sensitivity

Figures 6.17 through 6.20 show the tracking output under heavy clutter conditions ($\lambda = 0.30$, $p_d = 0.90$), and Table 6.5 summarizes the quantitative results. Both M-of-N engines again collapse to negative MOTA, consistent with the pattern observed in Scenario 1. RANSAC-GNNKF scores 13.24 ± 16.70% and RANSAC-TOMHT scores 34.87 ± 17.29%, with RANSAC-TOMHT pulling ahead of RANSAC-GNNKF by roughly 22 percentage points. This is the condition where the deferred commitment of the TOMHT engine can potentially provide a measurable advantage, as it can avoid committing to false associations around the collision event that GNNKF locks onto immediately. Both RANSAC engines see significant MOTA degradation relative to nominal, driven by increased false track promotion as clutter density approaches the point where random collinear alignments begin satisfying N_{min}. Runtime increases substantially for both RANSAC engines, reflecting the larger detection pool the incubator must search each scan. MOTP and RMSE remain lower for both RANSAC engines despite the heavy clutter, consistent with the pattern established in Scenario 1.

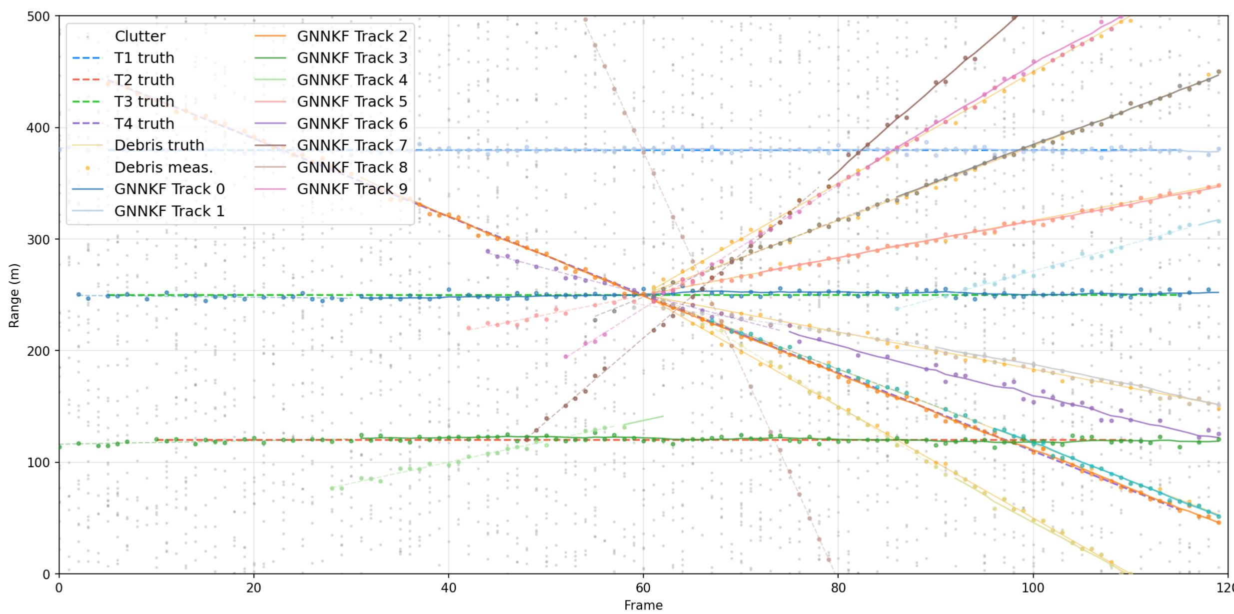

Figure 6.17: Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-GNNKF engine.

RANSAC-GNNKF still overlays all tracks closely despite the heavier clutter background. Its MOTA drops, reflecting increased false track promotion as clutter density approaches the point where random collinear alignments satisfy N_{min}.

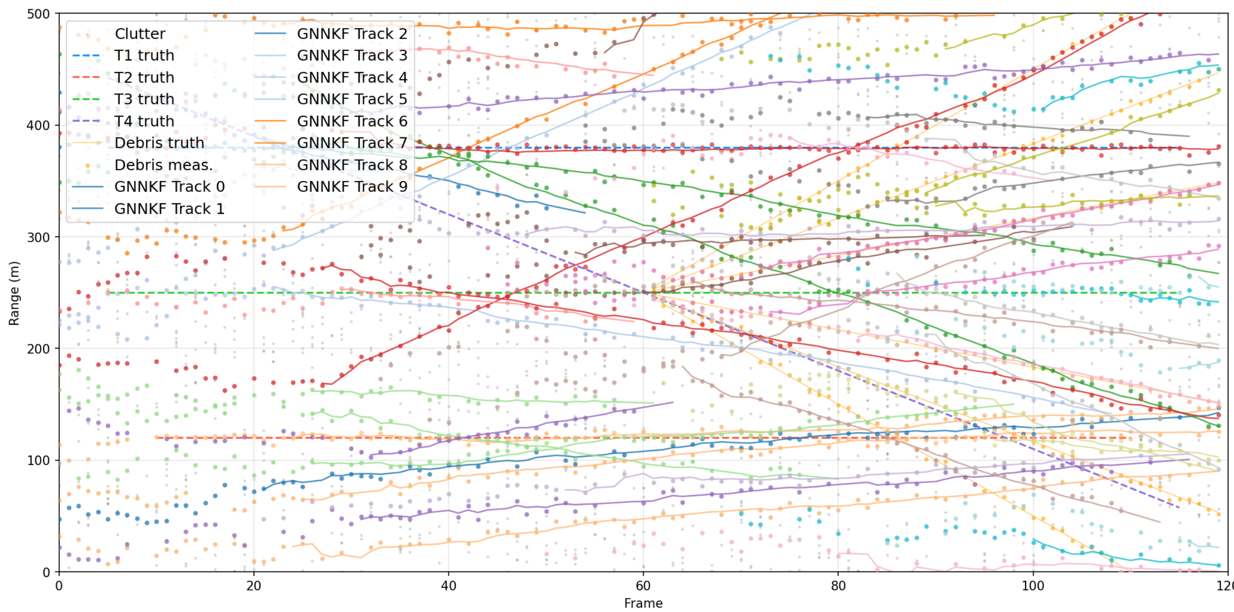

Figure 6.18: Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-GNNKF engine.

MofN-GNNKF completely collapses due to false track proliferation.

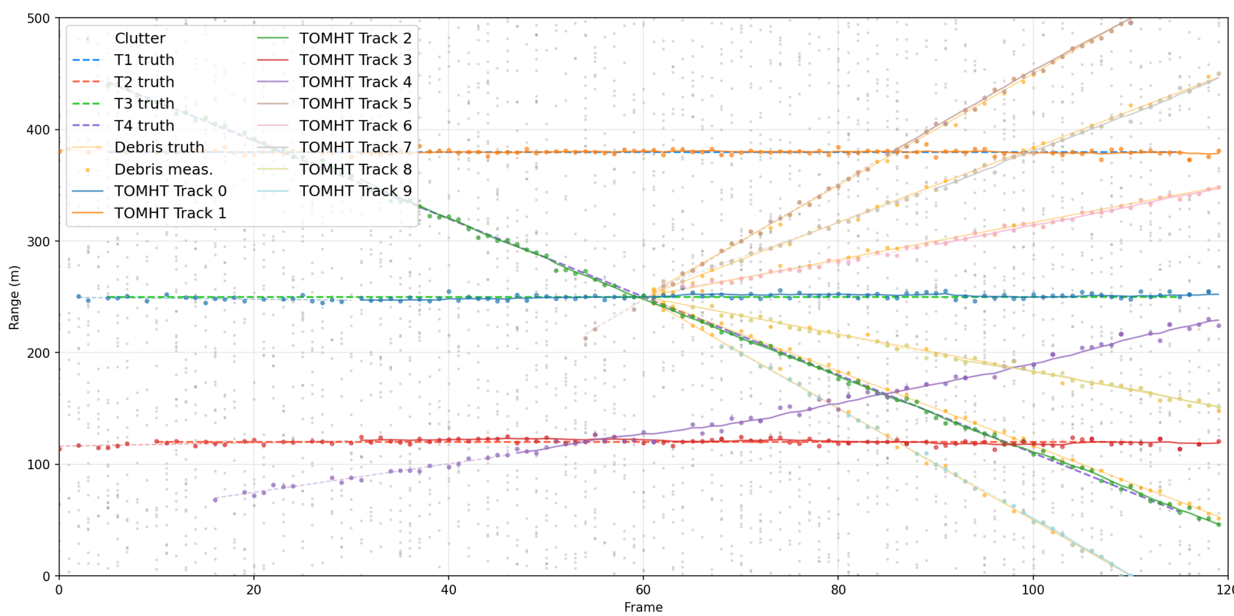

Figure 6.19: Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-TOMHT engine.

RANSAC-TOMHT keeps all tracks closely tracing their truth despite the heavy clutter. Its

MOTA leads the other engines for this scenario.

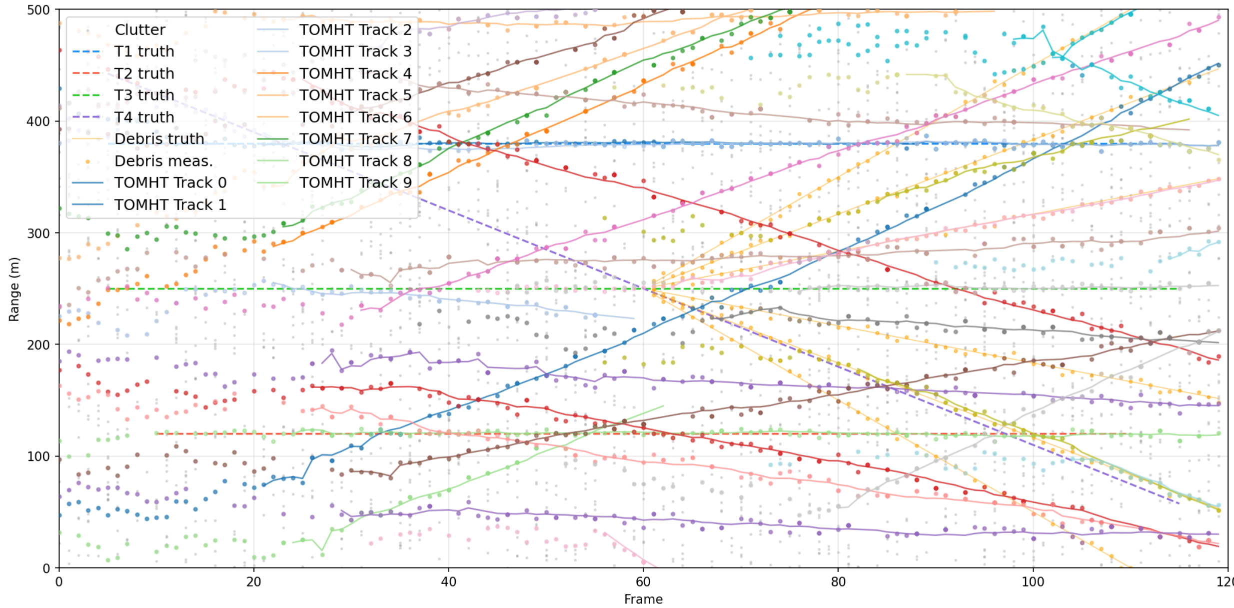

Figure 6.20: Scenario 2 — Horizontal and Angled, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-TOMHT engine.

MofN-TOMHT also collapses into a mass of spurious tracks under heavy clutter.

Table 6.5: Tracking Performance for Scenario 2 — Horizontal and Angled (Clutter Break Conditions: $\lambda = 0.30$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. Large negative MOTA scores correspond to conditions where the false track count significantly exceeded the total ground truth frame count.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	13.24 ± 16.70	1.4405 ± 0.1382	1.8336 ± 0.1515	0.51 ± 0.44	7.67 ± 0.19
MofN-GNNKF	-296.18 ± 20.59	1.6340 ± 0.1166	2.0489 ± 0.1183	0.74 ± 0.46	0.69 ± 0.04
RANSAC-TOMHT	34.87 ± 17.29	1.4081 ± 0.1425	1.7984 ± 0.1562	3.03 ± 1.73	7.23 ± 0.17
MofN-TOMHT	-208.61 ± 22.90	1.5863 ± 0.1378	2.0254 ± 0.1395	2.28 ± 1.02	2.03 ± 0.17

6.3.3 Dropout Sensitivity

Figures 6.21 through 6.24 show the tracking output under dropout conditions ($\lambda = 0.10$, $p_d = 0.70$), and Table 6.6 summarizes the quantitative results. RANSAC-GNNKF scores 62.87 ±

7.08% MOTA and RANSAC-TOMHT scores 57.37 ± 7.09%, both degraded relative to nominal but still significantly ahead of the M-of-N engines. The larger gap between the two RANSAC engines compared to nominal is consistent with RANSAC-TOMHT's detection claiming behavior becoming more costly when detections are already sparse. MofN-GNNKF scores 35.29 ± 10.20% and MofN-TOMHT scores 34.77 ± 10.24%. MOTP and RMSE remain lower for both RANSAC engines across this condition, consistent with the pattern established in Scenario 1.

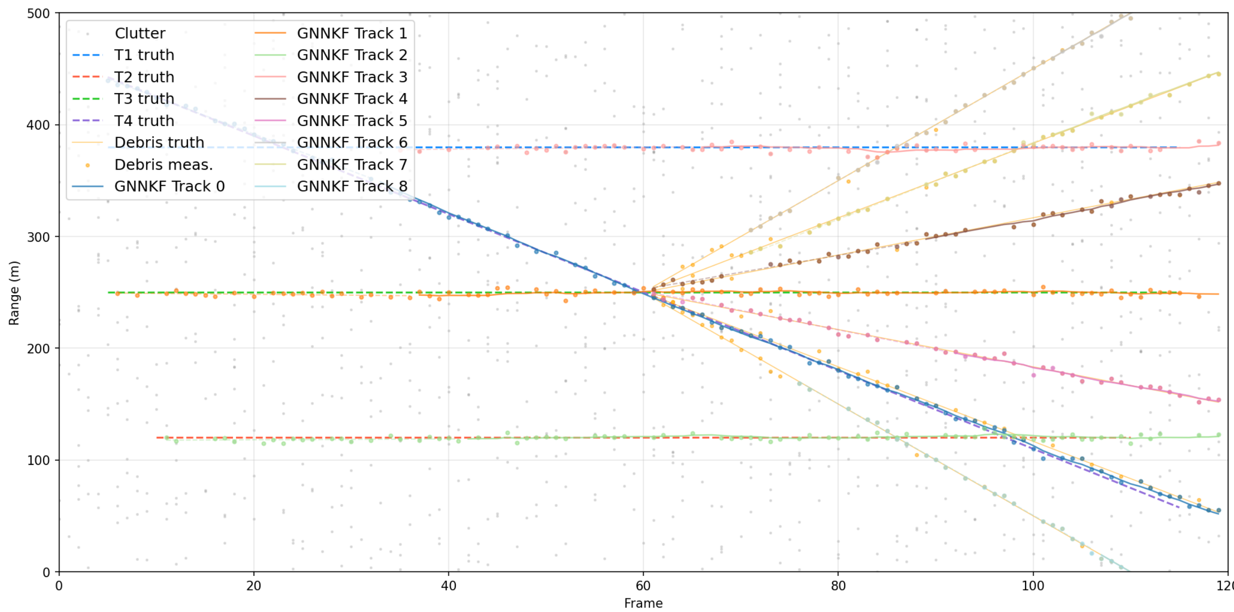

Figure 6.21: Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-GNNKF engine.

RANSAC-GNNKF holds together cleanly despite the reduced detection probability.

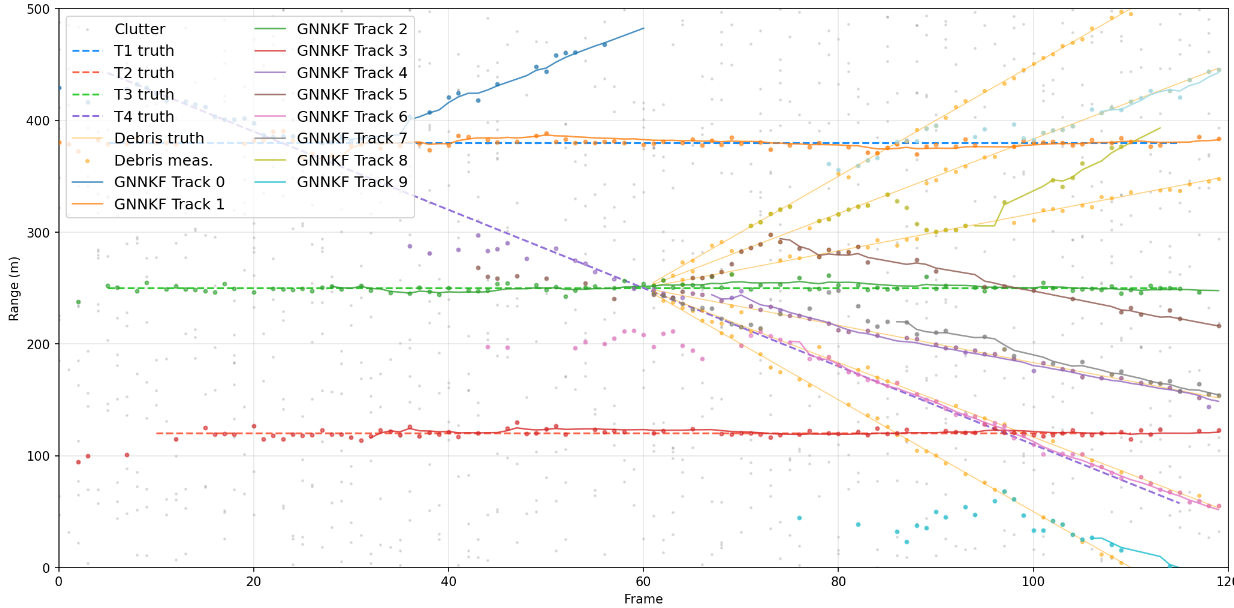

Figure 6.22: Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-GNNKF engine.

MofN-GNNKF shows several false tracks and a wandering segment unrelated to any truth line. Its MOTA is also lower than its RANSAC counterparts, reflecting the same zero-velocity failure mode observed in Scenario 1.

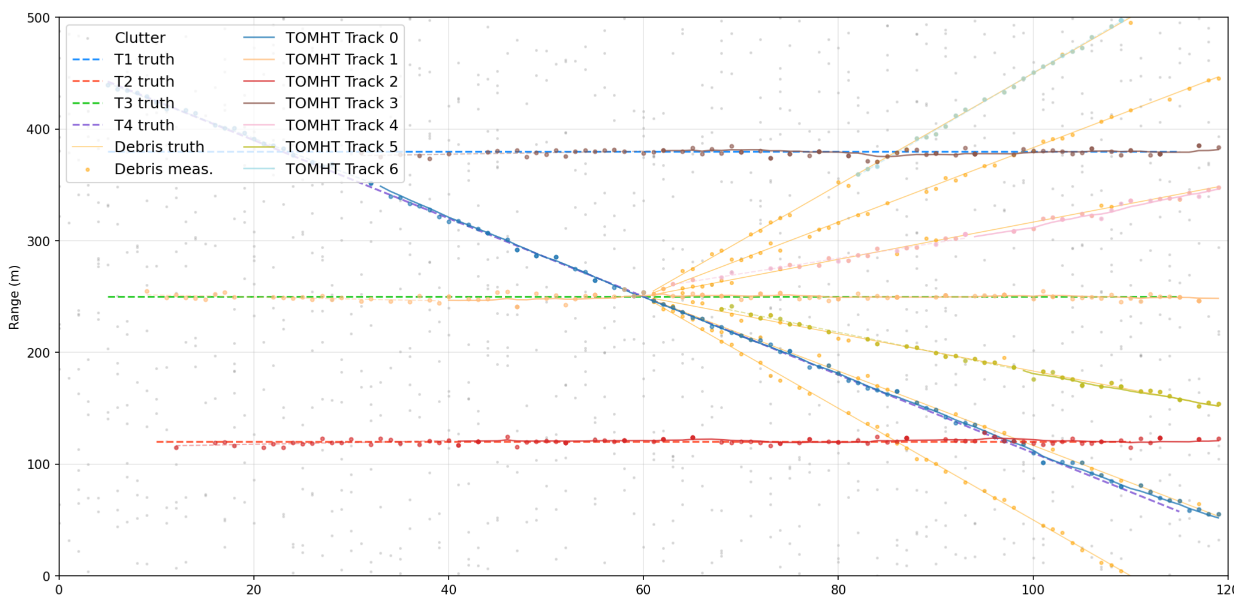

Figure 6.23: Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-TOMHT engine.

RANSAC-TOMHT tracks remain cleanly aligned with truth despite the sparse detections. Its MOTA trails RANSAC-GNNKF only slightly, both benefiting from the same WLS-derived velocity estimate.

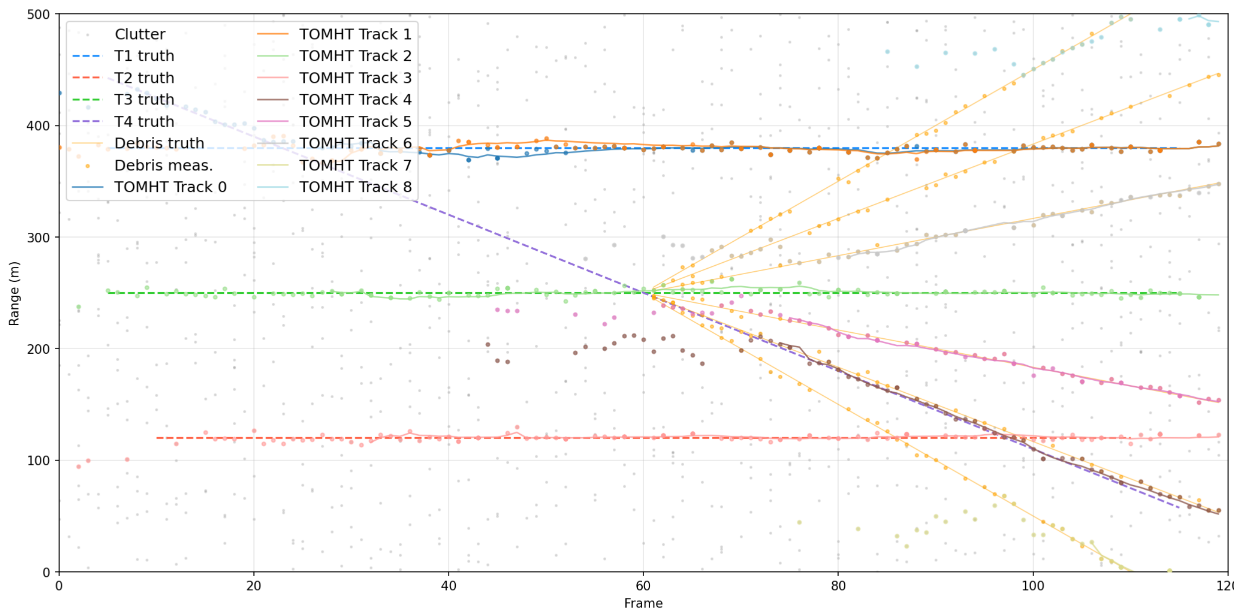

Figure 6.24: Scenario 2 — Horizontal and Angled, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-TOMHT engine.

MofN-TOMHT shows the same noisy false track behavior as MofN-GNNKF under dropout. Its MOTA confirms the zero-velocity failure mode persists regardless of association engine.

Table 6.6: Tracking Performance for Scenario 2 — Horizontal and Angled (Dropout Break Conditions: $\lambda = 0.10$, $p_d = 0.70$). Results are mean ± std across 1000 Monte Carlo trials.

Engine	**MOTA (%)**	**MOTP (m)**	**RMSE (m)**	**TTI (scans)**	**Runtime (s)**
RANSAC-GNNKF	62.87 ± 7.08	1.2964 ± 0.1199	1.6539 ± 0.1313	5.50 ± 3.38	2.72 ± 0.10
MofN-GNNKF	35.29 ± 10.20	1.5104 ± 0.1330	1.9128 ± 0.1379	3.99 ± 2.13	0.06 ± 0.01
RANSAC-TOMHT	57.37 ± 7.09	1.2959 ± 0.1237	1.6528 ± 0.1349	7.81 ± 3.79	2.60 ± 0.11
MofN-TOMHT	34.77 ± 10.24	1.4991 ± 0.1339	1.8987 ± 0.1398	5.90 ± 2.60	0.18 ± 0.02

6.4 Scenario 3 Results — Symmetric Crossing

This section presents the tracking performance of all four engines on the Symmetric Crossing scenario. Three conditions are evaluated for this scenario, each varying a single environmental factor from its nominal value while holding everything else fixed. The nominal condition uses a clutter density of $\lambda = 0.10$ and a probability of detection of $p_d = 0.90$. The clutter sensitivity condition raises clutter density to $\lambda = 0.30$ while holding $p_d = 0.90$, stressing the false track suppression capability of each initiator. The dropout sensitivity condition instead lowers probability of detection to $p_d = 0.70$ while holding $\lambda = 0.10$, stressing each engine's ability to maintain a track through missed detections.

6.4.1 Nominal Performance

Figures 6.25 through 6.28 show the tracking output for all four engines under nominal conditions ($\lambda = 0.10$, $p_d = 0.90$), and Table 6.7 summarizes the quantitative results. Scenario 3 presents the most geometrically challenging configuration of the three, with two targets approaching symmetrically from opposite ends of the range extent and crossing at 250m at $t = 60$s, creating maximum association ambiguity at the collision point. RANSAC-GNNKF scores $82.82 \pm 3.17\%$ MOTA and RANSAC-TOMHT scores $70.89 \pm 2.80\%$, a larger gap between the two RANSAC engines than in previous scenarios. This is consistent with the symmetric crossing geometry concentrating targets and debris at the same range at the collision frame, making the TOMHT detection claiming behavior more costly as more hypothesis branches compete for the same detections simultaneously, leaving less available for the incubator to promote debris tracks. This result is notable because TOMHT's deferred commitment is theoretically better suited to ambiguous crossing scenarios, yet the detection claiming cost appears to outweigh that benefit here, resulting in lower MOTA than GNNKF despite the increased association complexity. This finding warrants future work on returning pruned branch detections to the incubator pool after the N-scan commit point, discussed further in Section 7.2. MofN-GNNKF scores $49.22 \pm 10.33\%$ and MofN-TOMHT scores $46.98 \pm 9.27\%$, both lower than their Scenario 2 nominal results, reflecting the added difficulty of the crossing geometry. MOTP and RMSE remain lower for both RANSAC engines, consistent with the pattern established across previous scenarios.

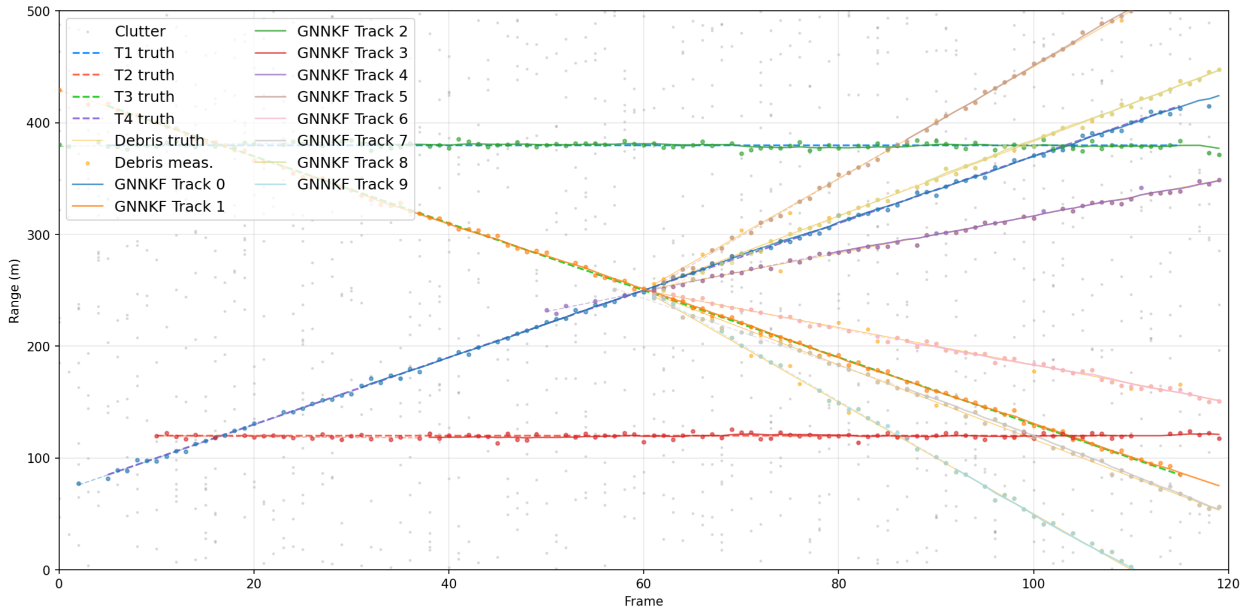

Figure 6.25: Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-GNNKF engine.

RANSAC-GNNKF cleanly resolves all tracks through the symmetric crossing, overlaying each truth line. Its MOTA holds up well despite the maximum association ambiguity of this scenario's geometry.

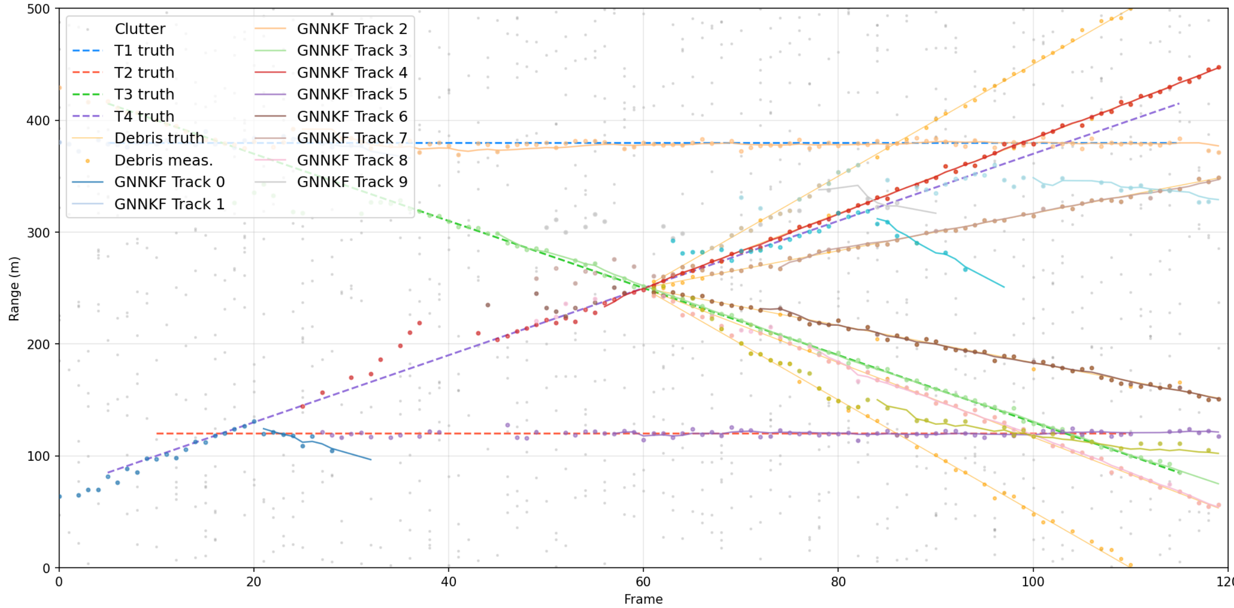

Figure 6.26: Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-GNNKF engine.

MofN-GNNKF shows fragmented tracks and wandering segments unrelated to any true target. Its MOTA is its lowest nominal result across all three scenarios, reflecting the added difficulty of the crossing geometry.

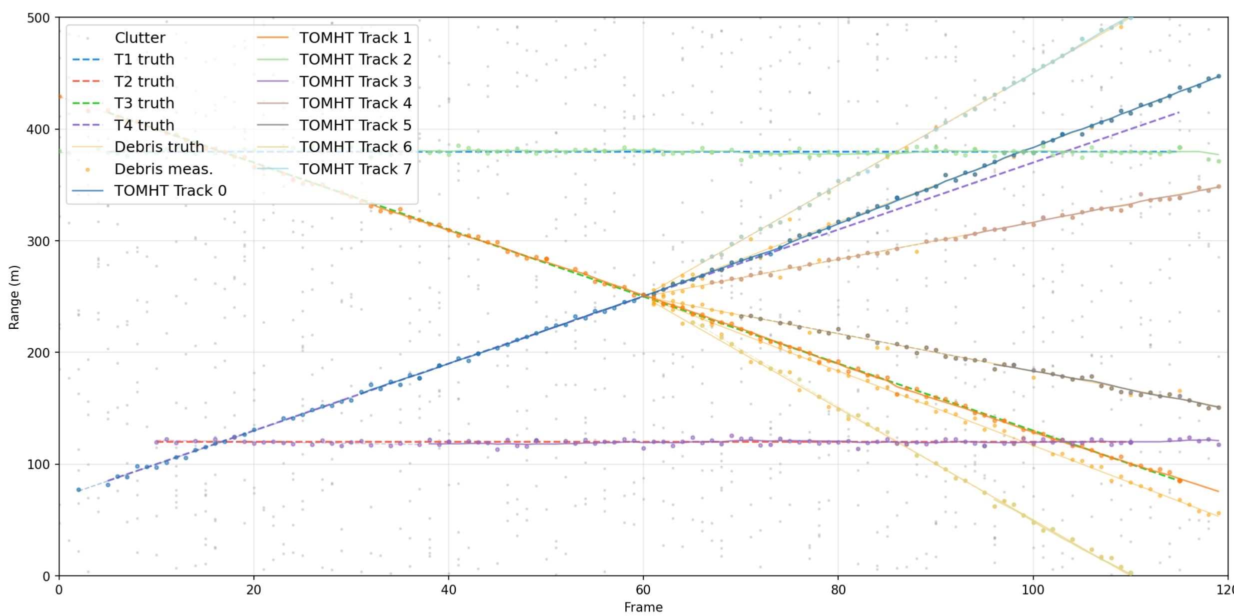

Figure 6.27: Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). RANSAC-TOMHT engine.

RANSAC-TOMHT closely overlays its truth. Its MOTA trails RANSAC-GNNKF by a decent margin due to the symmetric crossing increased claiming costs at the collision frame.

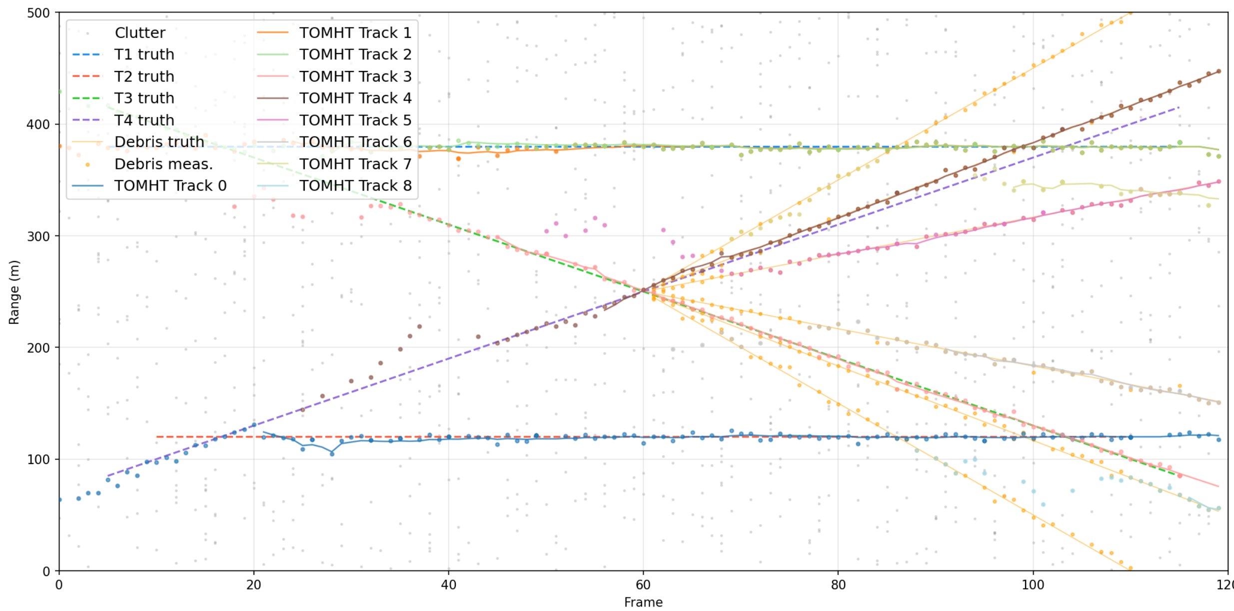

Figure 6.28: Scenario 3 — Symmetric Crossing, nominal conditions ($\lambda = 0.10$, $p_d = 0.90$). MofN-TOMHT engine.

MofN-TOMHT tracks the crossing targets reasonably well aside from a brief wobble in one track before it settles onto truth.

Table 6.7: Tracking Performance for Scenario 3 — Symmetric Crossing (Nominal Conditions: $\lambda = 0.10$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials.

Engine	**MOTA (%)**	**MOTP (m)**	**RMSE (m)**	**TTI (scans)**	**Runtime (s)**
RANSAC-GNNKF	82.82 ± 3.17	1.1770 ± 0.0987	1.5330 ± 0.1137	0.41 ± 0.43	2.67 ± 0.08
MofN-GNNKF	49.22 ± 10.33	1.3399 ± 0.1094	1.7359 ± 0.1195	1.84 ± 1.06	0.09 ± 0.01
RANSAC-TOMHT	70.89 ± 2.80	1.1669 ± 0.0983	1.5223 ± 0.1108	3.34 ± 1.76	2.53 ± 0.08
MofN-TOMHT	46.98 ± 9.27	1.3151 ± 0.1073	1.7086 ± 0.1186	5.56 ± 2.54	0.27 ± 0.02

6.4.2 Clutter Sensitivity

Figures 6.29 through 6.32 show the tracking output under heavy clutter conditions ($\lambda = 0.30$, $p_d = 0.90$), and Table 6.8 summarizes the quantitative results. Both M-of-N engines again collapse to negative MOTA, consistent with the pattern observed across all three scenarios. RANSAC-TOMHT scores $32.66 \pm 17.24\%$ and RANSAC-GNNKF scores $12.65 \pm 16.61\%$, with RANSAC-TOMHT pulling ahead by roughly 16 percentage points, consistent with the deferred commitment of the TOMHT engine providing benefit under heavy clutter where avoiding false associations at the collision frame can preserve multiple debris tracks. MOTP and RMSE remain lower for both RANSAC engines, consistent with the pattern established across previous scenarios.

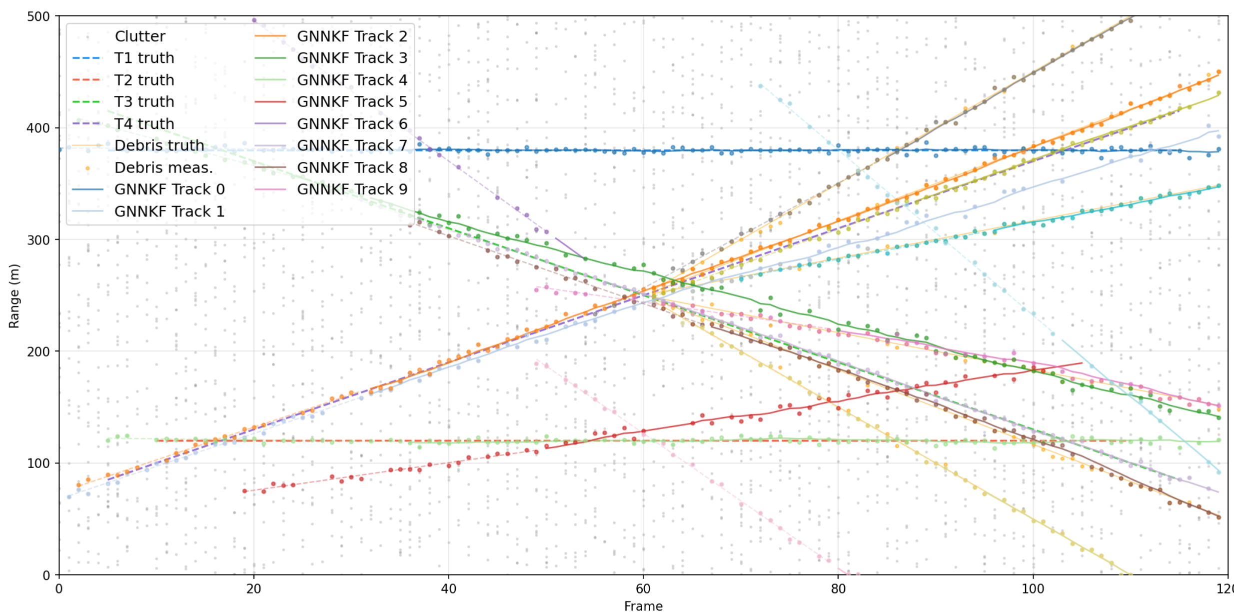

Figure 6.29: Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-GNNKF engine.

RANSAC-GNNKF tracks are noisier under heavy clutter, including various spurious tracks.

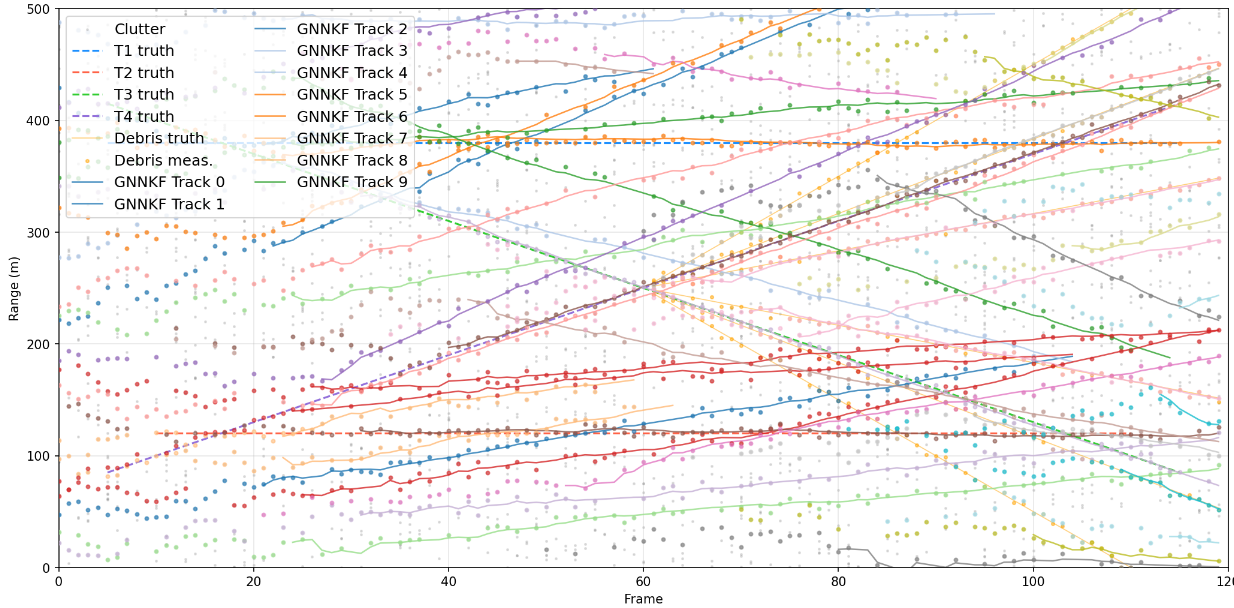

Figure 6.30: Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-GNNKF engine.

MofN-GNNKF collapses into overlapping false tracks.

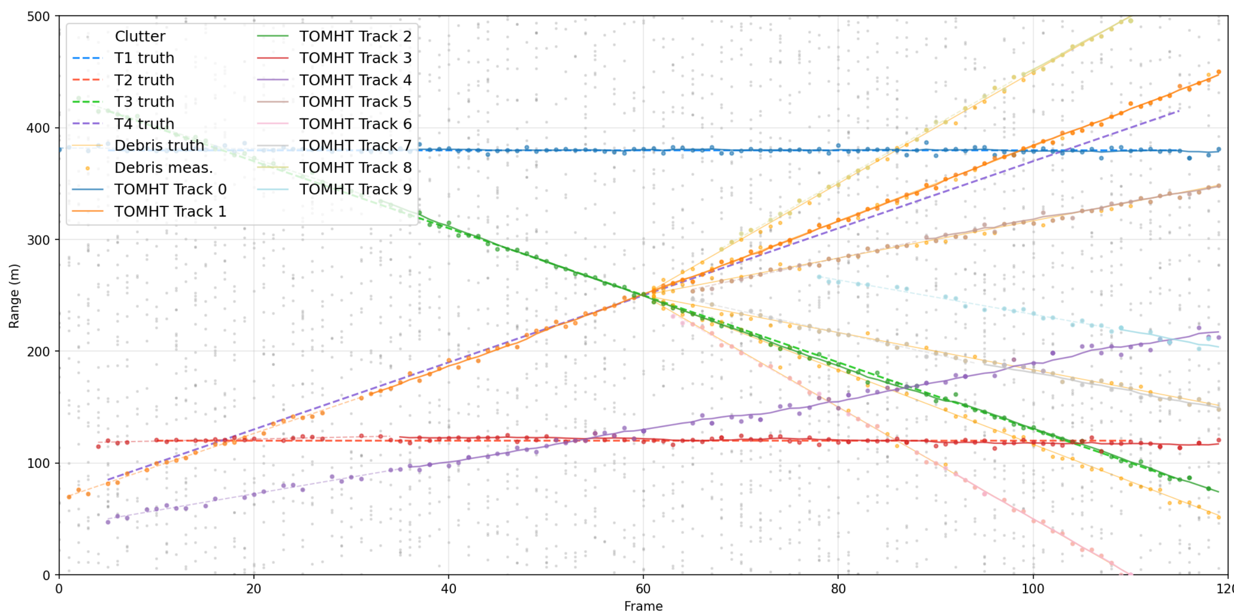

Figure 6.31: Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). RANSAC-TOMHT engine.

RANSAC-TOMHT tracks stay clean despite the elevated clutter.

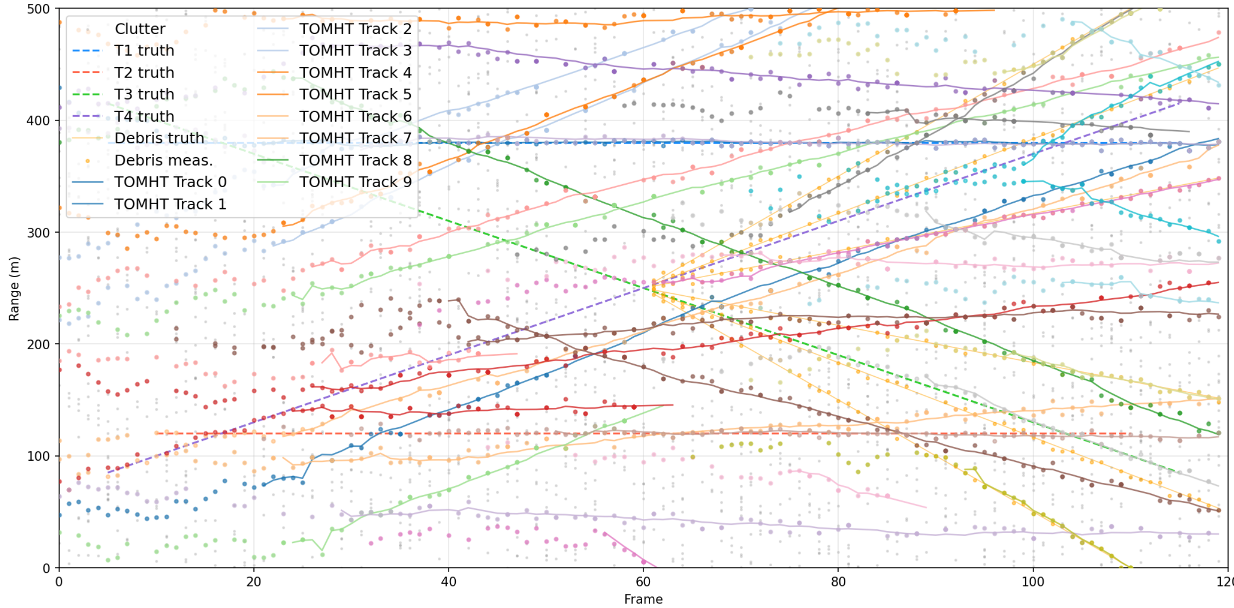

Figure 6.32: Scenario 3 — Symmetric Crossing, clutter break conditions ($\lambda = 0.30$, $p_d = 0.90$). MofN-TOMHT engine.

MofN-TOMHT collapses into numerous spurious and fragmented tracks.

Table 6.8: Tracking Performance for Scenario 3 — Symmetric Crossing (Clutter Break Conditions: $\lambda = 0.30$, $p_d = 0.90$). Results are mean ± std across 1000 Monte Carlo trials. Large negative MOTA scores correspond to conditions where the false track count significantly exceeded the total ground truth frame count.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	12.65 ± 16.61	1.4667 ± 0.1439	1.8622 ± 0.1583	0.53 ± 0.45	8.64 ± 0.61
MofN-GNNKF	-297.34 ± 21.60	1.6489 ± 0.1190	2.0660 ± 0.1220	0.72 ± 0.45	0.98 ± 0.22
RANSAC-TOMHT	32.66 ± 17.24	1.4197 ± 0.1461	1.8124 ± 0.1617	4.01 ± 2.17	8.38 ± 0.71
MofN-TOMHT	-210.51 ± 23.04	1.6028 ± 0.1390	2.0438 ± 0.1390	2.34 ± 1.05	2.91 ± 0.63

6.4.3 Dropout Sensitivity

Figures 6.33 through 6.36 show the tracking output under dropout conditions ($\lambda = 0.10$, $p_d = 0.70$), and Table 6.9 summarizes the quantitative results. RANSAC-GNNKF scores $60.91 \pm 6.95\%$ MOTA and RANSAC-TOMHT scores $53.28 \pm 5.74\%$, both degraded relative to nominal and consistent with the dropout pattern established in previous scenarios. MofN-GNNKF scores $32.65 \pm 10.66\%$ and MofN-TOMHT scores $31.65 \pm 10.38\%$, with MofN-GNNKF outperforming MofN-TOMHT. The zero velocity initialization failure mode remains the dominant driver of M-of-N degradation under dropout regardless of engine. MOTP and RMSE remain lower for both RANSAC engines, consistent with the pattern established across previous scenarios.

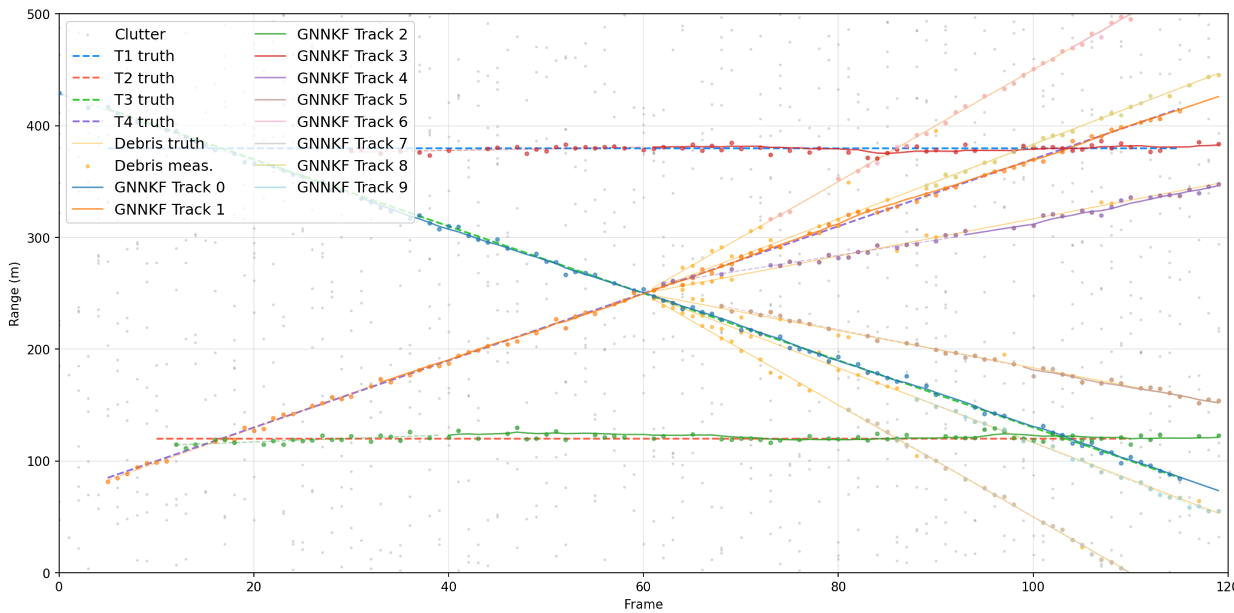

Figure 6.33: Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-GNNKF engine.

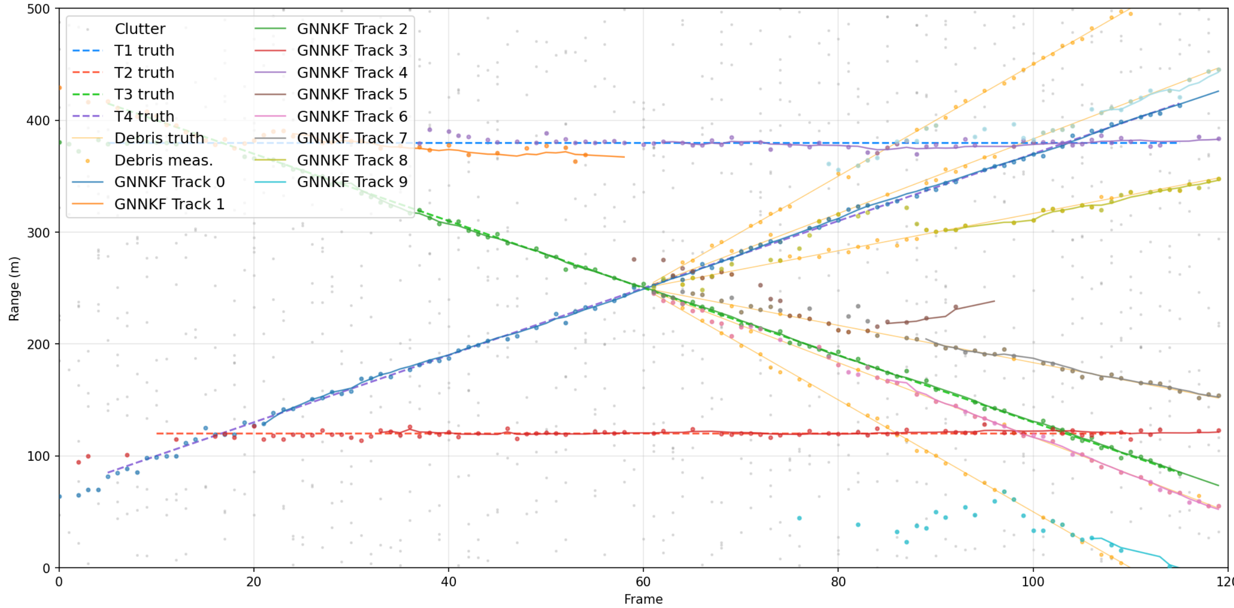

Figure 6.34: Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-GNNKF engine.

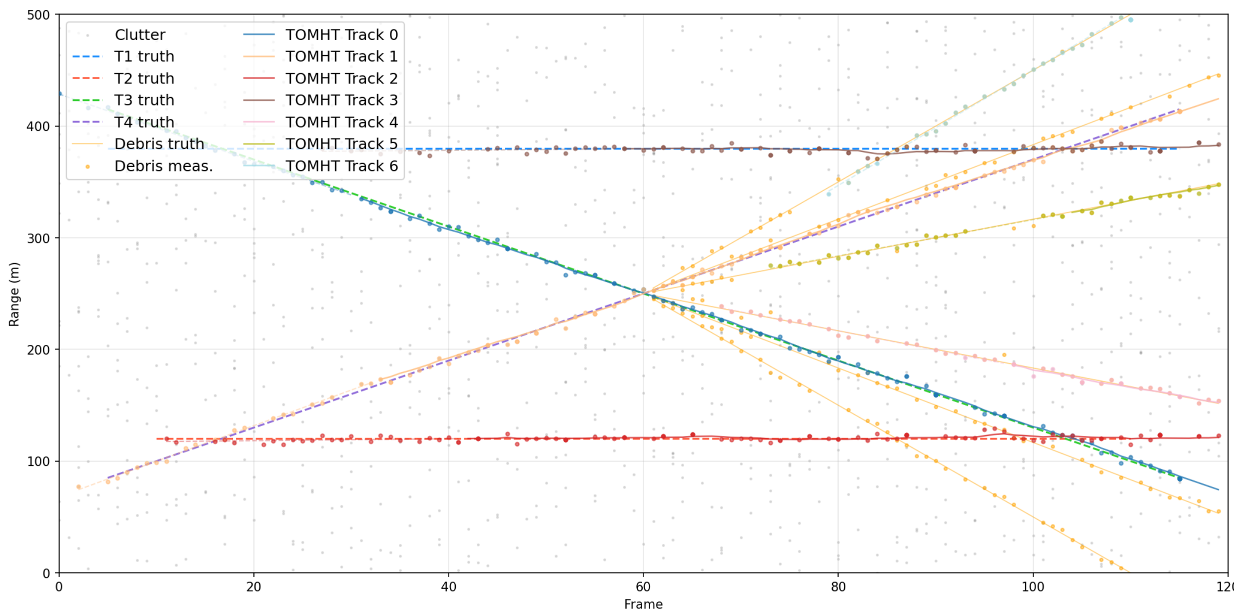

Figure 6.35: Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). RANSAC-TOMHT engine.

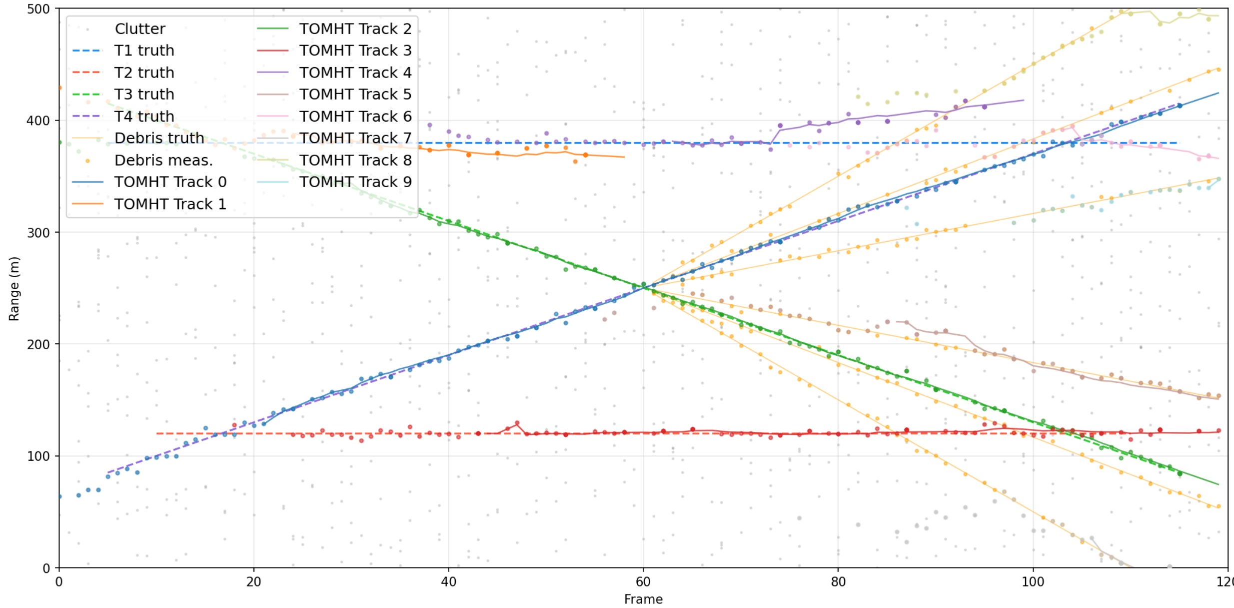

Figure 6.36: Scenario 3 — Symmetric Crossing, dropout break conditions ($\lambda = 0.10$, $p_d = 0.70$). MofN-TOMHT engine.

Table 6.9: Tracking Performance for Scenario 3 — Symmetric Crossing (Dropout Break Conditions: $\lambda = 0.10$, $p_d = 0.70$). Results are mean ± std across 1000 Monte Carlo trials.

Engine	MOTA (%)	MOTP (m)	RMSE (m)	TTI (scans)	Runtime (s)
RANSAC-GNNKF	60.91 ± 6.95	1.3363 ± 0.1250	1.7055 ± 0.1405	4.41 ± 2.79	2.81 ± 0.18
MofN-GNNKF	32.65 ± 10.66	1.5480 ± 0.1386	1.9531 ± 0.1438	3.16 ± 1.80	0.07 ± 0.01
RANSAC-TOMHT	53.28 ± 5.74	1.3367 ± 0.1336	1.7044 ± 0.1498	6.56 ± 3.22	2.66 ± 0.15
MofN-TOMHT	31.65 ± 10.38	1.5367 ± 0.1376	1.9398 ± 0.1446	5.65 ± 2.81	0.18 ± 0.03

6.5 WLS Initialization Validation

This section validates the WLS initialization covariance derived in Section 3 by comparing the theoretical covariance $\mathbf{P}_{kin}$ against the actual spread of estimation errors observed across many independent Monte Carlo trials. Two versions of the WLS estimator are evaluated side by side. The first, referred to simply as WLS, is fed the RANSAC inlier set exactly as produced by the incubation framework, which may include clutter points that slipped past the inlier threshold τ

or exclude genuine target detections that fell just outside it. The second, referred to as Oracle WLS, is fed the true target detections directly, with all clutter and misclassified points removed by construction. Oracle WLS is not a method that could be run in practice, since it requires perfect knowledge of which detections are genuine, but it serves as a controlled reference point. By comparing WLS against Oracle WLS, any overconfidence observed in the standard WLS covariance can be attributed specifically to contamination of the RANSAC inlier set, rather than to a flaw in the WLS formulation itself.

6.5.1 Monte Carlo Study Design

To validate the WLS initialization covariance derived in Section 3, a Monte Carlo study was conducted on the collision with debris scenario described in Section 5.7.4. A static ground truth was generated once and held fixed across all runs. 1,000 independent measurement sets were then generated by applying independent noise and clutter to the fixed ground truth, producing 1,000 statistically independent tracking scenarios from the same underlying target trajectories.

The RANSAC incubator was run on each measurement set independently using the two evaluation windows defined in Section 5.7.4, yielding 13,000 promoted track estimates across both windows and all runs. For each promoted track the WLS state estimate $\hat{\mathbf{x}}_{init}$ and covariance $\mathbf{P}_{kin}$ were recorded at the promotion frame and compared against the true target state at that frame. The empirical error covariance was computed across all runs as:

$$\mathbf{P}_{actual} = \frac{1}{N_{tracks}} \sum_{i=1}^{N_{tracks}} \mathbf{e}_i \mathbf{e}_i^T, \quad \mathbf{e}_i = \hat{\mathbf{x}}_{init,i} - \mathbf{x}_{true,i} \tag{6.4}$$

and compared against the mean theoretical covariance $\bar{\mathbf{P}}_{kin}$ averaged across all promoted tracks. The ratio of actual to theoretical diagonals gives the covariance deflation factor, where a value of 1 indicates a perfectly calibrated covariance and a value greater than 1 indicates the filter is initialized overconfident, meaning the theoretical covariance understates the true estimation uncertainty.

An Oracle WLS baseline was computed in parallel by replacing the RANSAC inlier set with the true target detections only, with all clutter removed. This provides a lower bound on covariance deflation achievable by WLS given perfect data, and serves as a reference to isolate the contribution of clutter contamination from any flaw in the WLS formulation itself. Figure 6.37 shows the RANSAC fits for both evaluation windows from a representative single run.

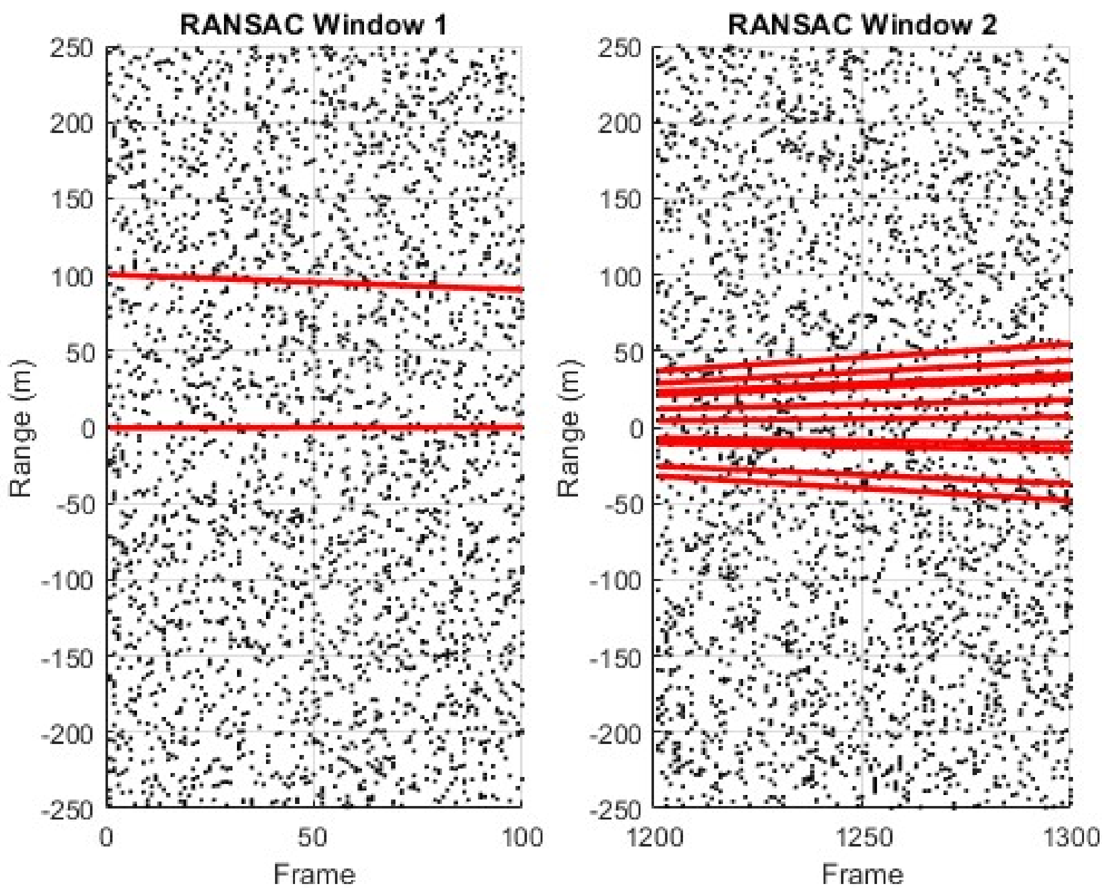

Figure 6.37: RANSAC line fits (red) for a representative Monte Carlo run in both evaluation windows. Window 1 shows two clean tracks in heavy clutter. Window 2 shows the RANSAC incubator successfully recovering multiple debris trajectories from the post-collision detection pool.

To examine the geometric contribution to overconfidence, the mean inverse observation geometry matrix $(\mathbf{H}^T\mathbf{H})^{-1}$ was computed for each method across all promoted tracks. Since $\mathbf{P}_{kin} = \sigma_r^2(\mathbf{H}^T\mathbf{H})^{-1}$, the shape and scale of the theoretical covariance are determined entirely by this term. A truth geometry matrix was computed for each promoted track by constructing $\mathbf{H}$ directly from the ground truth detection frames of the matched target within the evaluation window, with no RANSAC involvement. This gives the geometry matrix that WLS would see under perfect data and serves as an empirical reference against which the contaminated inlier geometry can be compared. Table 6.10 reports the mean $(\mathbf{H}^T\mathbf{H})^{-1}$ for each method.

Table 6.10: Mean inverse observation geometry matrix $(\mathbf{H}^T\mathbf{H})^{-1}$ averaged across all 13,000 promoted tracks. The Standard RANSAC $(\mathbf{H}^T\mathbf{H})^{-1}$ is systematically larger than truth, reflecting a modest inflation of the theoretical covariance from inlier set contamination. The Oracle WLS $(\mathbf{H}^T\mathbf{H})^{-1}$ matches truth exactly, confirming that the resulting overconfidence is driven by inlier set quality rather than any deficiency in the WLS formulation.

Method	$(\mathbf{H}^T\mathbf{H})^{-1}$
Truth	$\begin{bmatrix} 0.0394 & 0.1187 \\ 0.1187 & 0.4798 \end{bmatrix}$
Standard RANSAC	$\begin{bmatrix} 0.0486 & 0.1482 \\ 0.1482 & 0.6032 \end{bmatrix}$
Oracle WLS	$\begin{bmatrix} 0.0394 & 0.1187 \\ 0.1187 & 0.4798 \end{bmatrix}$

The Standard RANSAC $(\mathbf{H}^T\mathbf{H})^{-1}$ is systematically larger than truth across both diagonal entries. Clutter points that fall within the inlier threshold τ are admitted to the inlier set while genuine target detections that fall just outside τ are excluded, producing an inlier set that is artificially tight relative to the true observation geometry. This deflates $\mathbf{H}^T\mathbf{H}$ and correspondingly inflates its inverse, which in turn modestly inflates the theoretical covariance $\mathbf{P}_{kin}$ relative to Oracle WLS. This geometric effect accounts for only a small fraction of the overconfidence observed empirically. The dominant cause is instead that some admitted inliers are clutter points that happen to be colinear with the fitted line, introducing estimation error that $\mathbf{P}_{kin}$ cannot capture. This is primarily due to the use of a fixed σ_r, which is applied uniformly to every inlier and has no way to distinguish genuine detections from contaminated ones. The Oracle WLS

$(\mathbf{H}^T\mathbf{H})^{-1}$ matches truth to within numerical precision, confirming that the WLS formulation is correct and that overconfidence is a consequence of inlier set contamination rather than any deficiency in the estimator.

6.5.2 Covariance Deflation Results

Figure 6.38 shows the theoretical and actual 99% confidence covariance ellipses in position and velocity error space for both WLS and Oracle WLS across all 13,000 track promotions. Table 6.11 summarizes the covariance deflation factors.

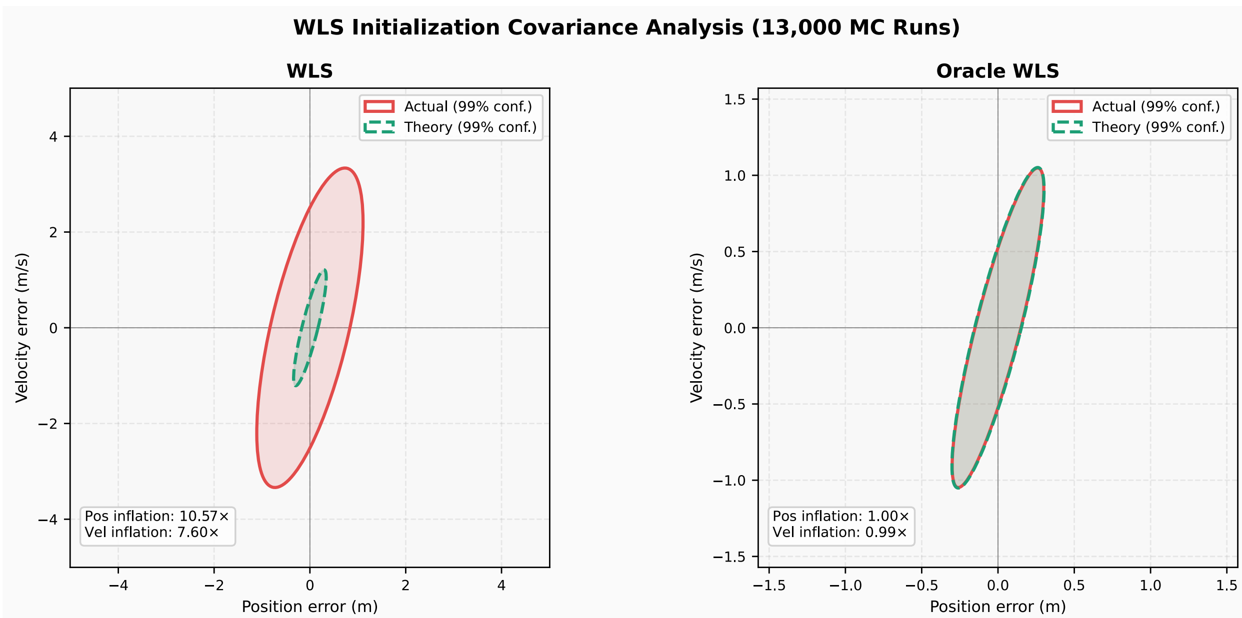

Figure 6.38: Theoretical versus actual 99% confidence covariance ellipses for WLS and Oracle WLS initialization across 13,000 Monte Carlo track promotions. The theoretical ellipse lies inside the actual ellipse in the WLS panel, reflecting filter overconfidence due to clutter contamination of the RANSAC inlier set. The Oracle WLS panel shows near-perfect calibration when true inliers are known.

The Oracle WLS result confirms that the WLS formulation is theoretically sound. When fed a clean inlier set, the theoretical covariance matches the actual estimation error to within 1%. The geometric analysis in Table 6.10 shows that this calibration is not coincidental. Oracle WLS recovers the true observation geometry exactly, so its $\mathbf{P}_{kin}$ correctly reflects the actual estimation uncertainty. The overconfidence observed in the standard WLS method is therefore

Table 6.11: WLS initialization covariance deflation from 13,000 Monte Carlo track promotions across two evaluation windows of the collision with debris scenario. Deflation factor is the ratio of actual to theoretical covariance diagonal. A value greater than 1.0 indicates filter overconfidence.

Method	$\sigma^2_{r,\text{theory}}$ (m^2)	$\sigma^2_{r,\text{actual}}$ (m^2)	Deflation (pos)	Deflation (vel)
WLS	0.0126	0.1336	10.57	7.60
Oracle WLS	0.0099	0.0098	1.00	0.99

attributable to inlier set contamination, which introduces estimation error that falls outside what $\mathbf{P}_{kin}$ is able to represent, rather than to the geometric deflation of $\mathbf{H}^T\mathbf{H}$ alone. This is an inherent characteristic of the RANSAC-based initiator operating in cluttered environments.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis proposed and evaluated a RANSAC-based track incubation framework for multi-target tracking in cluttered 1D range-time environments. The framework was paired with two tracking engines, a Global Nearest Neighbor Kalman Filter and a Track-Oriented Multiple Hypothesis Tracker, and benchmarked against M-of-N initialized counterparts across three simulated collision and debris scenarios under nominal, heavy clutter, and reduced detection probability conditions.

The results demonstrate that RANSAC incubation provides a consistent advantage over M-of-N initiation across all conditions. Both RANSAC engines maintained positive MOTA under heavy clutter while both M-of-N engines collapsed to negative MOTA, providing clear evidence that geometric validation at initiation is beneficial for reliable tracking in high clutter environments. Under nominal and dropout conditions, RANSAC engines outperformed their M-of-N counterparts on MOTA, MOTP, RMSE, and initiation TTI. The WLS initialization validation study confirmed that the theoretical covariance is well calibrated when fed a clean inlier set, but that inlier set contamination in cluttered environments causes the filter to initialize overconfident, an inherent limitation of the proposed RANSAC-based initiation that warrants further investigation.

Comparing the two RANSAC engines, RANSAC-GNNKF led on MOTA in most conditions due to its immediate release of detections to the incubator pool. RANSAC-TOMHT showed an advantage under heavy clutter in some scenarios, consistent with its deferred commitment reducing false associations around collision events, though this advantage was not consistent across all scenarios. The TOMHT detection claiming behavior was identified as a systematic cost of deferred commitment, reducing detections available to the incubator and indirectly increasing initiation TTI relative to the GNNKF engine. However, TOMHT may provide an advantage in scenarios involving maneuvering targets, where deferred commitment is more likely to resolve ambiguity correctly.

The zero velocity initialization used by the M-of-N initiator was identified as a compounding failure mode under dropout conditions, where gate misses after promotion consume miss count budget before the filter can converge to the correct velocity. RANSAC engines are less susceptible to this failure mode because WLS provides a velocity estimate at promotion, keeping the gate aligned with the target from promotion. The primary cost of RANSAC incubation relative to M-of-N initiation is runtime. Both RANSAC engines required significantly more computation per scenario than their M-of-N counterparts, reflecting the cost of repeated random sampling and line fitting across the sliding window detection pool at each scan.

7.2 Future Work

The following items identify extensions and open questions arising directly from the work presented in this thesis:

- Incorporation of hypothesis splitting and merging strategies from HO-RANSAC [23] into the internal verification procedure to improve hypothesis generation in extremely dense clutter environments.

- Investigate the potential for Joint Probabilistic Data Association [18] as an engine for the RANSAC Incubation Framework and compare its performance against the GNN and MHT engines evaluated in this work.

- Perform a parameter sensitivity analysis on the RANSAC incubation framework, characterizing the effects of window length N_w, minimum inlier count N_{min}, inlier threshold τ, and number of iterations K across varying clutter densities and target detection probabilities, following the methodology demonstrated by Niedfeldt et al. [27].

- Investigate methods for dynamically tuning RANSAC parameters including the inlier threshold τ, minimum inlier count N_{min}, and number of iterations K in response to observed clutter density and target detection rate. Adaptive approaches such as likelihood-based automatic threshold selection [28], universal RANSAC frameworks with adaptive stopping criteria [29], and threshold-free marginalized consensus [30] have demonstrated efficiency and accuracy gains in computer vision applications and could provide further refinement to the RANSAC incubator described in this thesis.

- Investigate blending the top-scoring TOMHT hypothesis branches at the N-scan commit step using probability weighting, rather than selecting the single best branch, as a potential method for reducing track loss at target crossings.

- Evaluate tracker performance using the Optimal Subpattern Assignment (OSPA) metric [31] as a unified measure of localization and cardinality error to complement the metrics reported in this work.

- The CLEAR MOT metrics used in this work, while well established, do not directly penalize localization errors for missed and false targets in a mathematically rigorous way. The Generalized Optimal Sub-Pattern Assignment (GOSPA) metric [32] potentially provides a more principled evaluation framework and could be a natural replacement for future work.

- The TOMHT engine claims all detections that gate against any hypothesis branch prior to N-scan pruning, regardless of whether those detections are ultimately assigned to the committed branch. Returning detections from pruned branches to the RANSAC incubator pool after commitment could reduce buffer starvation around collision events and improve debris track initiation TTI in the RANSAC-TOMHT engine.

- The current RANSAC incubator fits linear hypotheses under a constant velocity motion assumption. Extending the framework to run parallel RANSAC searches under multiple motion models, specifically constant velocity and constant acceleration, would allow the incubator to promote tracks whose trajectories are better described by parabolic fits in the range-time plane. At promotion, an Interacting Multiple Model filter [13], [33] could replace the single Kalman filter, with a Markov chain governing the probability weighting between parallel CV and CA sub-filters at each scan. This would extend the framework to maneuvering targets without sacrificing the geometric validation advantage of the RANSAC incubation stage.

- The RANSAC incubation framework presented here operates in a one dimensional measurement space; extension to higher-dimensional space is a natural generalization, with computational complexity scaling as the primary concern for future work.

References

[1] P. Niedfeldt, "Recursive-RANSAC: A novel algorithm for tracking multiple targets in clutter," *Theses and Dissertations*, Jul. 2, 2014. [Online]. Available: https://scholarsarchive.byu.edu/etd/4195.

[2] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, Jan. 2004. Accessed: Apr. 28, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/1263228.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 3464–3468. Accessed: Apr. 28, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/7533003.

[4] B. Ristic, S. Arulampalam, and J. McCarthy, "Target motion analysis using range-only measurements: Algorithms, performance and application to ISAR data," *Signal Processing*, vol. 82, no. 2, pp. 273–296, Feb. 1, 2002. Accessed: Apr. 26, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168401001876.

[5] M. McDonough and W. D. Blair, "Steady-state tracking with FMCW radars," in *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–7. Accessed: May 8, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8396730.

[6] A. Asif, S. Kandeepan, and R. J. Evans, "Passive radar tracking in clutter using range and range-rate measurements," *Sensors*, vol. 23, no. 12, p. 5451, Jan. 2023. Accessed: Jun. 12, 2026. [Online]. Available: https://www.mdpi.com/1424-8220/23/12/5451.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1, 1981. Accessed: Apr. 28, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/358669.358692.

[8] X. Li and C. He, "Optimal initialization of linear recursive filters," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 2, Dec. 1998, 2335–2340 vol.2. Accessed: May 8, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/758693.

[9] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, Sep. 1, 1975. Accessed: May 1, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109875900217.

[10] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1, 1960. Accessed: Apr. 28, 2026. [Online]. Available: `https://doi.org/10.1115/1.3662552`.

[11] A. Becker. “Kalman filter explained through examples. ”[Online]. Available: `https://kalmanfilter.net/`.

[12] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, “Sonar tracking of multiple targets using joint probabilistic data association,” *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, Jul. 1983. Accessed: Apr. 28, 2026. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/1145560`.

[13] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software* (A Wiley-Interscience Publication), Nachdr. New York, NY: Wiley, 2007, 558 pp.

[14] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics (NRL)*, vol. 52, no. 1, pp. 7–21, 2005. Accessed: Apr. 28, 2026. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20053`.

[15] I. S. Duff and J. Koster, “On algorithms for permuting large entries to the diagonal of a sparse matrix,” *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 4, pp. 973–996, Jan. 2001. Accessed: May 1, 2026. [Online]. Available: `https://epubs.siam.org/doi/abs/10.1137/S0895479899358443`.

[16] X. Rong Li and Y. Bar-Shalom, “Tracking in clutter with nearest neighbor filters: Analysis and performance,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 995–1010, Jul. 1996. Accessed: May 2, 2026. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/532259`.

[17] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979. Accessed: Apr. 28, 2026. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/1102177`.

[18] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, “Joint probabilistic data association revisited,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 3047–3055. Accessed: May 1, 2026. [Online]. Available: `https://ieeexplore.ieee.org/document/7410706`.

[19] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, “Multiple hypothesis tracking revisited,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 4696–4704. Accessed: Apr. 28, 2026. [Online]. Available: `https://ieeexplore.ieee.org/document/7410890`.

[20] A. Frank, P. Smyth, and A. Ihler, “A graphical model representation of the track-oriented multiple hypothesis tracker,” in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, Aug.

2012, pp. 768–771. Accessed: Apr. 28, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/6319817.

[21] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[22] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1, 1972. Accessed: Apr. 28, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/361237.361242.

[23] F. Yang, Y. Luo, Q. Li, and Y. Yin, "A hypothesis-optimized RANSAC algorithm for track initiation," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, Jul. 2020, pp. 1–6. Accessed: Apr. 28, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9190627.

[24] P. C. Niedfeldt and R. W. Beard, "Multiple target tracking using recursive RANSAC," in *2014 American Control Conference*, Jun. 2014, pp. 3393–3398. Accessed: May 19, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6859273.

[25] E. Vincent and R. Laganiere, "Detecting planar homographies in an image pair," in *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In Conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, Jun. 2001, pp. 182–187. Accessed: Jun. 11, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/938625.

[26] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008. Accessed: Apr. 28, 2026. [Online]. Available: http://jivp.eurasipjournals.com/content/2008/1/246309.

[27] P. C. Niedfeldt, K. Ingersoll, and R. W. Beard, "Comparison and analysis of recursive-RANSAC for multiple target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 461–476, Feb. 2017. Accessed: May 19, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7812590.

[28] A. Cohen and C. Zach, "The likelihood-ratio test and efficient robust estimation," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2282–2290. Accessed: May 15, 2026. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Cohen_The_Likelihood-Ratio_Test_ICCV_2015_paper.html.

[29] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: A universal framework for random sample consensus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013. Accessed: May 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6365642.

[30] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, “MAGSAC++, a fast, reliable and accurate robust estimator,” presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1304–1312. Accessed: May 15, 2026. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Barath_MAGSAC_a_Fast_Reliable_and_Accurate_Robust_Estimator_CVPR_2020_paper.html.

[31] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008. Accessed: Apr. 28, 2026. [Online]. Available: http://ieeexplore.ieee.org/document/4567674/.

[32] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, “Generalized optimal subpattern assignment metric,” in *2017 20th International Conference on Information Fusion (Fusion)*, Jul. 2017, pp. 1–8. Accessed: Apr. 28, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/8009645.

[33] H. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients,” *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988. Accessed: Jun. 11, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/1299.