ABSTRACT

Title of Dissertation:	NONLINEAR TARGET TRACKING METHODS FOR NEAR-RADIALLY INBOUND TARGETS
	Kevin Robert Ford Doctor of Philosophy, 2022
Dissertation directed by:	Professor John Baras Department of Electrical Engineering and Applied Mathematics

Maneuvering, near-radially inbound (NRI) anti-ship missile (ASM) targets pose a difficult problem for ship self-defense systems. The ongoing evolution of these targets is a primary driver of the research and development tracking algorithms. A series of studies have been performed to develop new endpoint-constrained (EPC) filtering algorithms to track these targets in real time and to evaluate the tracking error performance compared with legacy methods. The findings from these studies demonstrate that EPC filters deliver superior tracking accuracy, with many doing so in real time.

The EPC model is developed by first deriving the spherical constant velocity kinematic model. Taken together with the proportional navigation acceleration model, the spherical constant velocity model can be reformulated to include an endpoint constraint. This constraint was used to develop an EPC extended Kalman filter (EKF) with superior

NRI target tracking performance when compared with the legacy spherical EKF (S-EKF) and the Cartesian EKF (C-EKF).

Stepping beyond the EPC EKF, two additional discrete-time filters are developed in order to provide higher order nonlinear estimation capabilities. These two filters are the EPC unscented Kalman filter (UKF) and EPC bootstrap particle filter (BPF). Both filters offer even greater tracking performance improvements than legacy filters for the set of NRI targets.

Broadening the scope of this research, multiple-model algorithms have been developed to address the non-NRI target motion behavior. A pair of mixed EPC interacting multiple-model (IMM) algorithms have been shown to be more flexible than any of the EPC or legacy filters alone, allowing for accurate tracking of a larger array of target types. EPC IMM algorithms also deliver real-time performance, making them viable for use in self-defense combat systems.

The final chapter returns to first principles of stochastic filtering. The Zakai equation is leveraged to develop a set of continuous-discrete (C-D) filters via the robust Duncan-Mortensen-Zakai (DMZ) equation. These C-D methods are not widely used to track targets in spherical coordinates. A pair of C-D EPC optimal Bayesian filters (OBFs) are developed and found to offer superior performance to all legacy filters and most discrete-time EPC filters, albeit with a higher computational burden.

NONLINEAR TARGET TRACKING METHODS FOR NEAR-RADIALLY INBOUND TARGETS

by

Kevin Robert Ford

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor John Baras, Chair
Professor Ramalingam Chellappa
Professor William Levine
Professor Eyad Abed
Professor Nikhil Chopra

© Copyright by
Kevin Robert Ford
2022

Dedication

To Mom, Dad, Jeffrey, and Steven.

Acknowledgements

Professor John Baras – for taking me on as a part time student and pushing me to reach beyond my comfort zone to find better solutions to old problems. I've experienced personal and professional growth in this program that I never could have imagined, and it wouldn't have been possible without Dr. Baras's guidance.

My dissertation panel members – for guiding me and challenging me to produce the best work possible. I sincerely appreciate Dr. Chellappa, Dr. Levine, Dr. Abed, and Dr. Chopra for taking the time to review my dissertation and provide insightful feedback to improve it.

Dr. Anton Haug – for mentoring me and introducing me to the world of nonlinear filtering. Tony freely shared his time and expertise with me when many others would have happily coasted into retirement. His encouragement motivated me to publish my work and helped me overcome my self-doubt.

My wife Katie – for her encouragement, companionship, and patience. Every day with you is a blessing. You challenge me to be my best self and never lose sight of my goals, even if I tend to get distracted.

Wai Hom – for coaching and supporting me through 10 years of work at APL while I pursued my degree. Thank you for giving me the space, time, and resources to complete this difficult task. A3D is fortunate to have Wai as our supervisor.

Table of Contents

Dedication ii
Acknowledgements iii
Table of Contents iv
List of Figures vii
List of Tables x
List of Abbreviations xi
1. Introduction to NRI Targets and Study Methodology 1
1.1. Motivation 1
1.2. Maneuvering Targets 5
1.3. Trajectories of Interest 7
1.4. Constant Turn Rate Model 11
1.5. Target Trajectory Generator 14
1.6. The Combat System and its Sensors 16
1.7. Performance Evaluation Criteria 18
1.8. Contributions of the Thesis 22
2. Initial EPC Filter Development 25
2.1. Background 25
2.2. The Cartesian EKF 26
2.2.1. Cartesian EKF Development 26
2.2.2. Cartesian EKF Algorithm 29
2.2.3. Non-Gaussian Behavior in the Cartesian EKF 30
2.3. The Spherical EKF 49
2.3.1. Spherical EKF Model 49
2.3.2. Spherical EKF Dynamic Model Jacobian 52
2.3.3. Spherical EKF Observation Model 53
2.3.4. Spherical EKF Algorithm 54
2.4. The EPC EKF 54
2.4.1. EPC Filtering Background 54
2.4.2. EPC Development 56
2.4.3. EPC EKF Algorithm 60
2.5. EPC EKF Proportional Navigation Constant Selection 61
2.5.1. Purpose of Study 61
2.5.2. Findings for PN Constant Tuning 61
2.6. EPC EKF and Legacy Filter Performance Study 64
2.6.1. EPC EKF Comparison Study Goals 64
2.6.2. Radially Inbound Target 64
2.6.3. Dutch Roll Target 67
2.6.4. High-Diver Target 69
2.6.5. Dogleg Target 72
2.6.6. EPC vs Legacy Filter Overall Performance Assessment 75
3. Alternative Discrete-Time EPC Filters 78
3.1. EPC Unscented Kalman Filter 78
3.1.1. UKF Background 78
3.1.2. EPC UKF Development 80

3.1.3. EPC UKF Algorithm 82
3.2. EPC Particle Filter 83
3.2.1. Particle Filtering Background 83
3.2.2. The EPC Bootstrap Particle Filter 85
3.2.3. EPC BPF Algorithm 88
3.3. Performance Study of EPC Filters 90
3.3.1. Purpose of Study 90
3.3.2. Radially Inbound Target 91
3.3.3. Dutch Roll Target 94
3.3.4. High-diver Target 96
3.3.5. Dogleg Target 99
3.3.6. EPC Filter Comparison Overall Performance Assessment 102
4. Robust NRI Target Tracking Using Multiple Models 105
4.1. Multiple Model Algorithm Background 105
4.2. A Pair of EPC IMMs 107
4.2.1. EPC EKF IMM Implementation 107
4.2.2. EPC EKF IMM Performance for the High-diver Target 109
4.2.3. EPC EKF IMM Performance for the Dogleg Target 111
4.2.4. A More Flexible EPC IMM 113
4.3. EPC IMM Tracking Algorithm Performance Study 115
4.3.1. Purpose of Study 115
4.3.2. Radially Inbound Target 116
4.3.3. Dutch Roll Target 120
4.3.4. High-diver Target 124
4.3.5. Dogleg Target 127
4.3.6. EPC IMM Overall Performance Assessment 130
5. Continuous-Discrete Endpoint-Constrained Filters 133
5.1. Background 133
5.1.1. The Zakai Equation 133
5.1.2. The Robust Duncan-Mortensen-Zakai Equation 135
5.1.3. The Yau Filter 136
5.2. Continuous-Discrete Filtering 138
5.2.1. Optimal Bayesian Filtering 138
5.2.2. Spherical Fokker-Planck-Kolmogorov Forward Equation 141
5.3. Preliminary Work with OBFs 148
5.3.1. OBF on the Circle with Known Velocity 148
5.3.2. OBF on the Circle with Unknown Velocity 153
5.3.3. FTCS Finite Difference OBF Algorithm 157
5.4. Development of Practical EPC OBFs 159
5.4.1. EPC Spherical OBF Prediction 159
5.4.2. Planar EPC OBF with Separate Elevation Filter 163
5.4.3. EPC OBF on the Sphere with Separate Range Filter 167
5.5. Performance Study of C-D and D-D Filters 169
5.5.1. EPC OBF Study Goals 169
5.5.2. Radially Inbound Target 170
5.5.3. Dutch Roll Target 172

5.5.4. EPC OBF Study Overall Performance Assessment 175
6. Conclusion 177
6.1. Summary of Findings 177
6.2. Next Steps and Future Development Opportunities 178
References 180

List of Figures

Figure 1: HMCS Regina, a Halifax-class frigate, fires a Harpoon ASM in 2008 during exercises (Credit: article by S. Pasandideh [90]) 6
Figure 2: Example scenario where propagated track lags truth as target maneuvers 7
Figure 3: The radially inbound trajectory 8
Figure 4: The dogleg trajectory 9
Figure 5: The high-diver trajectory 9
Figure 6: The Dutch roll trajectory 10
Figure 7: Dutch roll trajectory position, velocity, and acceleration profiles 11
Figure 8: Example trajectory using trajectory generator inputs 16
Figure 9: AN/SPS-48 Radar on the USS Theodore Roosevelt (source: [6]) 17
Figure 10: Example radar model observation (red stars) and truth (blue line) data over time for a single Monte Carlo run of the dogleg trajectory 18
Figure 11: Box example plot with labels 20
Figure 12: ENU Coordinate System 26
Figure 13: The two dimensional coordinate system used for bearing PDF derivation 31
Figure 14: Plot of radially inbound target (45 degrees from North, σ = 0.2 data miles); Cartesian Gaussian samples (left) and computed bearing samples (right). 32
Figure 15: Incorrect application of inverse tangent function to obtain PDF for Θ results in double peaks ($\mu_X = \mu_Y = 3$ and $\sigma = 2$ for this example). 34
Figure 16: Plot depicting sample standard deviation and its approximation found by taking the square root of the variance for the wrapped Gaussian distribution 40
Figure 17. Plot of the bearing PDF for various values of μ_R with constant $\sigma = 0.1$ 41
Figure 18. Plot of the bearing PDF at $\mu_R = 0$ with $\sigma_X = 0.7$, $\sigma_Y = 1.0$, and $\rho = -0.5$ 43
Figure 19. RMS position error plot for radially inbound target using various process noise intensity values (with units of dmi^2/sec^3), $\Delta t = 1$ second, $\sigma_R = 0.25$ dmi, and $\sigma_\theta = 1$ degree 48
Figure 20: Position error box plot for PN constant tuning study 62
Figure 21: Velocity error box plot for PN constant tuning study 62
Figure 22: EPC EKF vs legacy filters – bearing RMSE for radially inbound scenario 65
Figure 23: EPC EKF vs legacy filters – elevation RMSE for radially inbound scenario 66
Figure 24: EPC EKF vs legacy filters – position error box plots for radially inbound scenario 66
Figure 25: EPC EKF vs legacy filters – velocity error box plots for radially inbound scenario 67
Figure 26: EPC EKF vs legacy filters – bearing RMSE for Dutch roll scenario 68
Figure 27: EPC EKF vs legacy filters – elevation RMSE for Dutch roll scenario 68
Figure 28: EPC EKF vs legacy filters – position error box plots for Dutch roll scenario 69
Figure 29: EPC EKF vs legacy filters – velocity error box plots for Dutch roll scenario 69
Figure 30: EPC EKF vs legacy filters – bearing RMSE for high-diver scenario 70
Figure 31: EPC EKF vs legacy filters – elevation RMSE for high-diver scenario 71
Figure 32: EPC EKF vs legacy filters – position error box plots for high-diver scenario 71
Figure 33: EPC EKF vs legacy filters – velocity error box plots for high-diver scenario 72
Figure 34: EPC EKF vs legacy filters – bearing RMSE for dogleg scenario 73
Figure 35: EPC EKF vs legacy filters – elevation RMSE for dogleg scenario 73

Figure 36: EPC EKF vs legacy filters – position error box plots for dogleg scenario 74
Figure 37: EPC EKF vs legacy filters – velocity error box plots for dogleg scenario 74
Figure 38: Sigma point prediction example in range and bearing 79
Figure 39: BPF algorithm flowchart 90
Figure 40: EPC filter comparison study – bearing RMSE for radially inbound scenario 92
Figure 41: EPC filter comparison study – elevation RMSE for radially inbound scenario 92
Figure 42: EPC filter comparison study – position error box plots for radially inbound scenario 93
Figure 43: EPC filter comparison study – velocity error box plots for radially inbound scenario 93
Figure 44: EPC filter comparison study – bearing RMSE for Dutch roll scenario 94
Figure 45: EPC filter comparison study – elevation RMSE for Dutch roll scenario..... 95
Figure 46: EPC filter comparison study – position error box plots for Dutch roll scenario 95
Figure 47: EPC filter comparison study – velocity error box plots for Dutch roll scenario 96
Figure 48 EPC filter comparison study – bearing RMSE for high-diver scenario 97
Figure 49: EPC filter comparison study – elevation RMSE for high-diver scenario 98
Figure 50: EPC filter comparison study – position error box plots for high-diver scenario 98
Figure 51: EPC filter comparison study – velocity error box plots for high-diver scenario 99
Figure 52: EPC filter comparison study – bearing RMSE for dogleg scenario..... 100
Figure 53: EPC filter comparison study – elevation RMSE for dogleg scenario 100
Figure 54: EPC filter comparison study – position error box plots for dogleg scenario 101
Figure 55: EPC filter comparison study – velocity error box plots for dogleg scenario 101
Figure 56: IMM algorithm block diagram 106
Figure 57: Plot of model probabilities for each *k* value for Dutch roll trajectory 109
Figure 58: EPC IMM model probabilities for high-diver trajectory..... 110
Figure 59: EPC EKF IMM elevation RMSE plot for non-NRI high-diver trajectory.... 111
Figure 60: EPC EKF IMM model probabilities for high-diver scenario 112
Figure 61: EPC EKF IMM bearing RMSE plot for non-NRI dogleg scenario 113
Figure 62: EPC EKF IMM elevation RMSE plot for non-NRI dogleg scenario..... 113
Figure 63: EPC IMM performance study – EPC EKF IMM probability weights for Dutch roll scenario..... 117
Figure 64: EPC IMM performance study – flexible EPC UKF IMM probability weights for Dutch roll scenario 117
Figure 65: EPC IMM performance study – bearing RMSE for radially inbound scenario 118
Figure 66: EPC IMM performance study – elevation RMSE for radially inbound scenario 118
Figure 67: EPC IMM performance study – position error box plots for radially inbound scenario 119
Figure 68: EPC IMM performance study – velocity error box plots for radially inbound scenario 119

Figure 69: EPC IMM performance study – EPC EKF IMM probability weights for Dutch roll scenario 121
Figure 70: EPC IMM performance study – flexible EPC UKF IMM probability weights for Dutch roll scenario 121
Figure 71: EPC IMM performance study – bearing RMSE for Dutch roll scenario 122
Figure 72: EPC IMM performance study – elevation RMSE for Dutch roll scenario ... 122
Figure 73: EPC IMM performance study – position error box plots for Dutch roll scenario 123
Figure 74: EPC IMM performance study – velocity error box plots for Dutch roll scenario 123
Figure 75: EPC IMM performance study – flexible EPC UKF IMM probability weights for dogleg scenario 125
Figure 76: EPC IMM performance study – bearing RMSE for high-diver scenario 125
Figure 77: EPC IMM performance study – bearing RMSE for high-diver scenario 126
Figure 78: EPC IMM performance study – position error box plots for high-diver scenario 126
Figure 79: EPC IMM performance study – velocity error box plots for high-diver scenario 127
Figure 80: EPC IMM performance study – flexible EPC UKF IMM probability weights for dogleg scenario 128
Figure 81: EPC IMM performance study – bearing RMSE for dogleg scenario 129
Figure 82: EPC IMM performance study – elevation RMSE for dogleg scenario 129
Figure 83: EPC IMM performance study – position error box plots for dogleg scenario 130
Figure 84: EPC IMM performance study – velocity error box plots for dogleg scenario 130
Figure 85: Flow diagram for MATLAB's linear system solver mldivide, given non-sparse input matrix (*A*) as shown on the MATLAB help page [81]. 151
Figure 86: Normalized PDF diffusion and drift for particle moving on the circle with known constant velocity. 152
Figure 87: Angular and velocity error comparison for the linear KF and OBF for the radially inbound trajectory. 156
Figure 88: Angular and velocity error comparison for the linear KF and OBF for the Dutch roll trajectory. 157
Figure 89: OBF performance study – bearing RMSE radially inbound trajectory. 171
Figure 90: OBF performance study – elevation RMSE for radially inbound scenario .. 171
Figure 91: OBF performance study – position error box for radially inbound scenario 172
Figure 92: OBF performance study – velocity error box plots for radially inbound scenario 172
Figure 93: OBF performance study – bearing RMSE for Dutch roll trajectory 173
Figure 94: OBF performance study – elevation RMSE for Dutch roll trajectory 174
Figure 95: OBF performance study – position error box plots for Dutch roll trajectory 174
Figure 96: OBF performance study – velocity error box plots for Dutch roll trajectory 175

List of Tables

Table 1: Trajectory generator maneuver types 15
Table 2: Measurement uncertainties for the notional VSR 18
Table 3: C-EKF Algorithm 30
Table 4: S-EKF Algorithm 54
Table 5: EPC EKF Algorithm 60
Table 6: PN constant selection study – overall RMSE table 63
Table 7: EPC EKF vs legacy filters – overall RMSE performance 75
Table 8: EPC EKF vs legacy filters – runtime performance table 76
Table 9: EPC UKF Algorithm 82
Table 10: EPC BPF Algorithm 88
Table 11: EPC filter comparison study – overall RMSE performance 103
Table 12: EPC filter comparison study – runtime performance table 103
Table 13: EPC IMM Algorithm 108
Table 14: Flexible IMM PN value table 115
Table 15: Overall RMSE Performance for EPC IMM Comparison Study 132
Table 16: IMM algorithm comparison study: runtime performance table 132
Table 17: FTCS OBF discretization table 159
Table 18: Overall RMSE performance for the family of EPC filters 176
Table 19: Runtime performance for the family of EPC filters 176

List of Abbreviations

ASM	Anti-ship missile
BPF	Bootstrap particle filter
BTCS	Backward-time centered-space
C-D	Continuous(state)-discrete(observation)
C-EKF	Cartesian extended Kalman filter
CV	Constant velocity
D-D	Discrete(state)-discrete(observation)
dmi	Data miles (1 dmi = 6000 ft)
DMZ	Duncan-Mortensen-Zakai (equation)
FTCS	Forward-time centered-space
g	Earth gravity at sea level (roughly 32.17 ft/s^2)
NRI	Near-radially inbound
EKF	Extended Kalman filter
ENU	East-North-Up
EPC	Endpoint constraint
IMM	Interacting multiple model
OBF	Optimal Bayesian filter
PDF	Probability density function
PF	Particle filter
RBE	Range-bearing-elevation
RTG	Range to go
S-EKF	Spherical extended Kalman filter
S-UKF	Spherical unscented Kalman filter
UKF	Unscented Kalman filter

1. Introduction to NRI Targets and Study Methodology

1.1.Motivation

Supersonic, highly maneuverable anti-ship missiles (ASMs) are a major driving factor for technological developments in the target tracking domain. Innovations on both the defensive side (tracking software, sensor hardware, etc.) and the offensive side (faster, more unpredictable, and maneuverable ASMs) have resulted in an ongoing arms race that continues to merit study by branches of armed services [25]. The penalty for falling behind in the race is non-trivial, but installing updated hardware on legacy vessels, such as high-accuracy and high-update rate sensors, can be quite costly. This dissertation introduces a set of cost-effective target tracking algorithm updates that can offer improved tracking performance against ASMs for cases in which the sensor capabilities are limited.

Many ASMs can be satisfactorily modeled as near-radially inbound (NRI) targets. We define an NRI target is defined as one that approaches the origin (where the sensor is located) and maneuvers along its radially inbound path. By definition, the approximate endpoint of the NRI target trajectory is known. Tracking a maneuvering NRI target is difficult for two reasons. First, a priori knowledge of a target's maneuvers is often unavailable, which can cause a mismatch between the filter's kinematic and noise models and the maneuver. Second, for NRI targets, the legacy Cartesian filter has problems tracking a target in the near field [35].

A new set of nonlinear filtering algorithms has been developed for tracking NRI targets, which adds endpoint-constrained (EPC) acceleration to the dynamic target kinematic model. This added acceleration consistently forces the trajectory to estimate

back towards the radial path. A near constant velocity (CV) kinematic model in spherical coordinates was used as the basic model for each new filter algorithm. Using these algorithms, the following questions will be addressed:

1. Do EPC filters offer improved tracking performance against maneuvering NRI targets compared to legacy filters?
2. If EPC filters offer improved NRI target performance, can they also be used with non-NRI targets?
3. Can the algorithms be executed in real time?

The primary motivation for using a kinematic model in spherical coordinates for an NRI target is that the commonly used CV Cartesian extended Kalman filter (C-EKF) requires a Cartesian-to-spherical transformation to predict the radar measurements. This transformation to spherical angles requires an inverse tangent of the ratio of two Cartesian Gaussian random variables. It is shown in Section 2.2.3 that as the range decreases, the probability distribution of the angle outputs from such a computation transitions from a truncated Gaussian to a uniform (over π to π) one. This transition results in an increasing variance in the angle estimates, which can result in poor tracking performance. The probability density resulting from the Cartesian-to-polar transformation has not been previously documented, and the findings in this dissertation were published in [35].

The second chapter of this dissertation documents a pair of legacy filters used for tracking ASMs. The C-EKF algorithm is developed, and its weaknesses are highlighted. An alternative filter, the spherical EKF (S-EKF) developed by Haug and Williams [46], is shown to be a bridge between the current Cartesian EKF approach and the proposed

family of EPC filters. The C-EKF and S- EKF are also used as a baseline to compare the performance of each new filter using root-mean-square error (RMSE) analysis. The second chapter concludes with a performance evaluation of the EPC spherical EKF (EPC EKF) when compared to legacy filters for a set of trajectories of interest. The results of this study have also been documented by Haug and Ford [45], the author of this dissertation, although these results are expanded here.

The third chapter explores the implementation of an EPC unscented Kalman filter (EPC UKF). UKFs have been shown to exhibit greater tolerance to nonlinearity because of the numerical integration that captures second-order behavior in the state and covariance estimates. Linearization performed by EKFs effectively captures the nonlinear model behavior in the first order [61]. The EPC UKF offers further tracking performance improvements over the EPC EKF for the same ASM trajectories, although it does not address instances in which the target's behavior is not NRI in nature.

Particle filters are also introduced in the third chapter as an alternative to methods that rely on numerical differentiation (EKFs) or affine transformation (UKFs) to estimate the state and covariance. Particle filters offer greater flexibility in nonlinear filtering on a sphere because fewer assumptions must be made regarding the Gaussian nature of the underlying distributions. The particles were passed through a nonlinear kinematic model and recursively filtered. The resulting particles can then be used to estimate the target state and covariance by using empirical mean and covariance calculations. Using one of these recursive sequential importance sampling (SIS) particle filtering methods, the EPC bootstrap particle filter (EPC BPF) delivers an even better track error performance than

the EPC EKF or EPC UKF. The results of these studies were documented by Ford and Haug [34].

Chapter 4 concerns multiple model approaches to solve the NRI target tracking problem in a more flexible manner, which supports a wider variety of trajectories. For the case where the target is only NRI for a portion of its trajectory, a single model proves insufficient. Interacting multiple model (IMM) algorithms is a well-known way to leverage a bank of Kalman filters to obtain state estimates when the target demonstrates a mixture of model behaviors. By implementing an IMM algorithm with an EPC filter and models designed to track non-NRI targets, one can achieve a further boost in performance for a wider variety of target trajectories than a single EKF. Ford and Haug demonstrated the capabilities of an EPC IMM algorithms [34].

Chapter 5 explores the implementation of a fully nonlinear filter that directly solves the underlying nonlinear partial differential equation for the NRI tracking problem without linearization, such as the EKF, numerical integration assuming a Gaussian distribution, such as the UKF, or managing a large number of particles, such as the BPF. This is accomplished by solving the Fokker-Planck-Kolmogorov forward equation to project the target state probability density forward in time, and Bayes' rule to correct the distribution estimate using the latest observation data. The filter development in this section begins with a series of simple examples and builds up to the EPC formulation. The fully nonlinear continuous-discrete (C-D) filtering algorithm has not been used for ASM tracking applications in 3D spherical coordinates in the currently available literature.

Chapter 6 presents the conclusions for each of the primary questions posed in this dissertation. Additional opportunities for future work in the field of EPC target tracking filters, and target tracking methods in general, are also discussed. This discussion includes research which could offer further improvements to some of the EPC filters which fail to address either the tracking error or real time criteria. References for all cited works are included after the conclusions chapter.

1.2.Maneuvering Targets

Anti-ship missile (ASM) usage, such as the one shown in Figure 1 can be traced back to at least the 1960s. The first ASM fired with the intent to harm was in 1967, when the Israeli Destroyer Eilat was attacked and sunk off the coast of Port Said in the Mediterranean Sea [97]. Many countries, including China, Russia, and the United States, continue to develop increasingly sophisticated and lethal ASMs at a rapid pace [65]. The effective range, velocity, and maneuverability of these targets are consistently improving, so self-defense systems must continue to evolve to keep pace. One approach is to continue adding to an already formidable multilayered defense system [90]. A key aspect to improving the defense system is the use of available sensor resources to form accurate target tracks to allow defenses to issue engagements.

Figure 1: HMCS Regina, a Halifax-class frigate, fires a Harpoon ASM in 2008 during exercises (Credit: article by S. Pasandideh [90])

Maneuvering targets, including ASMs, present a challenge to a missile defense system because their maneuvers often have a shorter period than the update rate of the sensor, as described in Section 1.1. Propagating the track between measurements would therefore cause target tracking errors to grow as the track overshoots the true target state due to the target maneuvering back in the other direction between measurements. An example of this phenomenon is shown in Figure 2. A solution for this problem is required. By leveraging the knowledge of the target's endpoint (the ship), a collection of filters is presented in this dissertation, which mitigate the error induced by a system with a low update rate sensor paired with a higher update rate combat system.

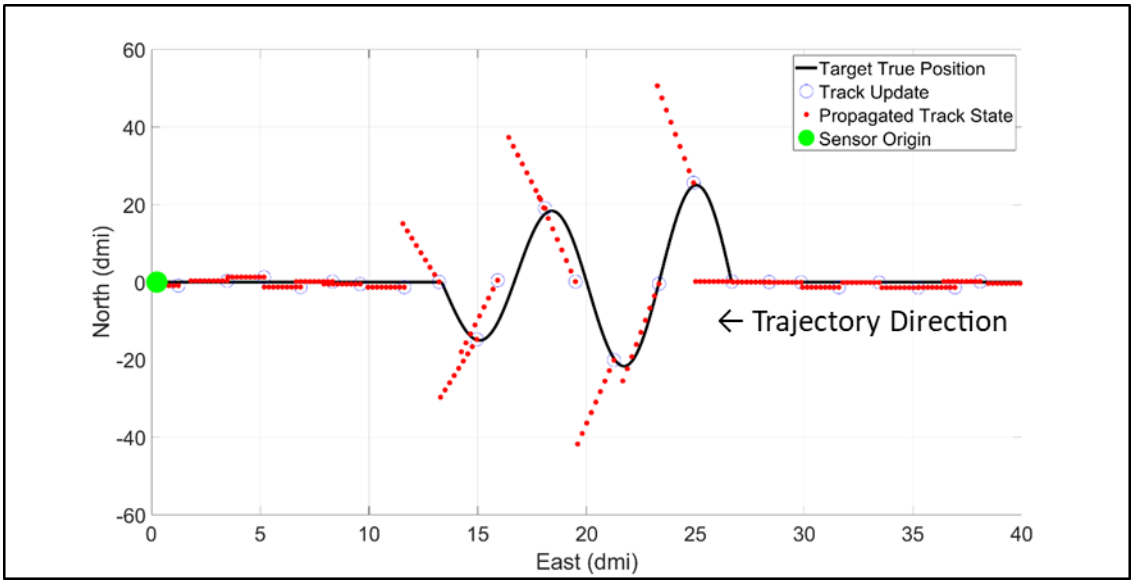

Figure 2: Example scenario where propagated track lags truth as target maneuvers

Ships that use legacy sensors with slow sensor update rates to track fast-moving, highly maneuverable targets present a nontrivial problem. The scenarios explored in this dissertation feature supersonic target trajectories whose maneuver periods are shorter than the update period for the sensor and executed up to 15g's of acceleration. This leads to track errors that may be unacceptable to self-defense systems, which depend on the track data coming out of the combat system. Our studies show that by leveraging the knowledge of the target's predicted endpoint (the ship), an endpoint-constrained target tracking filter can be used to augment the current combat system tracking filters to obtain a better estimate of the target position.

1.3. Trajectories of Interest

The studies covered in this dissertation feature performance evaluations for a variety of maneuvering target trajectories that approach the sensor origin. To highlight the tracking filters' strengths and weaknesses, these trajectories must be capable of

stressing both legacy and newly developed target tracking filters. The four supersonic trajectories selected for use in the performance evaluations were:

1. The radially inbound trajectory, a target that travels at a constant altitude in a roughly straight line towards the sensor origin, is shown in Figure 3. This trajectory is the baseline trajectory and, therefore, the least stressful case, in which all filters should be capable of handling. Failure to accurately track such a target indicates that a filter is not a viable candidate for use in a ship self-defense combat system.

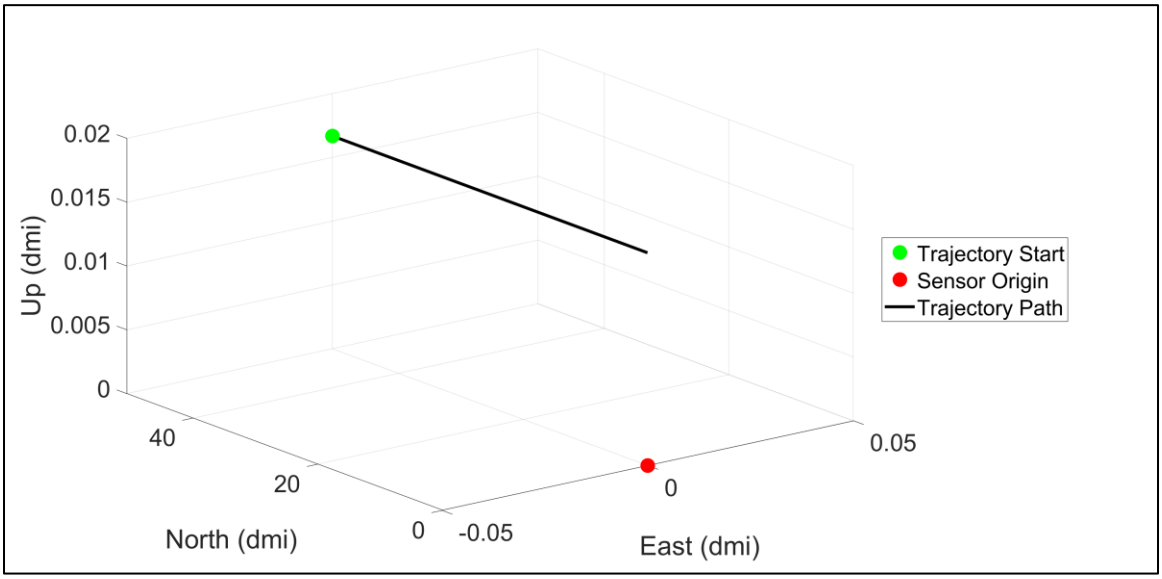

Figure 3: The radially inbound trajectory

2. The dogleg trajectory, in which the target travels in a roughly straight line at a constant velocity, does not approach the sensor origin until it makes an abrupt turn. The trajectory is shown in Figure 4. The dogleg trajectory tests the filters' ability to track the target through a single horizontal crossrange maneuver that features non-NRI behavior.

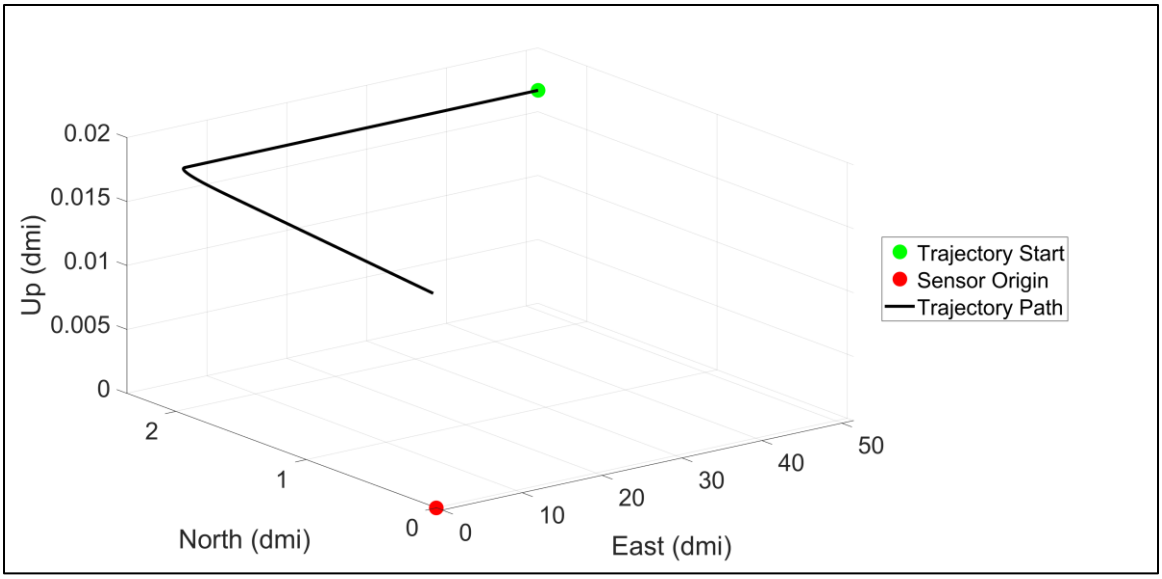

Figure 4: The dogleg trajectory

3. The high-diver trajectory is similar to the dogleg trajectory except that the target begins at a high altitude and makes an abrupt turn in the vertical dimension and heads toward the sensor origin. A high-diver is shown in Figure 5. This trajectory tests whether a tracker can handle maneuvers in the vertical cross-range direction.

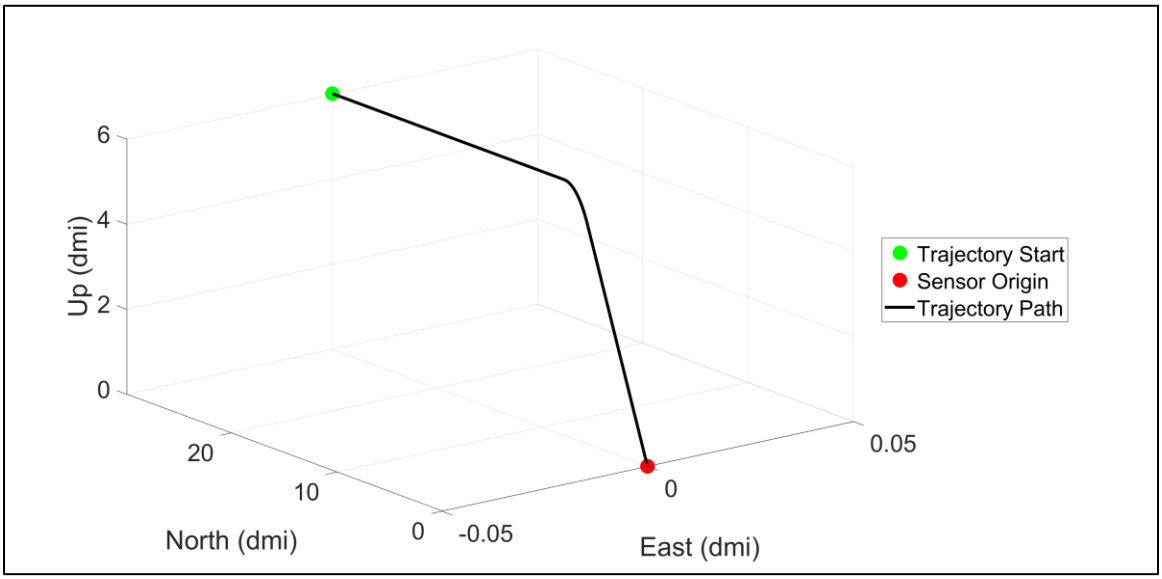

Figure 5: The high-diver trajectory

4. The Dutch roll trajectory was first identified as an undesired coupling of the body axis yaw and roll moments with the sideslip in an aircraft [29]. Among ASM trajectories, the Dutch roll trajectory proves to be quite stressful for tracking systems, as it features multiple high-g turns in both the horizontal and vertical directions. The Dutch roll is shown in Figure 6. Its maneuver profile is easier to understand when the position, velocity, and acceleration terms are plotted with respect to time; therefore, these are included in Figure 7. When the maneuver begins at approximately 80 seconds in the scenario, maneuvers in the radial direction result in high accelerations in the X (east), Y (north), and Z (up) directions.

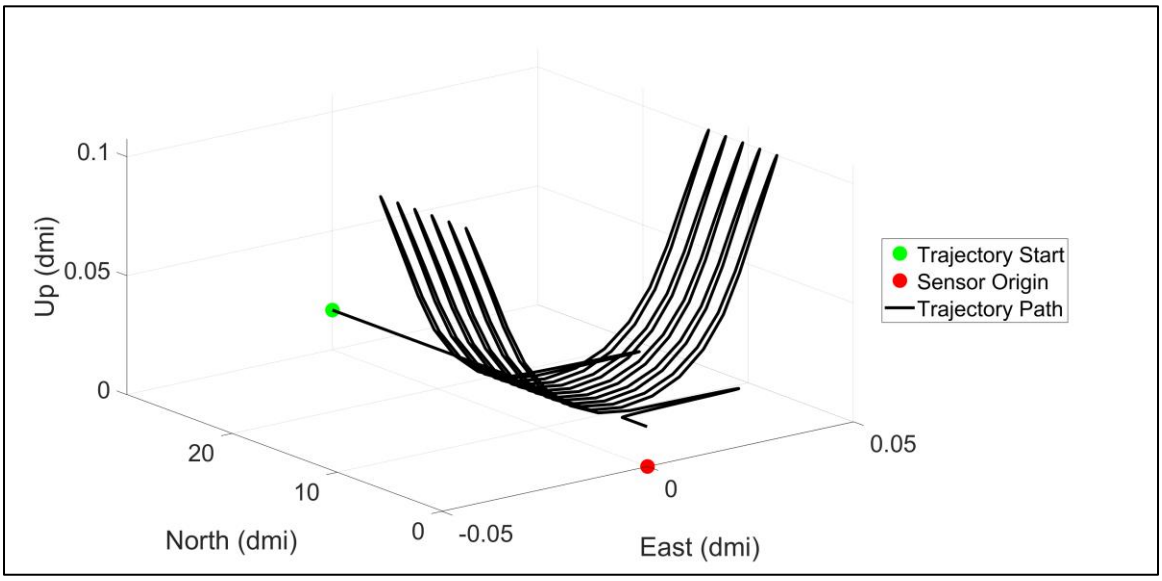

Figure 6: The Dutch roll trajectory

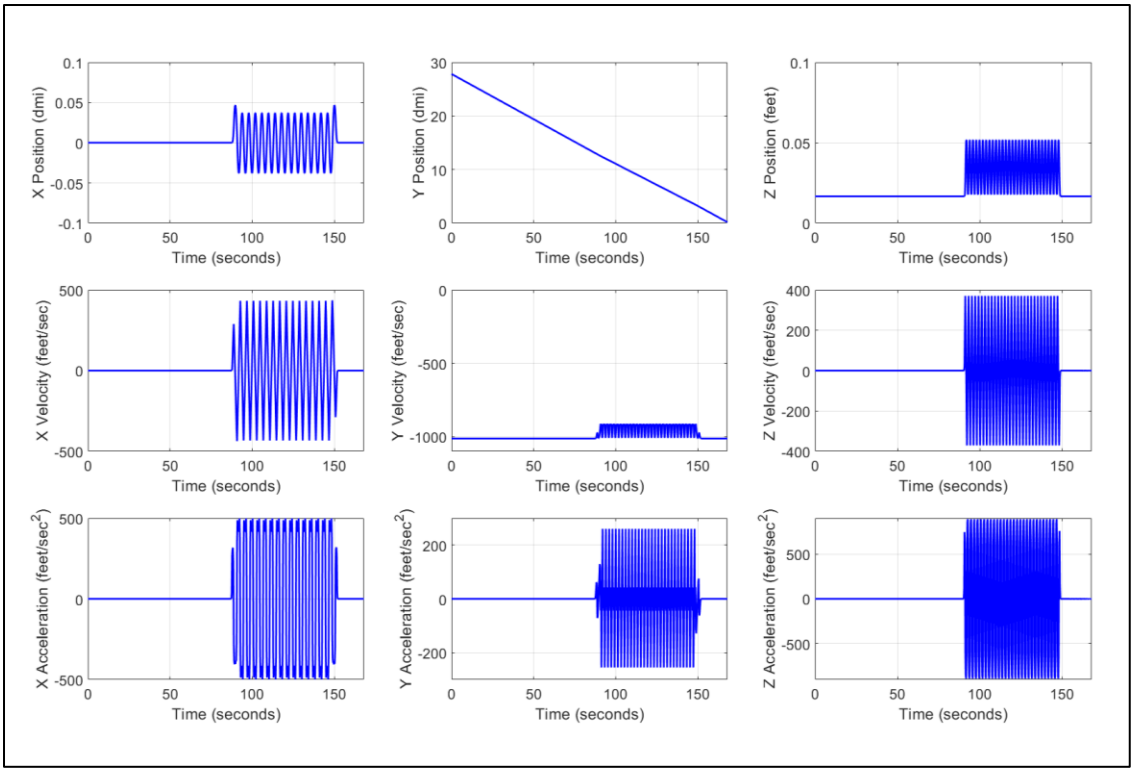

Figure 7: Dutch roll trajectory position, velocity, and acceleration profiles

All target maneuvers are performed at 15g, or 15 times the acceleration due to gravity. More details on the sensor model used to generate observation data for each target are provided in Section 1.6.

1.4. Constant Turn Rate Model

In target tracking applications, constant velocity models assume that the target is traveling at a near-constant velocity under the influence of unknown accelerations. Wind and other environmental effects influence this dynamic noise model, but by far, the largest accelerations are due to unexpected target maneuvers. In self-defense target tracking applications, target maneuvers are not known a priori; therefore, the combat

system's target tracking algorithms are typically tuned to the highest possible acceleration that a target can execute.

For our simulations, it is important to be able to parameterize maneuverability as the number of g's of acceleration that the target is capable of so that the dynamic models in our tracking algorithms are tuned properly. Consequently, a constant turn rate acceleration model was chosen to generate the target trajectories. The constant turn rate acceleration model was formally derived in [104]. All the simulated targets were generated in MATLAB, and the trajectories were broken up into discretized samples. The Cartesian state of the target at time *n* is defined as

$$\boldsymbol{x}_{\mathrm{n}} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}_{\mathrm{n}}^{\mathrm{T}} \tag{1.1}$$

To advance to the next time step (n+1) one must execute

$$\boldsymbol{x}_{\mathrm{n+1}} = \boldsymbol{F}_{\mathrm{CTR}} \boldsymbol{x}_{\mathrm{n}} \tag{1.2}$$

where $\boldsymbol{F}_{\mathrm{CTR}}$ is the dynamic transition matrix for the constant turn rate model. The turn rates in the horizontal and vertical Cartesian planes are:

$$\dot{\gamma}_{\mathrm{n}} = \frac{\hat{\omega}_{\mathrm{h}} a_h}{\sqrt{\dot{x}_{\mathrm{n}}^2 + \dot{y}_{\mathrm{n}}^2 + \dot{z}_{\mathrm{n}}^2}}, \tag{1.3}$$

and

$$\dot{\chi}_{\mathrm{n}} = \frac{-\hat{\omega}_{\mathrm{v}} a_{\mathrm{v}}}{\sqrt{\dot{x}_{\mathrm{n}}^2 + \dot{y}_{\mathrm{n}}^2 + \dot{z}_{\mathrm{n}}^2}}, \tag{1.4}$$

respectively, where $\hat{\omega}_{\mathrm{h}}$ is positive clockwise, $\hat{\omega}_{\mathrm{v}}$ is positive upward, a_{h} is the maneuver acceleration in the horizontal plane, and a_{v} is the maneuver acceleration in the vertical plane. These two acceleration terms are necessary later because their maximum values influence the filter process noise intensity selection by the filter designer.

The dynamic transition matrix used in the trajectory generator is

$$\boldsymbol{F}_{\mathrm{n}}^{\mathrm{CTR}}=\begin{bmatrix} 1 & \frac{g_1+g_2}{2} & 0 & \frac{h_1+h_2}{2} & 0 & \frac{h_1-h_2}{2}\sin\psi_{\mathrm{n}}-\frac{g_1-g_2}{2}\cos\psi_{\mathrm{n}} \\ 0 & \frac{c_1+c_2}{2} & 0 & \frac{s_1+s_2}{2} & 0 & \frac{s_1-s_2}{2}\sin\psi_{\mathrm{n}}-\frac{c_1-c_2}{2}\cos\psi_{\mathrm{n}} \\ 0 & -\frac{h_1+h_2}{2} & 1 & \frac{g_1+g_2}{2} & 0 & \frac{g_1-g_2}{2}\sin\psi_{\mathrm{n}}+\frac{h_1-h_2}{2}\cos\psi_{\mathrm{n}} \\ 0 & -\frac{s_1+s_2}{2} & 0 & \frac{c_1+c_2}{2} & 0 & \frac{c_1-c_2}{2}\sin\psi_{\mathrm{n}}+\frac{s_1-s_2}{2}\cos\psi_{\mathrm{n}} \\ 0 & \frac{1-d_1}{\dot{\chi}_{\mathrm{n}}\sin\psi_{\mathrm{n}}} & 0 & \frac{1-d_2}{\dot{\chi}_{\mathrm{n}}\cos\psi_{\mathrm{n}}} & 1 & \frac{\sin\dot{\chi}_{\mathrm{n}}T}{\dot{\chi}_{\mathrm{n}}} \\ 0 & \frac{d_2}{\sin\psi_{\mathrm{n}}} & 0 & \frac{d_2}{\cos\psi_{\mathrm{n}}} & 0 & \cos\dot{\chi}_{\mathrm{n}}T \end{bmatrix}. \tag{1.5}$$

The elements of (1.5) are constructed using the horizontal (1.3) and vertical (1.4) turn rates, the time since the last state projection (T), and the current target heading taken clockwise from the y-axis in the horizontal plane ψ_{n}.

$$\Omega_{\mathrm{n}}^{+}=\dot{\gamma}_{\mathrm{n}}+\dot{\chi}_{\mathrm{n}} \tag{1.6}$$

$$\Omega_{\mathrm{n}}^{-}=\dot{\gamma}_{\mathrm{n}}-\dot{\chi}_{\mathrm{n}} \tag{1.7}$$

$$c_1=\cos\left(\Omega_{\mathrm{n}}^{-}T\right) \tag{1.8}$$

$$c_2=\cos\left(\Omega_{\mathrm{n}}^{+}T\right) \tag{1.9}$$

$$s_1=\sin\left(\Omega_{\mathrm{n}}^{-}T\right) \tag{1.10}$$

$$s_2=\sin\left(\Omega_{\mathrm{n}}^{+}T\right) \tag{1.11}$$

$$g_1=\begin{cases} T & if \quad \Omega_{\mathrm{n}}^{-}=0 \\ \dfrac{s_1}{\Omega_{\mathrm{n}}^{-}} & else \end{cases} \tag{1.12}$$

$$g_2 = \begin{cases} T & if \quad \Omega_n^+ = 0 \\ \dfrac{s_2}{\Omega_n^+} & else \end{cases} \tag{1.13}$$

$$h_1 = \begin{cases} 0 & if \quad \Omega_n^- = 0 \\ \dfrac{1-c_1}{\Omega_n^-} & else \end{cases} \tag{1.14}$$

$$h_2 = \begin{cases} T & if \quad \Omega_n^+ = 0 \\ \dfrac{1-c_2}{\Omega_n^+} & else \end{cases} \tag{1.15}$$

$$d_1 = \begin{cases} \cos\left(\dot{\chi}_n T\right) & \text{if} \quad \dfrac{\pi}{4} \le \left|\psi_n\right| < \dfrac{3\pi}{4} \\ 1 & \text{else} \end{cases} \tag{1.16}$$

$$d_2 = \begin{cases} 1 & \text{if} \quad \dfrac{\pi}{4} \le \left|\psi_n\right| < \dfrac{3\pi}{4} \\ \cos\left(\dot{\chi}_n T\right) & \text{else} \end{cases} \tag{1.17}$$

$$d_3 = \begin{cases} \sin\left(\dot{\chi}_n T\right) & \text{if} \quad \dfrac{\pi}{4} \le \left|\psi_n\right| < \dfrac{3\pi}{4} \\ 0 & \text{else} \end{cases} \tag{1.18}$$

$$d_4 = \begin{cases} 0 & \text{if} \quad \dfrac{\pi}{4} \le \left|\psi_n\right| < \dfrac{3\pi}{4} \\ \sin\left(\dot{\chi}_n T\right) & \text{else} \end{cases} \tag{1.19}$$

1.5. Target Trajectory Generator

The trajectories described in Section 1.3 were created using MATLAB software developed to support our filter analyses. All trajectories are a composite of a set of user-specified maneuvers, with inputs such as the range to go (RTG) at which the maneuver should be executed, number of g's of acceleration used in the maneuver, and angle at which the target should be traveling relative to its velocity vector at the start of the

maneuver. All options available in this trajectory generator are listed in Table 1, and an example trajectory pointing out how the maneuvers are executed is shown in Figure 8.

Table 1: Trajectory generator maneuver types

Maneuver Name	**Inputs**	**Description**
Coast	None (automatic)	The coast maneuver propagates the target at a constant speed along the direction of the velocity vector from the previous sample. Coast is executed at any point in the trajectory in which the target has no specified maneuvers. If the first user-specified maneuver does not occur at the beginning of the target's trajectory, the object moves in the direction specified in the initial conditions. In this way, coast acts as filler, simplifying the task of stringing together separate maneuvers in the input file.
Forced Coast	▪ Coast distance (dmi) ▪ RTG (dmi)	Forced coast is a user-specified maneuver that propagates the target forward at a constant velocity in the same manner as the coast maneuver. However, the forced coast requires an input specifying the distance to be coasted in nautical miles.
Linear Acceleration	▪ RTG (dmi) ▪ Acceleration (g's) ▪ Final speed (ft/s)	The linear acceleration maneuver is a linear acceleration whereby the user specifies the acceleration the target will undergo (in g's) as well as the desired final speed. The trajectory generator executes this acceleration increase instantaneously, rather than lagging or gradually increasing.
Turn	▪ RTG (dmi) ▪ Turn acceleration (g's) ▪ Turn angle (deg) ▪ Dive acceleration (g's) ▪ Dive angle (deg)	The turn uses the constant turn-rate model to modify the heading and/or attitude of the target. In the case of a target traveling at a constant altitude, a turn would result in a change of direction in the *x-y* (east-north) plane. Dives and turns can be executed simultaneously using the turn maneuver by specifying the RTG at which the maneuver begins, the turn/dive angles, and the turn/dive turning g's.
Approach	None	Approach is a combination of a dive and a turn maneuver that causes the target to seek the shortest path to the origin within the limits of the target's initial conditions (the maximum g's it can turn). During the maneuver, the target first makes a horizontal turn toward the origin, then immediately follows with a dive toward the origin. The maneuver requires no input from the user, and its accuracy is strongly dependent on the sample rate of the trajectory.

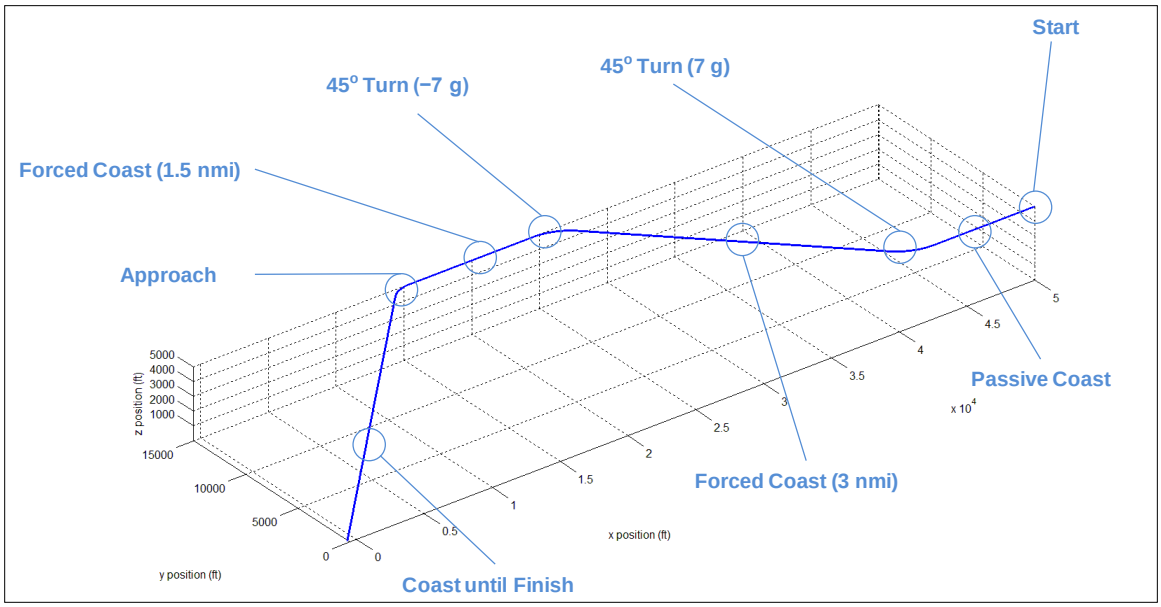

Figure 8: Example trajectory using trajectory generator inputs

1.6. The Combat System and its Sensors

One of the main motivations for developing an improved tracking filter is related to the update rate of the sensor (model) to be used in this set of studies. The sensor model used for the studies performed in this study is a simplified representation of the AN/SPS-48 radar (Figure 9). This selection was made because (1) the radar is commonly used in vessels throughout the US navy, (2) in older legacy vessels, the AN/SPS-48 may be the only method of tracking airborne targets, and (3) the AN/SPS-48 is capable of delivering the range, bearing, and elevation measurements needed for tracking in three dimensions (in addition to the range rate). Adding and integrating new sensors to aging vessels can be costly, and we will show that the target tracking effectiveness of ships with low update rate sensors, such as AN/SPS-48, can be significantly improved with cost-effective software changes.

Figure 9: AN/SPS-48 Radar on the USS Theodore Roosevelt (source: [6])

The sensor in question is a volume search radar (VSR) with an update period of 4 seconds (T=4) that delivers measurements of range, range rate, bearing, and elevation. The radar delivers its measurements to a combat system whose notional reporting period is 0.25 seconds. This period mismatch (4 seconds vs 0.25 seconds) means that the estimated target position or track must be propagated (or coasted) between radar updates.

Radar measurements for the trajectories shown in Figure 3–Figure 6 were generated using a model for a notional volume search radar (VSR) based on the AN/SPS-48, shown in Figure 9. The AN/SPS-48 variants are common throughout the US navy and have a relatively slow scan period, indicating that this is an ideal sensor for testing updated tracking algorithms that show improvements for slow-scanning sensors against highly maneuverable targets. The measurement accuracies of the VSR models are listed in Table 2.

Table 2: Measurement uncertainties for the notional VSR

Update Rate (sec)	Measurement Accuracies (1σ Values)			
	Range (dmi)	Range Rate (knots)	Bearing (deg)	Elevation (deg)
4	0.16	20	0.3	0.3

Assuming that the VSR is unbiased, these measurement accuracies can be used within the VSR model to add zero-mean Gaussian noise, with variances equal to the square of the measurement accuracies. Radar measurements were generated for each trajectory variant based on an east-north-up (ENU) sensor origin position of (0,0,0). A single Monte Carlo run's worth of measurements for the dogleg trajectory are shown in Figure 10.

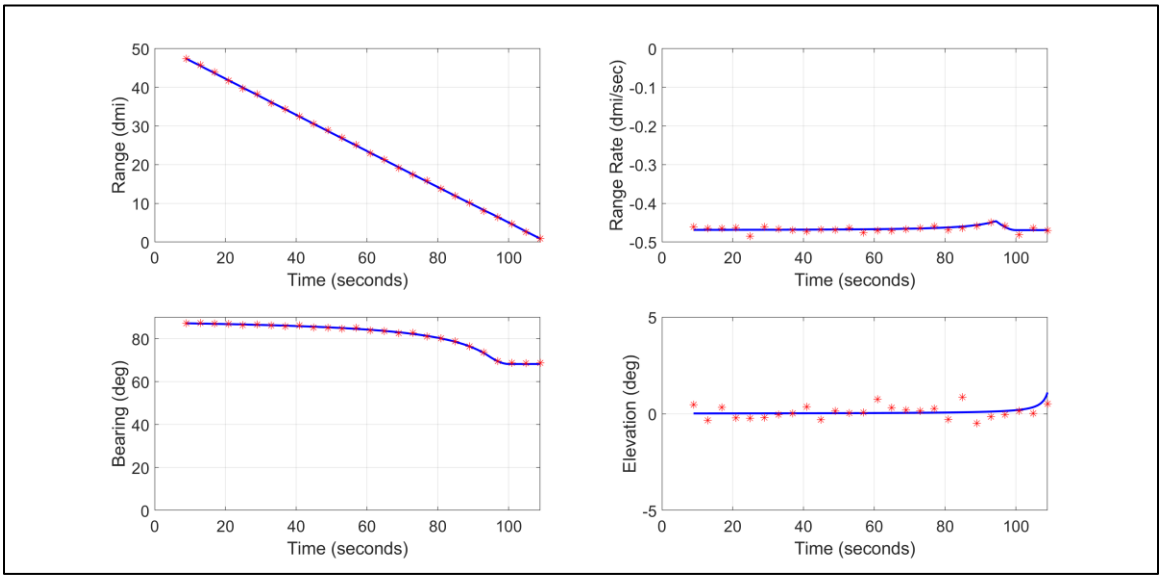

Figure 10: Example radar model observation (red stars) and truth (blue line) data over time for a single Monte Carlo run of the dogleg trajectory

1.7.Performance Evaluation Criteria

Observations were simulated using the radar model described in Section 1.6 for the trajectories described in Section 1.3. A total of 100 Monte Carlo runs were performed to generate the radar observations for each target scenario. These observations were

generated using MATLAB's random-number generator stream and were time-aligned to simplify the performance comparisons across multiple runs. This means that, while every sensor observation (range, range rate, bearing, and elevation) is unique for each run, sequential observations occur simultaneously.

Two primary criteria were used to evaluate the performance of the nonlinear tracking filters in this dissertation. The first is the tracking error, which is captured by subtracting the true state of the target at a given time from the current track state. As discussed previously, minimizing tracking errors, especially angle errors, is of paramount importance to downstream consumers of the self-defense tracker algorithm. The absolute values of these track errors were collected, and their statistics were displayed using box plots. This enables a visual comparison of the overall error performance of each filter based on the statistics for all track updates across all Monte Carlo runs. A labeled example of a box-and-whisker plot (also referred to as a box plot) is shown in Figure 11. The "non-outlier" maxima and minima represent errors that are furthest away from the median on either side of the box, but are still within a distance of 1.5 times the distance to the nearest quartile. Outliers represent any track update errors outside this range.

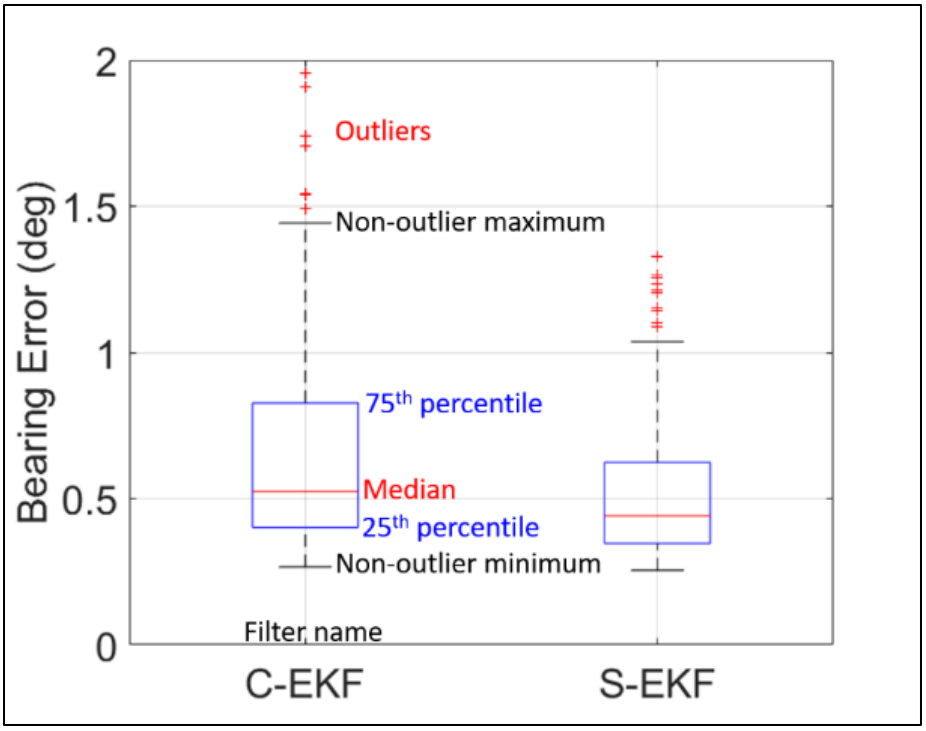

Figure 11: Box example plot with labels

The track error performance was also captured using root-mean-square error (RMSE), which delivers the square root of the quadratic mean of the differences between the filter estimates and the truth states [52]. In our analyses, RMSE was used in two ways:

1. To visualize the filter estimation errors normalized across all Monte Carlo runs at each instance in time and
2. As an overall measure of filter performance by taking the RMSE of all samples across all Monte Carlo runs.

An example calculation of the RMSE for a token estimate $\hat{x}$ at a given time (t_n) across a number of Monte Carlo runs (N_{MC}) is

$$\mathrm{RMSE}_{t_n}\left(\hat{x}\right) = \sqrt{\frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\left(\hat{x}_j - x_j^{\mathrm{truth}}\right)^2}. \tag{1.20}$$

The total RMSE for all simulation times (N_t) and across all Monte Carlo runs is

$$\mathrm{RMSE}_{\mathrm{total}}\left(\hat{x}\right)=\sqrt{\frac{1}{N_{\mathrm{t}}N_{\mathrm{MC}}}\sum_{n=1}^{N_t}\sum_{j=1}^{N_{\mathrm{MC}}}\left(\hat{x}_j\left(t_n\right)-x_j^{\mathrm{truth}}\left(t_n\right)\right)^2}. \tag{1.21}$$

The second criterion evaluated in each performance study was the run time. For each filter, a set of empirical tests was performed to track the time required to execute each algorithm across all the Monte Carlo runs. To support the combat system properly, each nonlinear filtering algorithm must be capable of delivering track updates within milliseconds of receiving the latest radar observations. The concept of "real time" is somewhat subjective, depending on the time-critical nature of systems relying on the target track data. For our analyses, a ratio of simulation time to scenario time is used to evaluate whether the algorithm satisfies the "real time" criteria. As a percentage, a simulation-scenario time ratio of 5% corresponds to 50 milliseconds out of every second used to report track updates to the combat system. Algorithms with ratios below 5% will be considered "real time".

Each subsequent chapter of this dissertation will be punctuated by a performance analysis. To engage an incoming ASM, it is important to provide the best possible angle track on the target so that anti-ASM countermeasures can be directed accurately. Consequently, minimizing bearing and elevation errors is crucial to the success of any given scenario. Therefore, RMSE performance for the bearing and elevation estimates of the target will be the focal point of each study, as well as a check for real-time tracker performance. The overall metrics will also be shared to show which filters perform best in each scenario.

1.8. Contributions of the Thesis

The first problem examined in this dissertation is the application of an endpoint-constrained spherical target motion model to the EKF algorithm in order to track NRI targets. The novel endpoint constraint, shown in Equations (2.140)-(2.144), was applied by leveraging corrective accelerations found in airborne proportional navigation, and was published in an IEEE FUSION conference paper [45]. In that paper, we showed that the EPC EKF delivers track state updates with smaller errors than the legacy constant velocity Cartesian EKF or spherical EKF. The EPC model development, EPC EKF implementation, and performance comparison between the EPC EKF and legacy filters are given in Chapter 2. The relevant publication is given below:

A. J. Haug and K. R. Ford, "An end-point-constrained extended Kalman filter for tracking maneuvering near-radially inbound targets," 2016 19th International Conference on Information Fusion (FUSION), 2016, pp. 312-318.

As part of the research into development of the EPC EKF, a fundamental weakness of the Cartesian EKF: the non-Gaussian behavior of the bearing observation predictions at close range. For this algorithm, the state and covariance data are maintained in Cartesian coordinates, but observations are delivered in spherical coordinates. This can lead to track divergence at close range if the filter designer does not make a judicious selection for the target dynamic (or plant) noise intensity. In our IEEE Access regular paper [35], the probability density function (PDF) for the bearing prediction is derived. This PDF has not previously been defined in the available literature, nor has the direct connection to Cartesian EKF track divergence been previously made. These topics are covered in Section 2.2.3. The relevant publication is:

K. R. Ford and A. J. Haug, "The Probability Density Function of Bearing Obtained from a Cartesian-to-Polar Transformation," IEEE Access Regular Papers, Vol. 10, pp. 32803-32809, 2022.

Following up on the results of studies covered in Chapter 2, a pair of nonlinear target tracking filters offering even better tracking performance than the EPC EKF are developed and presented in Chapter 3. EPC unscented Kalman filter and the EPC bootstrap particle filter, were developed in our IEEE Aerospace Transactions paper [34], the first time the EPC has been studied in these types of filter algorithms. Their performance was shown to be superior to the legacy EKFs as well as the EPC EKF for the highly maneuverable NRI Dutch Roll target. The relevant publication for this work is:

K. R. Ford and A. J. Haug, "A Study of Endpoint-Constrained Nonlinear Tracking Filters," IEEE Transactions on Aerospace and Electronic Systems, Vol. 57, pp. 3952-3961, 2021.

The results of the study published in [34] were used to develop a robust interacting multiple model algorithm capable of tracking both NRI and non-NRI targets. This algorithm leverages four filters with a mixture of EPC and non-EPC models and solves the track divergence problem caused when an EPC model is used against a target whose motion is non-NRI for a portion of its trajectory. The algorithm is developed in Chapter 4, while the algorithm is laid out in Table 13. The relevant publication is the second half of the IEEE Aerospace transactions paper which introduced the EPC UKF and EPC BPF:

K. R. Ford and A. J. Haug, "A Study of Endpoint-Constrained Nonlinear Tracking Filters," IEEE Transactions on Aerospace and Electronic Systems, Vol. 57, pp. 3952-3961, 2021.

The final problem examined in this dissertation pertains to using continuous-discrete optimal Bayesian filtering techniques to solve the NRI target tracking problem. As part of this research, the Fokker-Planck-Kolmogorov forward equation for constant velocity spherical motion, which has not been described in the open literature, is derived in Chapter 5 and given in Equation (5.65). This partial differential equation is used to propagate the estimate of the target state probability density function forward in time between observations as part of the optimal Bayesian filtering (OBF) algorithm. A pair of novel EPC OBFs are also presented in Chapter 5, where their tracking performance is demonstrated to be superior to the set of EPC filters presented in Chapter 2 and Chapter 3 for the set of NRI targets.

2. Initial EPC Filter Development

2.1. Background

Kalman filters have been extensively studied over the last several decades as a means of solving target tracking problems. Indeed, in any field where partially observable random processes appear, a Kalman filter is likely to be used in an attempt to estimate the state of that process. The Kalman filter can optimally solve linear systems with Gaussian errors [51]. They have been used to estimate the positions, velocities and orientations of cars [103], aircraft [2], ships [3], and most importantly, for our application, ASMs [16][34][46][86].

The Kalman filter algorithm was originally designed to optimally solve linear system problems that feature Gaussian distributions [58] for both dynamics and observations. Modifications must be made to the algorithm to solve nonlinear systems, as well as systems where the errors are non-Gaussian. These tend to be problems that have wider applications in the real world, because many physical phenomena and observation models cannot be related using only linear functions. Consequently, filter designers from diverse fields and backgrounds have been modifying Kalman's algorithm for decades to solve their problems. The fruits of their labor are multiple families of efficient filtering algorithms [5][43].

In Sections 2.2–2.4, several members of the discrete-time nonlinear filtering algorithm family are studied. The baseline filter used for our initial studies is the CV C-EKF, which is similar to the algorithms currently in use by some modern combat systems [107]. Other authors examined the CV S-EKF as an alternative to C-EKF with some modest tracking performance improvement [46]. The S-EKF dynamic model gives rise to

the model used to develop the kinematic models used in the family of EPC filters. All three filters are compared in Section 2.6.

2.2. The Cartesian EKF

2.2.1. Cartesian EKF Development

The filter that will be used as a baseline for comparison with the other filters developed in this research is the C-EKF. This filter has been used for decades in anti-missile defense systems [107]. Its longevity is partly due to the ease of modelling the target dynamics in Cartesian coordinates. The C-EKF has proven to be successful in tracking maneuvering targets, but its performance suffers both at close range and for highly maneuvering targets. A high-level overview of the C-EKF's development is sufficient to highlight these issues.

Consider an object moving at a constant velocity in a Cartesian East-North-Up (ENU) coordinate system, as shown in Figure 12.

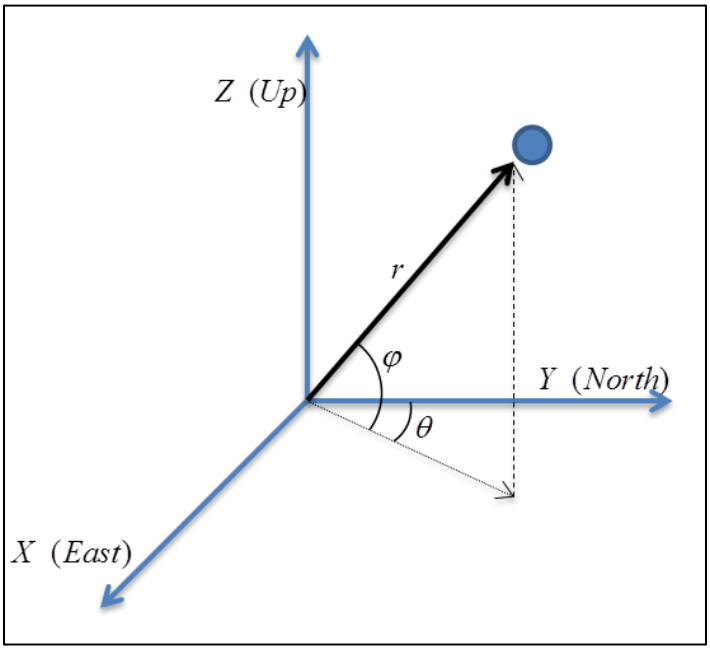

Figure 12: ENU Coordinate System

The state of this object (the circle) can be described by the vector

$$\boldsymbol{x} = \left[x, \dot{x}, y, \dot{y}, z, \dot{z}\right]^{\mathrm{T}}, \tag{2.1}$$

where x, y, and z represent the respective east, north, and up coordinates relative to the sensor (origin) and the dotted components represent the speed of in each direction. The finite difference equations of motion for a near-CV object in Cartesian coordinates then become:

$$\boldsymbol{x}_{\mathrm{n}} = \boldsymbol{F}\boldsymbol{x}_{\mathrm{n-1}} + \boldsymbol{v}_{\mathrm{n-1}}, \tag{2.2}$$

where

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{0}_2 & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & \boldsymbol{F}_1 & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & \boldsymbol{0}_2 & \boldsymbol{F}_1 \end{bmatrix}, \tag{2.3}$$

$$\boldsymbol{F}_1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \tag{2.4}$$

and

$$\boldsymbol{0}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.5}$$

In (2.4), T is the update period for the sensor. In equation (2.2), the vector $\boldsymbol{v}_{\mathrm{n}}$ represents the dynamic acceleration noise, which is characterized as

$$\boldsymbol{v}_n \sim \mathrm{N}\left(\boldsymbol{0}, \boldsymbol{Q}\right), \tag{2.6}$$

where the covariance matrix is found to be

$$\boldsymbol{Q} = \begin{bmatrix} q_{\mathrm{x}}\boldsymbol{Q}_1 & \boldsymbol{0}_2 & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & q_{\mathrm{y}}\boldsymbol{Q}_1 & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & \boldsymbol{0}_2 & q_{\mathrm{z}}\boldsymbol{Q}_1 \end{bmatrix}. \tag{2.7}$$

For the continuous white acceleration noise case [15],

$$\boldsymbol{Q}_1 = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}. \tag{2.8}$$

The dynamic noise intensity values q_x, q_y, and q_z represent the expected maneuverability of the target in each dimension. For a CV filter, these are tuned using the highest expected acceleration that any target being tracked may exhibit. The result for any of these 'q' values is

$$q = (2gC)^2, \tag{2.9}$$

where g is equal to the number of multiples of Earth's gravitational acceleration, and the conversion factor to dmi/s^2 is

$$C = 32.174 / 6000. \tag{2.10}$$

Note that the highly maneuverable targets examined in the performance studies featured in this dissertation execute maneuvers at 15g's.

Let the sensor at the origin have a measurement vector

$$\boldsymbol{z} = [R, \dot{R}, \theta, \varphi]^{\mathrm{T}}, \tag{2.11}$$

with a fixed measurement noise covariance matrix given by

$$\boldsymbol{R} = \begin{bmatrix} \sigma_R^2 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{R}}^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & \sigma_\varphi^2 \end{bmatrix}. \tag{2.12}$$

The observation vector can also be written in terms of the state vector $\boldsymbol{x}$ where

$$\boldsymbol{z}_n = \mathbf{h}(\boldsymbol{x}_n) + \boldsymbol{w}_n. \tag{2.13}$$

The vector function $\mathbf{h}(\boldsymbol{x}_n)$ performs a Cartesian-to-spherical transformation on the Cartesian state vector [43]:

$$\mathbf{h}(\boldsymbol{x}_{\mathrm{n}}) = \begin{bmatrix} \sqrt{x_{\mathrm{n}}^2 + y_{\mathrm{n}}^2 + z_{\mathrm{n}}^2} \\ \dfrac{x_{\mathrm{n}}\dot{x}_{\mathrm{n}} + y_{\mathrm{n}}\dot{y}_{\mathrm{n}} + z_{\mathrm{n}}\dot{z}_{\mathrm{n}}}{\sqrt{x_{\mathrm{n}}^2 + y_{\mathrm{n}}^2 + z_{\mathrm{n}}^2}} \\ \arctan\left(\dfrac{x_{\mathrm{n}}}{y_{\mathrm{n}}}\right) \\ \arctan\left(\dfrac{z_{\mathrm{n}}}{\sqrt{x_{\mathrm{n}}^2 + y_{\mathrm{n}}^2}}\right) \end{bmatrix}. \tag{2.14}$$

The Jacobian of $\boldsymbol{h}(\boldsymbol{x}_n)$ evaluated at the state estimate,

$$\hat{\mathbf{H}}(\hat{\boldsymbol{x}}_{\mathrm{n|n\text{-}1}}) = [\nabla_{\boldsymbol{x}} \mathbf{h}^{\mathrm{T}}(\boldsymbol{x})]^{\mathrm{T}}_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{\mathrm{n|n\text{-}1}}}, \tag{2.15}$$

is defined in [43]. This Jacobian must be used to perform the Cartesian-to-spherical transformation on the Cartesian state covariance matrix, as shown in Equation (2.20). The vector $\boldsymbol{w}_{\mathrm{n}}$ is a Gaussian random vector,

$$\boldsymbol{w}_{\mathrm{n}} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{R}), \tag{2.16}$$

with a mean vector of zeroes, and a covariance matrix $\boldsymbol{R}$ previously defined in (2.12).

2.2.2. Cartesian EKF Algorithm

With the basic computations developed, it is possible to construct a discrete-time C-EKF process [43]. The subscript $n|m$ indicates an estimate at time n, conditioned on the observation data up to time m. This convention will be used for all Kalman filter-based algorithms, where applicable. The C-EKF algorithm is shown in Table 3, and leverages the same dynamic transition matrix $\boldsymbol{F}$, observation prediction function $\mathbf{h}(\boldsymbol{x})$, and observation prediction Jacobian matrix described in Section 2.2.1.

Table 3: C-EKF Algorithm

State Prediction	$\hat{\boldsymbol{x}}_{\mathrm{n\|n\text{-}1}} = \boldsymbol{F}\hat{\boldsymbol{x}}_{\mathrm{n\text{-}1\|n\text{-}1}}$	(2.17)
State Covariance	$\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n\text{-}1}} = \boldsymbol{F}\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\text{-}1\|n\text{-}1}}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}$	(2.18)
Observation Prediction	$\hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}} = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{\mathrm{n\|n\text{-}1}}\right)$	(2.19)
Observation Covariance	$\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n\|n\text{-}1}} = \hat{\boldsymbol{H}}\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n\text{-}1}}\hat{\boldsymbol{H}}^{\mathrm{T}} + \boldsymbol{R}$	(2.20)
State-Observation Covariance	$\boldsymbol{P}^{\boldsymbol{xz}}_{\mathrm{n\|n\text{-}1}} = \boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n\text{-}1}}\hat{\boldsymbol{H}}^{\mathrm{T}}$	(2.21)
Kalman Gain	$\boldsymbol{K}_{\mathrm{n}} = \boldsymbol{P}^{\boldsymbol{xz}}_{\mathrm{n\|n\text{-}1}}\left[\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n\|n\text{-}1}}\right]^{-1}$	(2.22)
State Update	$\hat{\boldsymbol{x}}_{\mathrm{n\|n}} = \hat{\boldsymbol{x}}_{\mathrm{n\|n\text{-}1}} + \boldsymbol{K}_{\mathrm{n}}\left(\boldsymbol{z}_{\mathrm{n}} - \hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}}\right)$	(2.23)
State Covariance	$\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n}} = \boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n\text{-}1}} - \boldsymbol{K}_{\mathrm{n}}\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n\|n\text{-}1}}\boldsymbol{K}_{\mathrm{n}}^{\mathrm{T}}$	(2.24)

2.2.3. Non-Gaussian Behavior in the Cartesian EKF

As part of the study of Kalman filters, it was found that the probability density function (PDF) of the bearing random variable obtained from a bivariate Gaussian distribution and parameterized on the target range has not been computed previously. The material for the derivation of the PDF of bearing is based on the author's published work [35].

In radar target tracking applications [15][17][43], a Kalman filter (KF) is often used to track objects within the field-of-view of a sensor. In this context, sensors typically deliver measurements in spherical or polar coordinates. However, the state vector of an object being tracked (such as an ASM) is typically defined in Cartesian coordinates. This necessitates a Cartesian-to-polar transformation to make predictions for sensor measurements.

The azimuthal angle, or bearing, can be computed using only the coordinates in the east-north (EN) plane; the coordinate system is shown in Figure 13.

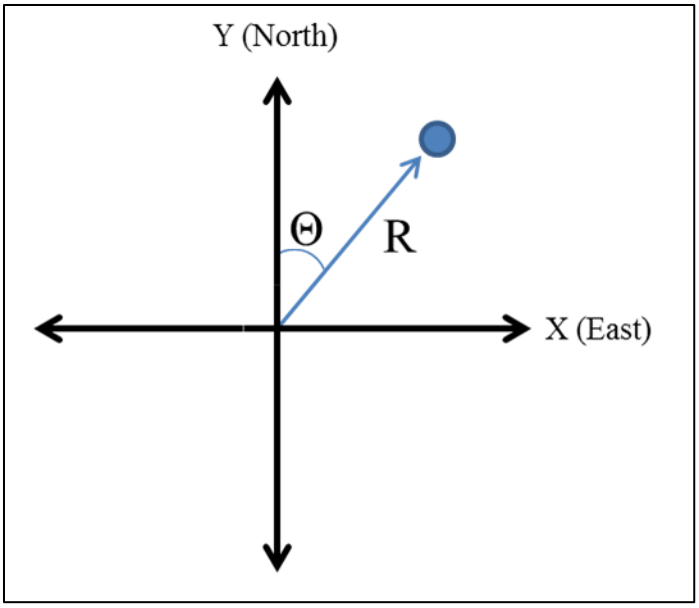

Figure 13: The two dimensional coordinate system used for bearing PDF derivation

The Cartesian-to-polar coordinate transformation requires one to take the inverse tangent

$$\Theta = \arctan\left(\frac{X}{Y}\right), \tag{2.25}$$

where X and Y are elements of a vector $\mathbf{X}$ and compose bivariate Gaussian distribution

$$\boldsymbol{X} \sim \mathrm{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \tag{2.26}$$

where the mean ($\boldsymbol{\mu}$) and covariance ($\boldsymbol{\Sigma}$) are

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_{\mathrm{X}} & \mu_{\mathrm{Y}} \end{bmatrix}^{\mathrm{T}}, \tag{2.27}$$

and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{\mathrm{X}}^2 & \rho\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}} \\ \rho\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}} & \sigma_{\mathrm{Y}}^2 \end{bmatrix}. \tag{2.28}$$

The bearing of the target (Θ) described in (2.25) is a random variable that behaves as a wrapped Gaussian [78] at long range. For the remainder of Section 2.2.3, random variables are referred to with capital letters, while deterministic variables are lower-case.

It will be shown in this section that as the target approaches the origin (collocated with the sensor), the variance increases and the distribution of Θ resembles a uniform (-π,π] random variable. As the behavior becomes less Gaussian, the bearing prediction begins to diverge, which can lead to poor tracking accuracy in the C-EKF. This problem can be ameliorated using a debiased converted measurement filter [1][16][68], which converts the polar measurements to Cartesian coordinates prior to executing the algorithm. Another approach is to maintain the filter state in polar coordinates, as was done in [45] and [46], thereby avoiding any coordinate system conversion. A third alternative is to use observation-only (O2) inference to estimate the state directly from the observations [70].

The Monte Carlo samples of *X* and *Y* for a radially inbound trajectory drawn from the PDF given in Equation (2.26) are shown in the left half of Figure 14. For simplicity, *X* and *Y* are assumed to be uncorrelated with equal variance in this case. Clearly, the bearing samples calculated from the Cartesian samples using (2.25) diverge within a close range, as shown in the right half of Figure 14.

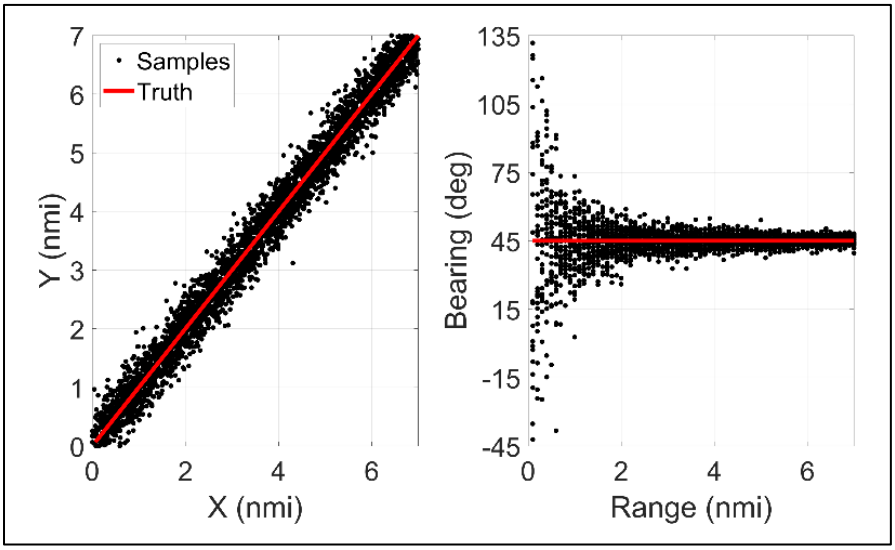

Figure 14: Plot of radially inbound target (45 degrees from North, σ = 0.2 data miles); Cartesian Gaussian samples (left) and computed bearing samples (right).

The available literature has already explored aspects of the distributions resulting from the Cartesian-to-polar transformation [48]. For instance, the Cauchy distribution is also the ratio distribution for two independent, zero-mean Gaussian-distributed random variables. Several authors have performed statistical analyses of the Cauchy distribution [21][79], but the arctangent of a Cauchy distributed random variable only represents the end-game behavior of Θ at zero range and assumes that X and Y are independent.

Other authors have explored the properties of the arctangent distribution and its relationship with the folded standard Gaussian distribution [92]. These results have value, but do not address the more general wrapped Gaussian distribution [78] or its application to target tracking performance.

While we focus on the target bearing, it should be noted that the target range (R) given by

$$R = \sqrt{X^2 + Y^2}, \tag{2.29}$$

where X and Y are drawn from (2.26), is a bivariate non-central chi distribution [64]. This distribution simplifies to the well-known Rice distribution [93] when X and Y are independent, with equal variance σ^2

$$R \sim \text{Rice}\left(\mu_{\text{R}}, \sigma\right), \tag{2.30}$$

where

$$\mu_{\text{R}} = \sqrt{\mu_{\text{X}}^2 + \mu_{\text{Y}}^2}. \tag{2.31}$$

Some authors, for instance Haug, have attempted to derive a density similar to the one desired here. However, the precise PDF for the bearing was not properly determined in [43][44]. Improper use of the direct transformation given in (2.25) causes a loss of sign information when taking the quotient of the two variables. In addition, the inverse tangent

function is periodic over (-π/2,π/2) and not the full circle. Mallick [77] pointed out in a recent note that the true transformation used in many tracking applications is the four-quadrant inverse tangent, despite many books, journals, and conferences using the measurement function, as in (2.25). In practice, a four-quadrant inverse tangent (such as that used in MATLAB) is used.

Using the two-quadrant inverse tangent, as Haug did in [43] and [44], results in PDFs with peaks centered on the true bearing as well as the true bearing ±π, as shown in Figure 3. Further analysis shows that the PDF integrates to 2 over the full support, disqualifying it as a true PDF.

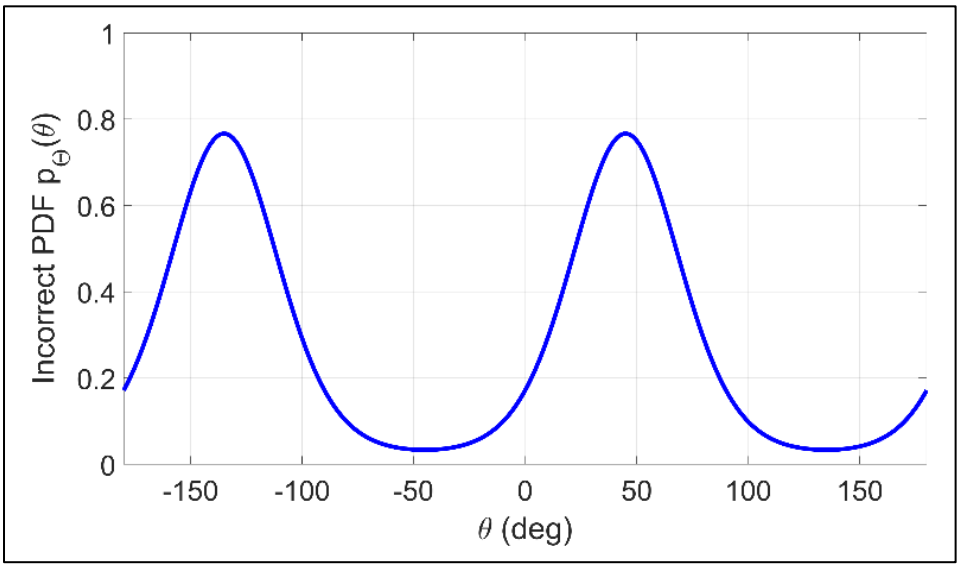

Figure 15: Incorrect application of inverse tangent function to obtain PDF for Θ results in double peaks ($\mu_X = \mu_Y = 3$ and $\sigma = 2$ for this example).

This study used simple polar relations and variable transformations to obtain the PDF of a bearing random variable derived from a Cartesian to polar transformation. The PDF for the bearing can be used to assess the viability of the Gaussian assumption for the bearing as a function of range.

The development of the PDF for bearing begins with the polar-to-Cartesian transformations of R and Θ into X and Y in an EN coordinate system, given by

$$X = R\sin(\Theta), \tag{2.32}$$

and

$$Y = R\cos(\Theta). \tag{2.33}$$

For convenience later in the derivation, it was helpful to redefine the mean values of X and Y in terms of their polar equivalents. Specifically, we define a mean target range μ_{R} and mean bearing μ_{Θ} such that

$$\mu_{\mathrm{X}} = \mu_{\mathrm{R}}\sin\left(\mu_{\Theta}\right), \tag{2.34}$$

and

$$\mu_{\mathrm{Y}} = \mu_{\mathrm{R}}\cos\left(\mu_{\Theta}\right), \tag{2.35}$$

with

$$\mu_{\Theta} = \arctan\left(\frac{\mu_{\mathrm{X}}}{\mu_{\mathrm{Y}}}\right). \tag{2.36}$$

The joint distribution for X and Y can be written as

$$\mathrm{f}_{\mathrm{XY}}(x,y) = \frac{1}{2\pi\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}\sqrt{1-\rho^2}}\exp\left[-\frac{\mathrm{z}(x,y)}{2\left(1-\rho^2\right)}\right], \tag{2.37}$$

where

$$\mathrm{z}(x,y) \equiv \frac{\left(x-\mu_{\mathrm{X}}\right)^2}{2\sigma_{\mathrm{X}}^2} + \frac{\left(y-\mu_{\mathrm{Y}}\right)^2}{2\sigma_{\mathrm{Y}}^2} - \frac{2\rho\left(x-\mu_{\mathrm{X}}\right)\left(y-\mu_{\mathrm{Y}}\right)}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}}. \tag{2.38}$$

Substituting (2.32)-(2.36) into (2.37) yields

$$\mathrm{f}_{\mathrm{XY}}\left(\mathrm{x}(r,\theta),\mathrm{y}(r,\theta)\right) = \frac{1}{2\pi\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}\sqrt{1-\rho^2}}\exp\left[-\frac{\mathrm{a}(\theta)r^2 - \mathrm{b}(\theta)r + c}{d}\right] \tag{2.39}$$

where

$$\mathrm{a}(\theta)=\frac{\sin^2(\theta)}{\sigma_{\mathrm{X}}^2}+\frac{\cos^2(\theta)}{\sigma_{\mathrm{Y}}^2}-\frac{\rho\sin(2\theta)}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}}; \tag{2.40}$$

$$\mathrm{b}(\theta)=2\mu_{\mathrm{R}}\left(\frac{\sin(\mu_{\Theta})\sin(\theta)}{\sigma_{\mathrm{X}}^2}+\frac{\cos(\mu_{\Theta})\cos(\theta)}{\sigma_{\mathrm{Y}}^2}-\frac{\rho\sin(\theta+\mu_{\Theta})}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}}\right); \tag{2.41}$$

$$c=\mu_{\mathrm{R}}^2\left(\frac{\sin^2(\mu_{\Theta})}{\sigma_{\mathrm{X}}^2}+\frac{\cos^2(\mu_{\Theta})}{\sigma_{\mathrm{Y}}^2}-\frac{\rho\sin(2\mu_{\Theta})}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}}\right); \tag{2.42}$$

$$d=2\left(1-\rho^2\right). \tag{2.43}$$

A change of variable [22] can now be performed to obtain the joint distribution in polar coordinates, $\mathrm{f}_{\mathrm{R}\Theta}(r,\theta)$:

$$\mathrm{f}_{\mathrm{R}\Theta}(r,\theta)=\left|\mathbf{J}(r,\theta)\right|\mathrm{f}_{\mathrm{XY}}\left(\mathrm{x}(r,\theta),\mathrm{y}(r,\theta)\right) \tag{2.44}$$

where $\mathbf{J}(r,\theta)$ is the Jacobian [89] of the transformation equations (2.32)–(2.33) given by

$$\mathbf{J}(r,\theta)=\begin{vmatrix}\dfrac{\partial(r\sin(\theta))}{\partial r} & \dfrac{\partial(r\sin(\theta))}{\partial\theta}\\ \dfrac{\partial(r\cos(\theta))}{\partial r} & \dfrac{\partial(r\cos(\theta))}{\partial\theta}\end{vmatrix}=-r. \tag{2.45}$$

This leads to

$$\mathrm{f}_{\mathrm{R}\Theta}(r,\theta)=\frac{r}{2\pi\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}\sqrt{1-\rho^2}}\exp\left[-\frac{1}{d}\left(\mathrm{a}(\theta)r^2-\mathrm{b}(\theta)r+c\right)\right]. \tag{2.46}$$

The joint PDF for R and Θ can be simplified by completing the square in the argument of the exponential to obtain

$$\mathrm{f}_{\mathrm{R}\Theta}(r,\theta)=\frac{r}{2\pi\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}\sqrt{1-\rho^2}}\times \exp\left[c-\frac{\mathrm{b}^2(\theta)}{4\mathrm{a}(\theta)}\right]\exp\left[-\frac{\mathrm{a}(\theta)}{d}\left(r-\frac{\mathrm{b}(\theta)}{\mathrm{a}(\theta)}\right)^2\right]. \tag{2.47}$$

To obtain the marginal PDF of Θ, we simply integrate (2.47) over the support of R, resulting in

$$\begin{aligned}\mathrm{f}_{\Theta}(\theta)&=\int_0^{\infty}\mathrm{f}_{\mathrm{R}\Theta}(r,\theta)\mathrm{d}r\\&=\frac{1}{2\pi\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}\sqrt{1-\rho^2}}\times \exp\left[\frac{c}{d}-\frac{\mathrm{b}^2(\theta)}{4\mathrm{a}(\theta)d}\right]\int_0^{\infty}r\exp\left[-\frac{\mathrm{a}(\theta)}{d}\left(r-\frac{\mathrm{b}(\theta)}{\mathrm{a}(\theta)}\right)^2\right]\mathrm{d}r.\end{aligned} \tag{2.48}$$

Using standard integral methods or referring to [40], we can obtain the marginal distribution for Θ

$$\mathrm{f}_{\Theta}(\theta)=\frac{1}{2\pi\mathrm{a}(\theta)\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}\sqrt{1-\rho^2}}\exp\left(\frac{c}{d}-\frac{\mathrm{b}^2(\theta)}{4\mathrm{a}(\theta)d}\right)\left[\frac{d}{2}\exp\left(\frac{\mathrm{b}^2(\theta)}{\mathrm{a}(\theta)d}\right)+\mathrm{b}(\theta)\sqrt{\frac{\pi d}{\mathrm{a}(\theta)}}\Phi\left(\mathrm{b}(\theta)\sqrt{\frac{2}{\mathrm{a}(\theta)\mathrm{d}}}\right)\right], \tag{2.49}$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the standard Gaussian distribution N(0,1).

When X and Y are uncorrelated ($\rho = 0$) with equal variance ($\sigma_{\mathrm{X}} = \sigma_{\mathrm{Y}} = \sigma$), equation (2.49) simplifies to

$$\mathrm{f}_{\Theta}(\theta)=\frac{1}{2\pi}\exp\left[-\frac{\mu_{\mathrm{R}}^2}{2\sigma^2}\right]+\frac{\alpha(\theta)}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{\mu_{\mathrm{R}}^2-\alpha(\theta)^2}{2\sigma^2}\right]\cdot\Phi\left(\frac{\alpha(\theta)}{\sigma}\right), \tag{2.50}$$

where

$$\alpha(\theta) \equiv \mu_R \cos\left(\theta - \mu_\Theta\right). \tag{2.51}$$

The asymptotic behaviors of the general PDF (2.49) and simplified PDF (2.50) are discussed next.

The mixed uniform and Gaussian behavior of the distribution given in Equation (2.50) may not be immediately obvious. Thus, by observing the asymptotic behavior of each term in with respect to μ_R. The first term of the sum in (2.50) is $f_\Theta^{(1)}(\theta)$ and the second term is $f_\Theta^{(2)}(\theta)$. Let

$$f_\Theta^{(1)}(\theta) = \frac{1}{2\pi}\exp\left[-\frac{\mu_R^2}{2\sigma^2}\right]. \tag{2.52}$$

Now it is more easily shown that

$$\lim_{\mu_R \to 0} \frac{1}{2\pi}\exp\left[-\frac{\mu_R^2}{2\sigma^2}\right] = \frac{1}{2\pi} \tag{2.53}$$

and

$$\lim_{\mu_R \to \infty} \frac{1}{2\pi}\exp\left[-\frac{\mu_R^2}{2\sigma^2}\right] = 0. \tag{2.54}$$

Thus, as μ_R approaches zero, $f_\Theta^{(1)}(\theta)$ becomes a wrapped uniform PDF over $(-\pi,\pi]$, whereas as μ_R approaches infinity, $f_\Theta^{(1)}(\theta)$ vanishes.

Now examine the second term in (2.50) and define

$$f_\Theta^{(2)}(\theta) = \frac{\alpha(\theta)}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{\mu_R^2 - \alpha(\theta)^2}{2\sigma^2}\right]\cdot\Phi\left(\frac{\alpha(\theta)}{\sigma}\right). \tag{2.55}$$

From (2.51) it is clear that $\alpha(\theta)$ approaches zero as μ_R approaches zero because $\alpha(\theta)$ is linear with respect to μ_R. Therefore, the close range limit of (2.55) is

$$\lim_{\mu_R \to 0} \frac{\alpha(\theta)}{\sqrt{2\pi}\sigma}\exp\left[-\frac{\mu_R^2 - \alpha(\theta)^2}{2\sigma^2}\right]\cdot\Phi\left(\frac{\alpha(\theta)}{\sigma}\right) = 0. \tag{2.56}$$

Thus

$$\lim_{\mu_R \to 0} f_\Theta^{(2)}(\theta) = 0, \tag{2.57}$$

and the result for the distribution when X and Y are uncorrelated with equal variance is

$$\lim_{\mu_R \to 0} f_\Theta(\theta) \to U(-\pi,\pi). \tag{2.58}$$

The standard normal CDF also exhibits simple behavior in the limiting cases, such that it approaches constant scaling factors

$$\lim_{\mu_R \to 0} \Phi\left(\frac{\alpha(\theta)}{\sigma}\right) = \frac{1}{2} \tag{2.59}$$

and

$$\lim_{\mu_R \to \infty} \Phi\left(\frac{\alpha(\theta)}{\sigma}\right) = 1. \tag{2.60}$$

As μ_R increases, the influence of $f_\Theta^{(2)}(\theta)$ becomes more pronounced. Taking the Taylor series expansion of the leading term in (2.55) about μ_Θ yields

$$\frac{\alpha(\theta)}{\sigma\sqrt{2\pi}} = \frac{\mu_R}{\sigma\sqrt{2\pi}}\left[1 - \frac{1}{2}(\theta - \mu_\Theta)^2 + \ldots\right]. \tag{2.61}$$

For θ near μ_Θ (2.61) reduces to

$$\frac{\alpha(\theta)}{\sqrt{2\pi}\sigma} \approx \frac{\mu_R}{\sqrt{2\pi}\sigma}. \tag{2.62}$$

Similarly, the exponential term in (2.55) becomes

$$\frac{\mu_R^2 - \alpha(\theta)^2}{2\sigma^2} \approx \frac{\mu_R^2}{2\sigma^2}(\theta - \mu_\Theta)^2. \tag{2.63}$$

By applying the approximations in (2.62) and (2.63) to (2.55) we can find the limit of $f_\Theta^{(2)}(\theta)$ as μ_R approaches infinity:

$$\lim_{\mu_R \to \infty} f_\Theta^{(2)}(\theta) \approx \frac{1}{\sqrt{2\pi}\left(\frac{\sigma}{\mu_R}\right)} \exp\left[-\frac{(\theta-\mu_\Theta)^2}{2\left(\frac{\sigma}{\mu_R}\right)^2}\right]. \tag{2.64}$$

Because $f_\Theta^{(1)}(\theta)$ vanishes for large μ_R and $f_\Theta^{(2)}(\theta)$ can be approximated using (2.64), the PDF of bearing at long ranges can be written as a (wrapped) Gaussian distribution

$$\lim_{\mu_R \to \infty} f_\Theta(\theta) \to \mathrm{N}\left(\mu_\Theta, \left(\frac{\sigma}{\mu_R}\right)^2\right). \tag{2.65}$$

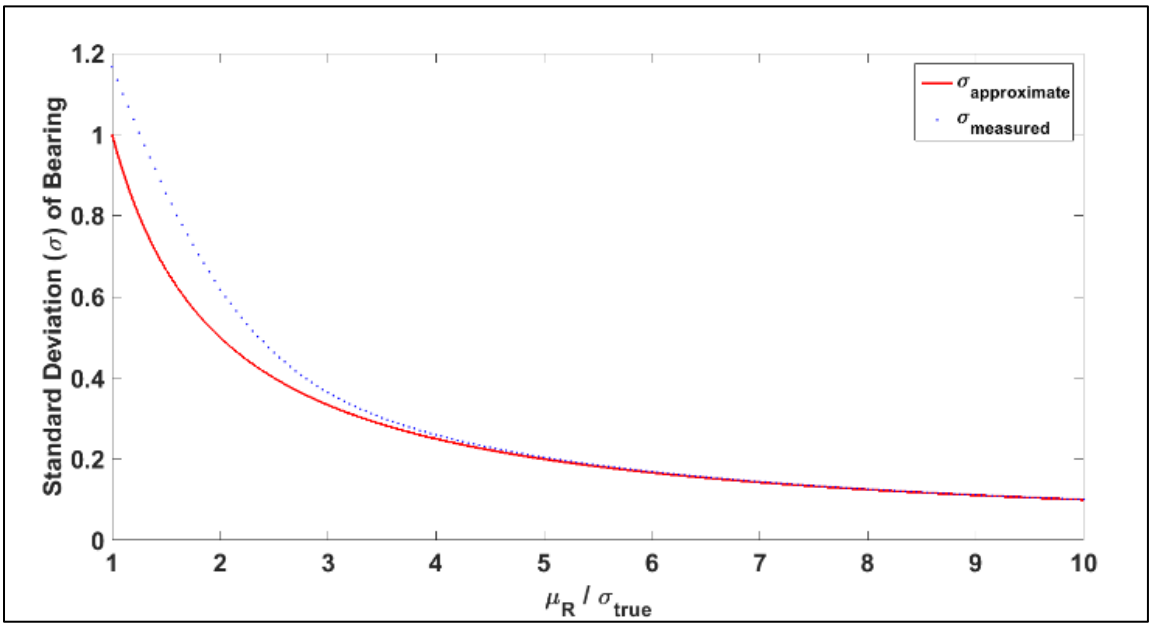

Figure 16: Plot depicting sample standard deviation and its approximation found by taking the square root of the variance for the wrapped Gaussian distribution

A simple test can be used to demonstrate that the PDF of Θ behaves like the wrapped Gaussian distribution in (2.65). Figure 16 shows that as the ratio μ_R/σ increases (where the true σ is constant), the sample standard deviation (represented by the blue dots in Figure 16) quickly converges to the approximated standard distribution. From the plot in Figure 16, the difference becomes indistinguishable at approximately $\mu_R/\sigma = 5$. The approximated standard deviation was obtained by taking the square root of the variance

from (43), whereas the true standard deviation refers to the standard deviation of the samples found on the right-hand side of Figure 14.

A plot of the simplified bearing PDF from Equation (2.50) for a variety of bearing values and mean ranges is shown in Figure 17. A careful inspection of the plot in Figure 17 points to the distribution being a wrapped Gaussian centered on the nominal bearing (μ_Θ = 45 °) when μ_R is large. It can be seen that as μ_R decreases the PDF transitions from that of a wrapped Gaussian with support (-π,π] to uniform over the same support.

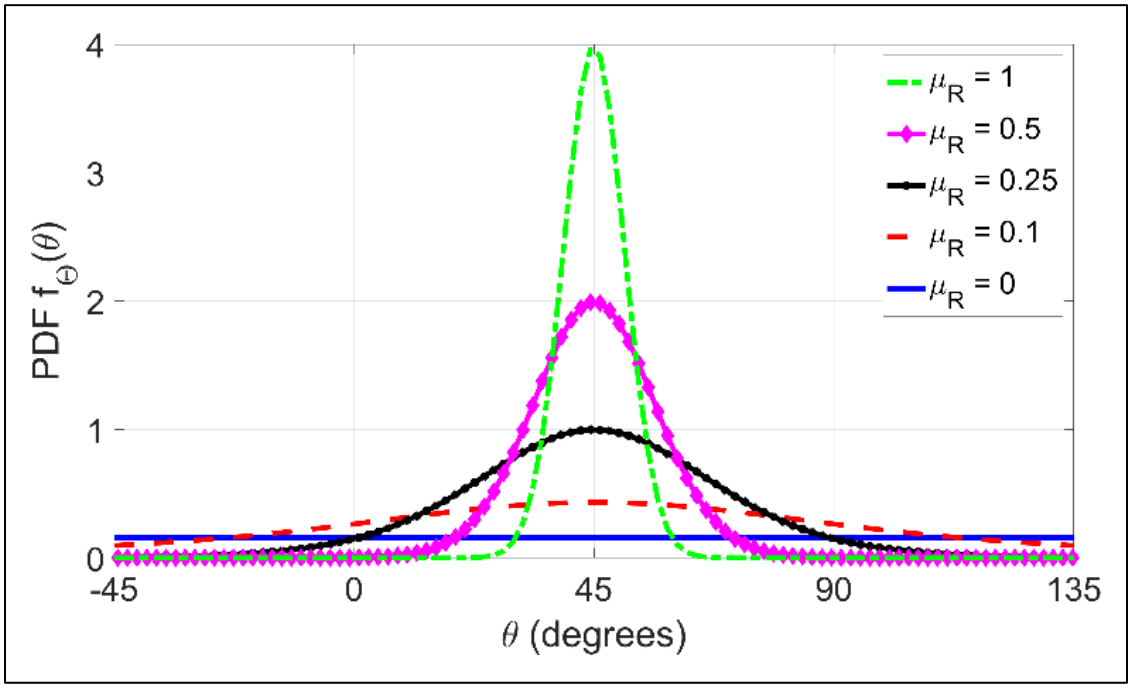

Figure 17. Plot of the bearing PDF for various values of μ_R with constant σ = 0.1

The asymptotic behavior of the generalized version of the distribution in (2.49) is not as convenient as that of the simplified version displayed in Figure 17. At long range, the distribution is somewhat similar to a wrapped Gaussian, but for the sake of simplicity, here we focus on the behavior of the distribution at close range. The key takeaway is related to the expected value of the distribution

$$\lim_{\mu_R \to 0} \mathrm{E}[\Theta] = 0. \tag{2.66}$$

The conclusion drawn in (2.66) is clear for the simple case presented in (2.58) when the distribution approaches uniform over (–π, π]. Determining the expected value for a general distribution requires additional computations.

The limiting behavior for each of the scalar values in (2.40)–(2.43) is simple to determine,

$$\lim_{\mu_R \to 0} \mathrm{a}(\theta) = \frac{\sin^2(\theta)}{\sigma_X^2} + \frac{\cos^2(\theta)}{\sigma_Y^2} - \frac{\rho \sin(2\theta)}{\sigma_X \sigma_Y} \tag{2.67}$$

$$\lim_{\mu_R \to 0} \mathrm{b}(\theta) = 0, \tag{2.68}$$

$$\lim_{\mu_R \to 0} c = 0, \tag{2.69}$$

$$\lim_{\mu_R \to 0} d = 2\left(1 - \rho^2\right). \tag{2.70}$$

At close range the general distribution becomes

$$\lim_{\mu_R \to 0} \mathrm{f}_\Theta(\theta) = \frac{d}{4\pi \mathrm{a}(\theta) \sigma_X \sigma_Y \sqrt{1-\rho^2}}. \tag{2.71}$$

The right-hand side of (2.71) is a function of θ because of the presence of a(θ), and does not represent a uniform distribution unless σ_X and σ_Y are equal and uncorrelated. However, simple trigonometric identities can be used to show that a(θ) is twice periodic over (–π,π)]. Thus, Equation (2.66) also applies to the generalized version of the distribution Equation (2.49). This result can also be computed explicitly using the method to find the moments of wrapped random variables [78]. An example of a distribution in the zero range is shown in Figure 18.

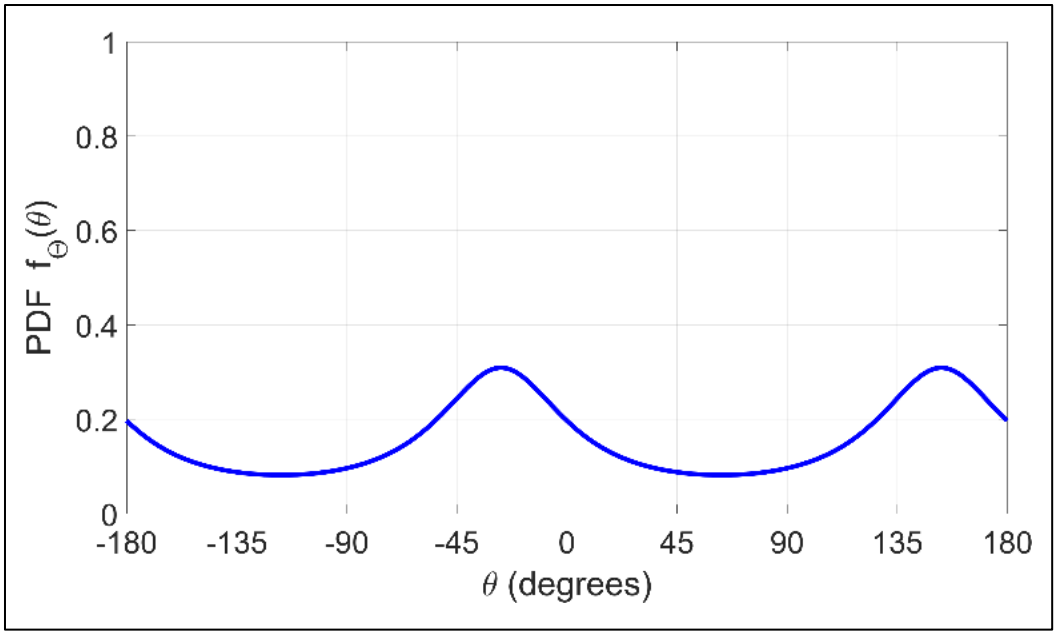

Figure 18. Plot of the bearing PDF at $\mu_R = 0$ with $\sigma_X = 0.7$, $\sigma_Y = 1.0$, and $\rho = -0.5$

Curiously, the distribution for the case in Figure 18 resembles the incorrect distribution shown in Figure 15. However, the dual peaks in Figure 15 are due to the ambiguity introduced by the arctangent function in general, while the peaks in Figure 18 are artifacts caused by a(θ). This distribution also integrates to unit area, unlike the incorrect distribution presented earlier. Next, we discuss the impact of these findings on the Kalman filter.

The Kalman filter algorithm is well known to perform optimally for linear functions of Gaussian distributed variables [58]. A Cartesian KF which relies on measurements in polar coordinates, clearly violates the linearity requirement because of the nonlinear functions (2.25) and (2.29).

The state model for the two-dimensional Cartesian is described by

$$\boldsymbol{x}_n = \boldsymbol{f}\left(\boldsymbol{x}_{n-1}\right) + \boldsymbol{v}_{n-1}, \tag{2.72}$$

where the vector $\boldsymbol{v}$ is zero mean Gaussian distributed with covariance matrix $\boldsymbol{Q}$.

The state vector (including velocity terms) is assumed to be

$$\boldsymbol{x}_{\mathrm{n}} = \begin{bmatrix} x_{\mathrm{n}} & y_{\mathrm{n}} & \dot{x}_n & \dot{y}_n \end{bmatrix}^T \tag{2.73}$$

where x_{n} and y_{n} represent the Cartesian position of the target at time n and their dotted counterparts represent the Cartesian velocities.

Using the constant velocity (CV) assumption for the state model, the vector function $\mathbf{f}(\boldsymbol{x})$ is linear,

$$\boldsymbol{f}\left(\boldsymbol{x}_{\mathrm{n-1}}\right) = \boldsymbol{F}\boldsymbol{x}_{\mathrm{n-1}}. \tag{2.74}$$

The state transition matrix $\boldsymbol{F}$ used in the two dimensional case simplifies from (2.3) to

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.75}$$

where T is the time difference between measurement updates.

The state prediction ($\hat{\boldsymbol{x}}$) and error covariance ($\boldsymbol{P}^{\mathrm{xx}}$) predictions at time n based on observations up to time n-1 are also linear for the CV model:

$$\hat{\boldsymbol{x}}_{\mathrm{n|n-1}} = \boldsymbol{F}\hat{\boldsymbol{x}}_{\mathrm{n-1|n-1}}, \tag{2.76}$$

$$\boldsymbol{P}^{\mathrm{xx}}_{n|n-1} = \boldsymbol{F}\boldsymbol{P}^{\mathrm{xx}}_{n-1|n-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}_n. \tag{2.77}$$

where $\boldsymbol{Q}_{\mathrm{n}}$ is the process noise matrix. Assuming that the target is equally maneuverable in all directions and the measurement update period is constant, the process noise matrix becomes

$$\boldsymbol{Q} = \begin{bmatrix} \frac{q(T)^3}{3} & 0 & \frac{q(T)^2}{2} & 0 \\ 0 & \frac{q(T)^3}{3} & 0 & \frac{q(T)^2}{2} \\ \frac{q(T)^2}{2} & 0 & q(T) & 0 \\ 0 & 0 & 0 & q(T) \end{bmatrix}, \tag{2.78}$$

where q is the process noise intensity. In practice, the filter developer selects the process noise intensity to account for the uncertainty of the dynamic model. This tuning parameter directly influences the position variance terms on the main diagonal of the predicted covariance (2.77).

The observation model is

$$\boldsymbol{z}_{\mathrm{n}} = \boldsymbol{h}\left(\boldsymbol{x}_{\mathrm{n}}\right) + \boldsymbol{w}_{\mathrm{n}}, \tag{2.79}$$

where the vector $\boldsymbol{w}$ is Gaussian distributed with zero mean and covariance matrix $\boldsymbol{S}$ and is independent of $\boldsymbol{v}$. The covariance matrix for the observations is often treated as independent noise on the range and bearing measurements,

$$\boldsymbol{S} = \begin{bmatrix} \sigma_{\mathrm{R}}^2 & 0 \\ 0 & \sigma_{\Theta}^2 \end{bmatrix}. \tag{2.80}$$

The observation model is the same pair of nonlinear functions discussed previously

$$\mathbf{h}\left(\boldsymbol{x}_{\mathrm{n}}\right) = \begin{bmatrix} r_{\mathrm{n}} \\ \theta_{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} \sqrt{x_{\mathrm{n}}^2 + y_{\mathrm{n}}^2} \\ \arctan\left(x_{\mathrm{n}} / y_{\mathrm{n}}\right) \end{bmatrix} \tag{2.81}$$

It has already been shown that the resulting distribution for the bearing will appear Gaussian at long range and progressively less Gaussian as μ_{R} shrinks. Given the

knowledge of the probability density of the target state for all prior measurements $p(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})$, the predicted bearing approaches zero at close range,

$$\hat{\boldsymbol{z}}_{n|n-1} = \begin{bmatrix} \int_{\Omega_x} \sqrt{x_n^2 + y_n^2}\, p(\boldsymbol{x}_n \mid \boldsymbol{z}_{1:n-1}) d\boldsymbol{x}_n \\ 0 \end{bmatrix}. \tag{2.82}$$

This means that when the typical Kalman filter equations [17][43] are used, the observation prediction is

$$\hat{\boldsymbol{z}}_{n|n-1} = \int_{\Omega_x} \left(\mathbf{h}(\boldsymbol{x}_n) + \boldsymbol{w}_n\right) p(\boldsymbol{x}_n \mid \boldsymbol{z}_{1:n-1})\, d\boldsymbol{x}_n, \tag{2.83}$$

where $p(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})$ is the probability distribution for the state given observations up to time n-1. The observations for the tracking filter of interest are range and bearing, thus

$$\mathbf{h}(\boldsymbol{x}_n) = \begin{bmatrix} r_n \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} \\ \arctan(x_n / y_n) \end{bmatrix}. \tag{2.84}$$

It has already been shown that the expected value (2.66) of bearing in (2.84) approaches zero at close range. This means that, with knowledge of the probability density of the state given all prior measurements $p(\boldsymbol{x}_n|\boldsymbol{z}_{1:n-1})$, the predicted bearing approaches zero at close range

$$\hat{\boldsymbol{z}}_{n|n-1} = \begin{bmatrix} \int_{\Omega_x} \sqrt{x_n^2 + y_n^2}\, p(\boldsymbol{x}_n \mid \boldsymbol{z}_{1:n-1}) d\boldsymbol{x}_n \\ 0 \end{bmatrix}. \tag{2.85}$$

The residual vector $\boldsymbol{\xi}_n$ for the filter is

$$\boldsymbol{\xi}_n = \boldsymbol{z}_n^o - \hat{\boldsymbol{z}}_{n|n-1}, \tag{2.86}$$

using the latest observation data, $\boldsymbol{z}_n^o$. The residual in the bearing dimension increases as the bearing predictions become uniformly randomly distributed. Residual growth of this

nature is indicative of a model that is poorly matched to the tracking problem [42] and may result in poor tracking performance.

Another key takeaway is that increasing the process noise has the same effect as decreasing μ_R and vice versa because of its direct impact on the variance terms in Equations (2.40)–(2.42). This means that a filter designer can unintentionally degrade the accuracy of their EKF tracker if they are not judicious in their selection of the tuning parameter q given their measurement update period constraints.

In many target tracking applications, target maneuvers are not known a priori; therefore, they are often modeled as random accelerations in the CV model. A filter designer focused on tracking a maneuvering inbound target must include appropriate levels of process noise to account for these maneuvers based on knowledge of the target's maneuverability. However, a combination of large process noise and closing the target range can have a negative impact on the filter performance owing to the combination of the nonlinear and non-Gaussian behavior of the bearing model.

Many have chosen to address the nonlinearity of $\mathbf{h}(\boldsymbol{x}_n)$ for mixed Cartesian and polar tracking problems using the EKF [46][63][96]. In the EKF formulation, the estimate of (2.79) can be simplified to

$$\hat{\boldsymbol{z}}_{n|n-1} = \mathbf{h}\left(\hat{\boldsymbol{x}}_{n|n-1}\right). \tag{2.87}$$

However, linearizing the filter regarding the state and covariance does not address the fact that, at a close range, the arctangent function is still quite nonlinear.

To demonstrate the effects of range and process noise on tracking performance, a series of simple experiments were performed using a CV Cartesian EKF to track a simulated radially inbound target (approaching the sensor origin at a 45° angle clockwise

from north) with range and bearing measurements. RMS position error statistics were collected over the course of 100 Monte Carlo runs, where measurements were generated by adding Gaussian random errors consistent with Equation (2.80) to the truth trajectory. Figure 19 shows the impact of the nonlinearity of the bearing at close range, as well as the impact of varying the process noise.

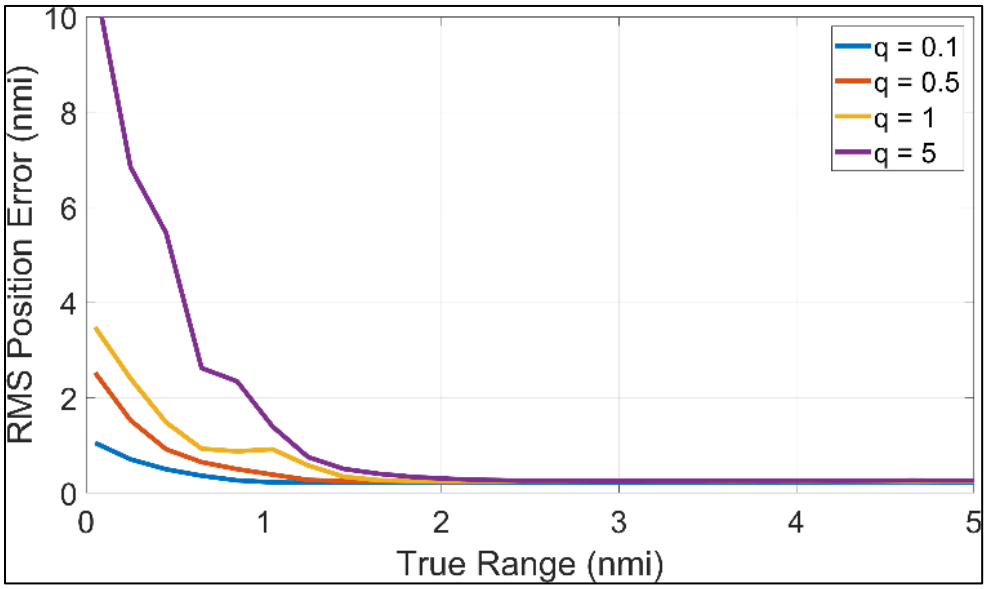

Figure 19. RMS position error plot for radially inbound target using various process noise intensity values (with units of dmi^2/sec^3), Δt = 1 second, σ_R = 0.25 dmi, and σ_θ = 1 degree

The results of this simple experiment were in agreement with the findings of Haug and Williams [46]. In their study, the error performance for a variety of Cartesian and spherical tracking filters showed that, at close range, the tracks that relied on a Cartesian-to-spherical transformation tended to diverge, unlike the fully spherical filters. The results are also in agreement with the findings of Lerro and Bar-Shalom [68], who showed that a debiased converted measurement filter offers superior error and consistency performance compared with the Cartesian EKF.

A study on this state error growth for a constant velocity model was performed by Haug and Williams [46]. The error performance for a variety of Cartesian and spherical

tracking filters shows that at close range, the tracks that rely on a Cartesian-to-spherical transformation tend to diverge, unlike the fully spherical filters. These results apply directly to the NRI target case, as we will see in the performance evaluation of the set of legacy filters when compared to the EPC EKF in Section 2.6.

2.3. The Spherical EKF

2.3.1. Spherical EKF Model

In Section 2.2, a C-EKF was developed, and some of its weaknesses were highlighted. Next, it is necessary to lay out the S-EKF from [18],[43], and[46] as its dynamic model will lay important groundwork for developing the EPC EKF. The essential components of this filter are developed in this section, and some are modified in the next section to implement the EPC filters.

The CV Spherical EKF tracks an object with the state vector

$$\boldsymbol{x} = \begin{bmatrix} r & \dot{r} & \theta & \dot{\theta} & \varphi & \dot{\varphi} \end{bmatrix}^T. \tag{2.88}$$

For a CV target, the spherical coordinate system must rotate, leading to pseudo-accelerations that appear in [23] [18] [105],

$$\begin{aligned} \boldsymbol{a} = &\left[\ddot{r} - \dot{\varphi}^2 r - r\dot{\theta}^2 \cos^2(\varphi) \right] \hat{\boldsymbol{r}} + \\ &\left[r\ddot{\theta}\cos(\varphi) + 2\dot{r}\dot{\theta}\cos(\varphi) - 2\dot{\varphi}\dot{\theta}r\sin(\varphi) \right] \hat{\boldsymbol{\theta}} + \\ &\left[\ddot{\varphi} r - r\dot{\theta}^2 \cos(\varphi)\sin(\varphi) + 2\dot{\varphi}\dot{r} \right] \hat{\boldsymbol{\varphi}}. \end{aligned} \tag{2.89}$$

Note that ($\hat{\boldsymbol{r}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\varphi}}$) represent the spherical coordinate unit vectors.

For a CV target, all components of acceleration must be set to zero. Rearranging the acceleration terms and setting them to zero yields

$$0 = \ddot{r} - \dot{\varphi}^2 r - r\dot{\theta}^2 \cos^2(\varphi), \tag{2.90}$$

$$0 = r\ddot{\theta}\cos(\varphi) + 2\dot{r}\dot{\theta}\cos(\varphi) - 2\dot{\varphi}\dot{\theta}r\sin(\varphi), \tag{2.91}$$

$$0 = \ddot{\varphi}r + r\dot{\theta}^2\cos(\varphi)\sin(\varphi) + 2\dot{\varphi}\dot{r}. \tag{2.92}$$

Solving the equations from (2.90)-(2.92) with respect to the remaining acceleration terms yields

$$\ddot{r} = \dot{\varphi}^2 r + r\dot{\theta}^2\cos^2(\varphi), \tag{2.93}$$

$$\ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r} + 2\dot{\varphi}\dot{\theta}\tan(\varphi) \tag{2.94}$$

$$\ddot{\varphi} = \dot{\theta}^2\cos(\varphi)\sin(\varphi) - \frac{2\dot{\varphi}\dot{r}}{r}. \tag{2.95}$$

For each equation from (2.93), time-based Taylor series expansions can be performed to obtain the dynamic equations of each element in the state vector [43]. The results are shown in Equations (2.96)–(2.101).

$$r_{\mathrm{n}} = r_{\mathrm{n\text{-}1}} + T\dot{r}_{\mathrm{n\text{-}1}} + \frac{T^2}{2} r_{\mathrm{n\text{-}1}}\left[\dot{\theta}^2_{\mathrm{n\text{-}1}}\cos^2\varphi_{\mathrm{n\text{-}1}} + \dot{\varphi}^2_{\mathrm{n\text{-}1}}\right] \tag{2.96}$$

$$\dot{r}_{\mathrm{n}} = \dot{r}_{\mathrm{n\text{-}1}} + Tr_{\mathrm{n\text{-}1}}\left[\dot{\theta}^2_{\mathrm{n\text{-}1}}\cos^2\varphi_{\mathrm{n\text{-}1}} + \dot{\varphi}^2_{\mathrm{n\text{-}1}}\right] \tag{2.97}$$

$$\theta_{\mathrm{n}} = \theta_{\mathrm{n\text{-}1}} + T\dot{\theta}_{\mathrm{n\text{-}1}} - T^2\dot{\theta}_{\mathrm{n\text{-}1}}\left[\dot{\varphi}_{\mathrm{n\text{-}1}}\tan\varphi_{n-1} - \frac{\dot{r}_{\mathrm{n\text{-}1}}}{r_{\mathrm{n\text{-}1}}}\right] \tag{2.98}$$

$$\dot{\theta}_{\mathrm{n}} = \dot{\theta}_{\mathrm{n\text{-}1}} + 2T\dot{\theta}_{\mathrm{n\text{-}1}}\left[\dot{\varphi}_{\mathrm{n\text{-}1}}\tan\varphi_{n-1} - \frac{\dot{r}_{\mathrm{n\text{-}1}}}{r_{\mathrm{n\text{-}1}}}\right] \tag{2.99}$$

$$\varphi_{\mathrm{n}} = \varphi_{\mathrm{n\text{-}1}} + T\dot{\varphi}_{\mathrm{n\text{-}1}} - \frac{T^2}{2}\left[2\frac{\dot{r}_{n-1}}{r_{n-1}}\dot{\varphi}_{\mathrm{n\text{-}1}} + \dot{\theta}^2_{\mathrm{n\text{-}1}}\sin\varphi_{\mathrm{n\text{-}1}}\cos\varphi_{\mathrm{n\text{-}1}}\right] \tag{2.100}$$

$$\dot{\varphi}_{\mathrm{n}} = \dot{\varphi}_{\mathrm{n\text{-}1}} - T\left[2\frac{\dot{r}_{\mathrm{n\text{-}1}}}{r_{\mathrm{n\text{-}1}}}\dot{\varphi}_{\mathrm{n\text{-}1}} + \dot{\theta}^2_{\mathrm{n\text{-}1}}\sin\varphi_{\mathrm{n\text{-}1}}\cos\varphi_{\mathrm{n\text{-}1}}\right] \tag{2.101}$$

which are dynamic transition equations for the elements of the state vector found in (2.88).

These equations can be further broken down into linear and nonlinear functions of the state vector components. The result from [18] becomes

$$\boldsymbol{x}_{\mathrm{n}} = \mathbf{f}(\boldsymbol{x}_{\mathrm{n-1}}) = \boldsymbol{F}\boldsymbol{x}_{\mathrm{n-1}} + \boldsymbol{G}\mathbf{u}(\boldsymbol{x}_{\mathrm{n-1}}) + \boldsymbol{v}_{\mathrm{n-1}}. \tag{2.102}$$

In equation (2.102), the matrix $\boldsymbol{F}$ is the same as shown in Equation (2.3), as these are the linear terms of the state update equations. The vector $\mathbf{u}(\boldsymbol{x}_{\mathrm{n-1}})$, which contains the nonlinear acceleration terms from (2.90) is shown to be

$$\mathbf{u}(\boldsymbol{x}) = \begin{bmatrix} r\left(\dot{\theta}^2 \cos^2\varphi + \dot{\varphi}^2\right) \\ 2\dot{\theta}\left(\dot{\varphi}\tan\varphi - \dfrac{\dot{r}}{r}\right) \\ -\left(2\dfrac{\dot{r}\dot{\varphi}}{r} + \dot{\theta}^2 \sin\varphi\cos\varphi\right) \end{bmatrix}. \tag{2.103}$$

Note that $\mathbf{u}(\boldsymbol{x}_{\mathrm{n-1}})$ is not an input to the system, but simply a portion of the kinematic model.

The matrix $\mathbf{G}$ is defined as

$$\boldsymbol{G} = \begin{bmatrix} \dfrac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \dfrac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & \dfrac{T^2}{2} \\ 0 & 0 & T \end{bmatrix}. \tag{2.104}$$

The state of the object being tracked is presumed to have an additive noise component, represented by $\boldsymbol{v}_{n-1}$. This noise component is treated as a Gaussian random variable with zero mean and covariance matrix

$$\boldsymbol{Q}_n = \begin{bmatrix} q_r \boldsymbol{Q}_1 & \boldsymbol{0}_2 & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & q_\theta(\boldsymbol{x}_n)\boldsymbol{Q}_1 & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & \boldsymbol{0}_2 & q_\varphi(\boldsymbol{x}_n)\boldsymbol{Q}_1 \end{bmatrix}, \tag{2.105}$$

where $\boldsymbol{0}_2$ and $\boldsymbol{Q}_1$ are the same as those described in Equations (2.5) and (2.8), respectively. q_r, q_θ, and q_φ represent the dynamic acceleration noise variances, which are expected to cause deviations from the true CV trajectory model. These variances are expressed as follows:

$$q_r = (2g_r C)^2, \tag{2.106}$$

$$q_\theta(\boldsymbol{x}_n) = \left(\frac{2g_{h_\perp} C}{r\cos\varphi}\right)^2, \tag{2.107}$$

and

$$q_\varphi(\boldsymbol{x}_n) = \left(\frac{2g_{v_\perp} C}{r}\right)^2. \tag{2.108}$$

These terms are defined using the number of expected accelerations (in g's) in the radial, horizontal cross-range ($h_\perp$), and vertical cross-range ($v_\perp$) directions. The constant C is the same conversion factor that applies the proper measurement units to each variance term (2.10).

2.3.2. Spherical EKF Dynamic Model Jacobian

The Kalman Filter that is being developed using the equations from (2.96) is an EKF that requires

$$\hat{\mathbf{F}}\left(\boldsymbol{x}_{n-1}\right)=\left[\nabla_{\boldsymbol{x}}\mathbf{f}^{\mathrm{T}}\left(\boldsymbol{x}\right)\right]_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{\mathrm{n|n-1}}}^{\mathrm{T}}=\nabla_{\boldsymbol{\rho}}\left[\left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{x}^{\mathrm{T}}+\boldsymbol{G}^{\mathrm{T}}\mathbf{u}^{\mathrm{T}}\left(\boldsymbol{x}\right)\right)\right]_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{n|n-1}}^{\mathrm{T}}, \quad (2.109)$$

where $\hat{\boldsymbol{F}}$ is the dynamic model Jacobian. The EKF allows for the linearization of nonlinear state update equations; as a result, we must take spatial derivatives as part of the linearization process. After simplification, the Jacobian of the dynamic transition equations becomes

$$\hat{\mathbf{F}}\left(\hat{\boldsymbol{x}}_{\mathrm{n|n-1}}\right)=\boldsymbol{F}+\boldsymbol{G}\hat{\mathbf{U}}\left(\hat{\boldsymbol{x}}_{n|n-1}\right) \quad (2.110)$$

where

$$\hat{\mathbf{U}}\left(\hat{\boldsymbol{x}}_{\mathrm{n|n-1}}\right)=\left[\nabla_{\boldsymbol{x}}\mathbf{u}^{\mathrm{T}}(\boldsymbol{x})\right]_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{\mathrm{n|n-1}}}^{\mathrm{T}} \quad (2.111)$$

is the Jacobian of the vector $\mathbf{u}(\boldsymbol{x}_{n-1})$ and $\boldsymbol{F}$ is the same as the matrix shown in (2.3). Because matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ were previously defined, the only component left to compute is

$$\hat{\mathbf{U}}(\boldsymbol{x})=\begin{bmatrix}\dot{\theta}^2\cos^2\varphi+\dot{\varphi}^2 & \frac{2\dot{r}\dot{\theta}}{r^2} & \frac{2\dot{r}\dot{\varphi}}{r^2}\\ 0 & -\frac{2\dot{\theta}}{r} & -\frac{2\dot{\varphi}}{r}\\ 0 & 0 & 0\\ 2r\dot{\theta}\cos^2\varphi & 2(\dot{\varphi}\tan\varphi-\frac{\dot{r}}{r}) & -2\dot{\theta}\sin\varphi\cos\varphi\\ -2r\dot{\theta}^2\sin\varphi\cos\varphi & 2\dot{\theta}\dot{\varphi}\sec^2\varphi & \dot{\theta}^2(\sin^2\varphi-\cos^2\varphi)\\ 2r\dot{\varphi}^2 & 2\dot{\theta}\tan\varphi & -\frac{2\dot{r}}{r}\end{bmatrix}^{\mathrm{T}}. \quad (2.112)$$

2.3.3. Spherical EKF Observation Model

The effort expended to develop the dynamic model resulted in a significantly simpler observation model. Several major elements of the model remain the same as in

the C-EKF case, such as the observation vector from Equation (2.11) and the observation covariance matrix $\boldsymbol{R}$ in Equation (2.12). The main difference is that a coordinate transformation no longer needs to take place to relate the observations and the state; therefore, $\boldsymbol{H}$ becomes

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.113}$$

2.3.4. Spherical EKF Algorithm

Using the information developed in the previous two sections, the CV Spherical EKF equation set can be constructed as shown in Table 4.

Table 4: S-EKF Algorithm

State Prediction	$\hat{\boldsymbol{x}}_{n\|n-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{n-1\|n-1} + \boldsymbol{G}\boldsymbol{u}(\hat{\boldsymbol{x}}_{n-1\|n-1})$	(2.114)
State Covariance	$\boldsymbol{P}^{xx}_{n\|n-1} = \hat{\mathbf{F}}(\hat{\boldsymbol{x}}_{n-1\|n-1})\boldsymbol{P}^{xx}_{n-1\|n-1}\left[\hat{\mathbf{F}}(\hat{\boldsymbol{x}}_{n-1\|n-1})\right]^{T} + \mathbf{Q}(\hat{\boldsymbol{x}}_{n-1\|n-1})$	(2.115)
Observation Prediction	$\hat{\boldsymbol{z}}_{n\|n-1} = \boldsymbol{H}\hat{\boldsymbol{x}}_{n\|n-1}$	(2.116)
Observation Covariance	$\boldsymbol{P}^{zz}_{n\|n-1} = \boldsymbol{H}\boldsymbol{P}^{xx}_{n\|n-1}\boldsymbol{H}^{T} + \boldsymbol{R}$	(2.117)
State-Observation Covariance	$\boldsymbol{P}^{xz}_{n\|n-1} = \boldsymbol{P}^{xx}_{n\|n-1}\boldsymbol{H}^{T}$	(2.118)
Kalman Gain	$\boldsymbol{K}_{n} = \boldsymbol{P}^{xz}_{n\|n-1}\left[\boldsymbol{P}^{zz}_{n\|n-1}\right]^{-1}$	(2.119)
State Update	$\hat{\boldsymbol{x}}_{n\|n} = \hat{\boldsymbol{x}}_{n\|n-1} + \boldsymbol{K}_{n}(\boldsymbol{z}_{n} - \hat{\boldsymbol{z}}_{n\|n-1})$	(2.120)
State Covariance	$\boldsymbol{P}^{xx}_{n\|n} = \boldsymbol{P}^{xx}_{n\|n-1} - \boldsymbol{K}_{n}\boldsymbol{P}^{zz}_{n\|n-1}\boldsymbol{K}_{n}^{T}$	(2.121)

2.4. The EPC EKF

2.4.1. EPC Filtering Background

In the paper by Haug and Ford [45] we introduced the EPC concept, an NRI target is defined as one that is inbound toward the sensor origin and maneuvers about the radial

direction as it closes in on the sensor. According to this definition, the approximate endpoint of an NRI target trajectory is known. The problem of tracking a maneuvering NRI target is particularly difficult for two reasons. First, there will be a lack of knowledge on what kind of maneuver is taking place, causing a mismatch between the filter dynamic model and the maneuver. Second, as shown in Section 2.2.3, the commonly used C-EKF has difficulty tracking a target in the near field.

Several authors have added constraints to target tracking methods to obtain improved performance when prior information is available regarding the trajectory of the maneuvering target [106] [4]. In [106], a 9-state Singer model was used for tracking, with a linear constraint added as a pseudo-measurement to track a coordinated turn maneuver. Similarly, in [4], a converted measurement filter was used to track a coordinated turn maneuver with a nonlinear constraint that required that $\boldsymbol{e}_v \bullet \boldsymbol{a} = 0$, where $\boldsymbol{e}_v$ is a unit vector in the direction of the velocity and $\boldsymbol{a}$ is the linear acceleration vector. The constraint was added as a pseudo-measurement in an additional set of update steps executed after the filter update steps.

Kalman filtering algorithms with constraints have been applied in several areas including fault detection [100], process control [108] [60], tracking along road networks [115], head pose tracking [56], aircraft flight plans [125], and others. A comprehensive survey of many methods for applying constraints for tracking along with a long list of citations can be found in [101].

A problem that is mathematically similar to tracking a maneuvering NRI target with an endpoint constraint is that of a homing missile tracking its target. The interceptor missile uses a coordinate system centered at its center of gravity, and its trajectory is

guided along a path with an endpoint at its target. One guidance law used frequently to provide guidance to interceptor missiles is that of proportional navigation (PN). Both two-dimensional [88] and three-dimensional [116] PN models have been developed for spherical coordinate systems. It will be shown that the homing missile acceleration kinematic algorithms required to maintain the target as its trajectory endpoint can be used in the kinematic model for tracking an EPC maneuvering an NRI target.

2.4.2. EPC Development

The EPC EKF leverages the PN principles developed in [116] to form its dynamic equations. To apply the EPC condition to the Spherical EKF developed in the previous section, we need only refer to the equation for the total acceleration in (2.89) and modify it using the conditions set forth in [116]. Since the sensor is stationary, applying the PN accelerations will force the predicted target state back toward the direct radial path to the sensor origin.

To add the endpoint constraint, one must begin with the observation that a vector pointing to a general location in a spherical coordinate system can be described as:

$$\boldsymbol{\rho} = R\hat{\boldsymbol{r}}. \tag{2.122}$$

It follows immediately that

$$\dot{\boldsymbol{\rho}} = \dot{R}\hat{\boldsymbol{r}} + R\dot{\theta}\cos\varphi\hat{\boldsymbol{\theta}} + R\dot{\varphi}\hat{\boldsymbol{\varphi}} \tag{2.123}$$

represents the velocity of the object at that location.

Let $\boldsymbol{\omega}$ be the angular velocity vector of a rotating coordinate system. We can leverage the knowledge of the time derivatives of the spherical unit vectors [43] and their relationship to the angular velocity vector to obtain

$$\dot{\hat{\boldsymbol{r}}} = \boldsymbol{\omega} \times \hat{\boldsymbol{r}} = \dot{\theta}\cos\varphi\hat{\boldsymbol{\theta}} + \dot{\varphi}\hat{\boldsymbol{\varphi}}, \tag{2.124}$$

$$\dot{\hat{\boldsymbol{\theta}}} = \boldsymbol{\omega} \times \hat{\boldsymbol{\theta}} = -\dot{\theta}\cos\varphi\hat{\boldsymbol{r}} + \dot{\theta}\sin\varphi\hat{\boldsymbol{\varphi}}, \tag{2.125}$$

and

$$\dot{\hat{\boldsymbol{\varphi}}} = \boldsymbol{\omega} \times \hat{\boldsymbol{\varphi}} = -\dot{\varphi}\hat{\boldsymbol{r}} - \dot{\theta}\sin\varphi\hat{\boldsymbol{\theta}}. \tag{2.126}$$

Taking the results for equations (2.124)–(2.126), one can solve for $\boldsymbol{\omega}$ to show

$$\boldsymbol{\omega} = \dot{\theta}\sin\varphi\hat{\boldsymbol{r}} - \dot{\varphi}\hat{\boldsymbol{\theta}} + \dot{\theta}\cos\varphi\hat{\boldsymbol{\varphi}}, \tag{2.127}$$

as given in [116].

It is now possible to add a constraint that maintains the endpoint of the trajectory at the origin by taking the cross product of the linear and angular velocities:

$$\boldsymbol{a}_{\mathrm{c}} = k\dot{\boldsymbol{\rho}} \times \boldsymbol{\omega}. \tag{2.128}$$

This produces an acceleration vector identical to the 3D proportional navigation acceleration found in [116],

$$\begin{aligned}\boldsymbol{a}_{\mathrm{c}} = {} & kr\left(\dot{\theta}^2\cos^2\varphi + \dot{\varphi}^2\right)\hat{\boldsymbol{r}} - k\left(\dot{\theta}\dot{r}\cos\varphi - r\dot{\varphi}\sin\varphi\right)\hat{\boldsymbol{\theta}} - \\ & k\left(\dot{\varphi}\dot{r} + r\frac{\dot{\theta}^2}{2}\sin 2\varphi\right)\hat{\boldsymbol{\varphi}},\end{aligned} \tag{2.129}$$

where k is the PN constant. The total acceleration can now be obtained for the end-point constrained kinematics by combining Equations (2.129) and (2.89). The resulting total acceleration is

$$\begin{aligned}\boldsymbol{a} = \boldsymbol{a}_{\mathrm{p}} + \boldsymbol{a}_{\mathrm{c}} = {} & \left[\ddot{r} + (k-1)\dot{\varphi}^2 r + (k-1)r\dot{\theta}^2\cos^2(\varphi)\right]\hat{\boldsymbol{r}} + \\ & \left[r\ddot{\theta}\cos(\varphi) + (2-k)\dot{r}\dot{\theta}\cos(\varphi) + (k-2)\dot{\varphi}\dot{\theta}r\sin(\varphi)\right]\hat{\boldsymbol{\theta}} + \\ & \left[\ddot{\varphi}r + (1-k)r\dot{\theta}^2\cos(\varphi)\sin(\varphi) + (2-k)\dot{\varphi}\dot{r}\right]\hat{\boldsymbol{\varphi}}.\end{aligned} \tag{2.130}$$

By substituting this equation in for (2.89), one can follow the same steps to obtain the dynamic transition equations with the new constraint; thus, the equations

$$\ddot{r} = (1-k)\dot{\varphi}^2 r + (1-k) r \dot{\theta}^2 \cos^2(\varphi), \tag{2.131}$$

$$\ddot{\theta} = \frac{(k-2)\dot{r}\dot{\theta}}{r} + (k-2)\dot{\varphi}\dot{\theta}\tan(\varphi), \tag{2.132}$$

and

$$\ddot{\varphi} = (k-1)\dot{\theta}^2 \cos(\varphi)\sin(\varphi) + \frac{(k-2)\dot{\varphi}\dot{r}}{r} \tag{2.133}$$

can be solved using the finite-difference techniques described previously. Thus, Equations (2.96)–(2.101) become

$$r_{\mathrm{n}} = r_{\mathrm{n-1}} + T\dot{r}_{\mathrm{n-1}} + (1-k)\frac{T^2}{2} r_{\mathrm{n-1}}\left[\dot{\theta}^2_{\mathrm{n-1}} \cos^2 \varphi_{\mathrm{n-1}} + \dot{\varphi}^2_{\mathrm{n-1}}\right], \tag{2.134}$$

$$\dot{r}_{\mathrm{n}} = \dot{r}_{\mathrm{n-1}} + (1-k) T r_{\mathrm{n-1}}\left[\dot{\theta}^2_{\mathrm{n-1}} \cos^2 \varphi_{\mathrm{n-1}} + \dot{\varphi}^2_{\mathrm{n-1}}\right], \tag{2.135}$$

$$\theta_{\mathrm{n}} = \theta_{\mathrm{n-1}} + T\dot{\theta}_{\mathrm{n-1}} + (k-2)\frac{T^2}{2}\dot{\theta}_{\mathrm{n-1}}\left[\dot{\varphi}_{\mathrm{n-1}} \tan\varphi_{\mathrm{n-1}} - \frac{\dot{r}_{\mathrm{n-1}}}{r_{\mathrm{n-1}}}\right], \tag{2.136}$$

$$\dot{\theta}_{\mathrm{n}} = \dot{\theta}_{\mathrm{n-1}} + (k-2) T \dot{\theta}_{\mathrm{n-1}}\left[\dot{\varphi}_{\mathrm{n-1}} \tan\varphi_{\mathrm{n-1}} - \frac{\dot{r}_{\mathrm{n-1}}}{r_{\mathrm{n-1}}}\right], \tag{2.137}$$

$$\varphi_{\mathrm{n}} = \varphi_{\mathrm{n-1}} + T\dot{\varphi}_{\mathrm{n-1}} - \frac{T^2}{2}\left[(2-k)\frac{\dot{r}_{\mathrm{n-1}}}{r_{\mathrm{n-1}}}\dot{\varphi}_{\mathrm{n-1}} + (1-k)\dot{\theta}^2_{\mathrm{n-1}} \sin\varphi_{\mathrm{n-1}} \cos\varphi_{\mathrm{n-1}}\right], \tag{2.138}$$

$$\dot{\varphi}_{\mathrm{n}} = \dot{\varphi}_{\mathrm{n-1}} - T\left[(2-k)\frac{\dot{r}_{\mathrm{n-1}}}{r_{\mathrm{n-1}}}\dot{\varphi}_{\mathrm{n-1}} + (1-k)\dot{\theta}^2_{\mathrm{n-1}} \sin\varphi_{\mathrm{n-1}} \cos\varphi_{\mathrm{n-1}}\right]. \tag{2.139}$$

The dynamic state equations yield another matrix equation similar to (2.102),

$$\boldsymbol{x}_{\mathrm{n}} = \mathbf{f}_{\mathrm{EPC}}(\boldsymbol{x}_{\mathrm{n-1}}) = \boldsymbol{F}\boldsymbol{x}_{\mathrm{n-1}} + \boldsymbol{G}\mathbf{u}_{\mathrm{EPC}}(\boldsymbol{x}_{\mathrm{n-1}}) + \boldsymbol{v}_{\mathrm{n-1}}. \tag{2.140}$$

The difference between (2.140) and (2.102) is that the nonlinear terms are altered by applying EPC. The vector containing the nonlinear components of the state-update equation is

$$\mathbf{u}_{\mathrm{EPC}}(\boldsymbol{x}) = \begin{bmatrix} (1-k)r\left(\dot{\theta}^2\cos^2\varphi + \dot{\varphi}^2\right) \\ (2-k)\dot{\theta}\left(\dot{\varphi}\tan\varphi - \dfrac{\dot{r}}{r}\right) \\ (k-2)\dfrac{\dot{r}\dot{\varphi}}{r} + (k-1)\dot{\theta}^2\sin\varphi\cos\varphi \end{bmatrix}. \tag{2.141}$$

Now that the dynamic equations have been constructed for the EPC filter, all that remains alter the appropriate Kalman filter components that are affected by the introduction of $\mathbf{u}_{EPC}(\boldsymbol{x})$. The Jacobian of the vector transition equation $\hat{\boldsymbol{F}}$ from (2.110) becomes

$$\hat{\mathbf{F}}_{\mathrm{EPC}}(\boldsymbol{x}) = \boldsymbol{F} + \boldsymbol{G}\hat{\mathbf{U}}_{\mathrm{EPC}}(\boldsymbol{x}), \tag{2.142}$$

where

$$\hat{\mathbf{U}}_{\mathrm{EPC}}\left(\hat{\boldsymbol{x}}_{n|n-1}\right) = \left[\nabla_{\boldsymbol{x}}\mathbf{u}_{\mathrm{EPC}}{}^{\mathrm{T}}(\boldsymbol{x})\right]^{\mathrm{T}}_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{n|n-1}}. \tag{2.143}$$

All elements of the Jacobian have previously been defined, except for $\hat{\boldsymbol{U}}_{EPC}$, which must be altered to account for the PN coefficients, as shown in (2.144).

$$\hat{\boldsymbol{U}}_{EPC}(\boldsymbol{\rho}) = \begin{bmatrix} (1-k)\left(\dot{\theta}^2\cos^2\varphi + \dot{\varphi}^2\right) & \dfrac{(2-k)\dot{r}\dot{\theta}}{r^2} & \dfrac{(2-k)\dot{r}\dot{\varphi}}{r^2} \\ 0 & \dfrac{(k-2)\dot{\theta}}{r} & \dfrac{(k-2)\dot{\varphi}}{r} \\ 0 & 0 & 0 \\ 2(1-k)r\dot{\theta}\cos^2\varphi & (2-k)\left(\dot{\varphi}\tan\varphi - \dfrac{\dot{r}}{r}\right) & 2(k-1)\dot{\theta}\sin\varphi\cos\varphi \\ 2(k-1)r\dot{\theta}^2\sin\varphi\cos\varphi & (2-k)\dot{\theta}\dot{\varphi}\sec^2\varphi & (1-k)\dot{\theta}^2\left(\sin^2\varphi - \cos^2\varphi\right) \\ 2(1-k)r\dot{\varphi} & (2-k)\dot{\theta}\tan\varphi & \dfrac{(k-2)\dot{r}}{r} \end{bmatrix}^{\mathrm{T}}$$

(2.144)

2.4.3. EPC EKF Algorithm

With the exception of the state and covariance prediction equations, the remaining aspects of the EPC filter are identical to those of the S-EKF algorithm in Table 4. To complete the EPC filter, we simply substitute the endpoint condition components into (2.145) and (2.146) to complete Table 5.

Table 5: EPC EKF Algorithm

State Prediction	$\hat{\boldsymbol{x}}_{\mathrm{n}\mid\mathrm{n-1}} = \boldsymbol{F}\hat{\boldsymbol{x}}_{\mathrm{n-1}\mid\mathrm{n-1}} + \boldsymbol{G}\mathbf{u}_{\mathrm{EPC}}(\hat{\boldsymbol{x}}_{\mathrm{n-1}\mid\mathrm{n-1}})$	(2.145)
State Covariance	$\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n}\mid\mathrm{n-1}} = \hat{\boldsymbol{F}}_{\mathrm{EPC}}(\hat{\boldsymbol{x}}_{\mathrm{n-1}\mid\mathrm{n-1}})\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n-1}\mid\mathrm{n-1}}\left[\hat{\boldsymbol{F}}_{\mathrm{EPC}}(\hat{\boldsymbol{x}}_{\mathrm{n-1}\mid\mathrm{n-1}})\right]^{\mathrm{T}} + \boldsymbol{Q}(\hat{\boldsymbol{x}}_{\mathrm{n-1}\mid\mathrm{n-1}})$	(2.146)
Observation Prediction	$\hat{\boldsymbol{z}}_{\mathrm{n}\mid\mathrm{n-1}} = \boldsymbol{H}\hat{\boldsymbol{x}}_{\mathrm{n}\mid\mathrm{n-1}}$	(2.147)
Observation Covariance	$\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n}\mid\mathrm{n-1}} = \boldsymbol{H}\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n}\mid\mathrm{n-1}}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}$	(2.148)
State-Observation Covariance	$\boldsymbol{P}^{\boldsymbol{xz}}_{\mathrm{n}\mid\mathrm{n-1}} = \boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n}\mid\mathrm{n-1}}\boldsymbol{H}^{\mathrm{T}}$	(2.149)
Kalman Gain	$\boldsymbol{K}_{\mathrm{n}} = \boldsymbol{P}^{\boldsymbol{xz}}_{\mathrm{n}\mid\mathrm{n-1}}\left[\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n}\mid\mathrm{n-1}}\right]^{-1}$	(2.150)
State Update	$\hat{\boldsymbol{x}}_{\mathrm{n}\mid\mathrm{n}} = \hat{\boldsymbol{x}}_{\mathrm{n}\mid\mathrm{n-1}} + \boldsymbol{K}_{\mathrm{n}}(\boldsymbol{z}_{\mathrm{n}} - \hat{\boldsymbol{z}}_{\mathrm{n}\mid\mathrm{n-1}})$	(2.151)
State Covariance	$\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n}\mid\mathrm{n}} = \boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n}\mid\mathrm{n-1}} - \boldsymbol{K}_{\mathrm{n}}\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n}\mid\mathrm{n-1}}\boldsymbol{K}_{\mathrm{n}}^{\mathrm{T}}$	(2.152)

2.5.EPC EKF Proportional Navigation Constant Selection

2.5.1. Purpose of Study

In the original EPC filter publication [45], the PN constant *k* was set to a value of 10 based on a series of brief tests using NRI target trajectories. The purpose of this study was to demonstrate the methodology used to select this value. The preferred PN constant delivers the best bearing and elevation error performance without sacrificing the performance in the range, range rate, bearing rate, or elevation rate. Although these four state variables are not the primary motivators for the development of the EPC filter, failure to account for them could result in the development of a filter with no practical value.

In this study, a variety of PN constant values ranging from k=1 to k=50 were evaluated against the Dutch roll trajectory. Preliminary studies using the radially inbound trajectory found that the PN constant could be set arbitrarily high to achieve slightly better results. However, the radially inbound trajectory nearly perfectly matched to the kinematic model used in the EPC filter. Any NRI target that does not follow a perfectly radial path would cause the filter track to diverge if the PN constant was set too high. Because a priori knowledge of the target maneuvers is not available, it was determined that a weaving NRI target such as the Dutch roll target is preferable for filter tuning.

2.5.2. Findings for PN Constant Tuning

Owing to the large number of filters evaluated in this study, only the overall box diagrams for each PN constant and the overall result table are shown. A PN constant of 10 was found to be optimal for the Dutch roll target. Multiple values near k=10 were also

evaluated; however, all were found to offer nearly identical results with slightly worse performance in bearing and elevation. The position error statistics are shown in Figure 20 and the velocity error statistics are shown in Figure 21. These statistics are captured as box plots (see Figure 11 for a labeled example plot).

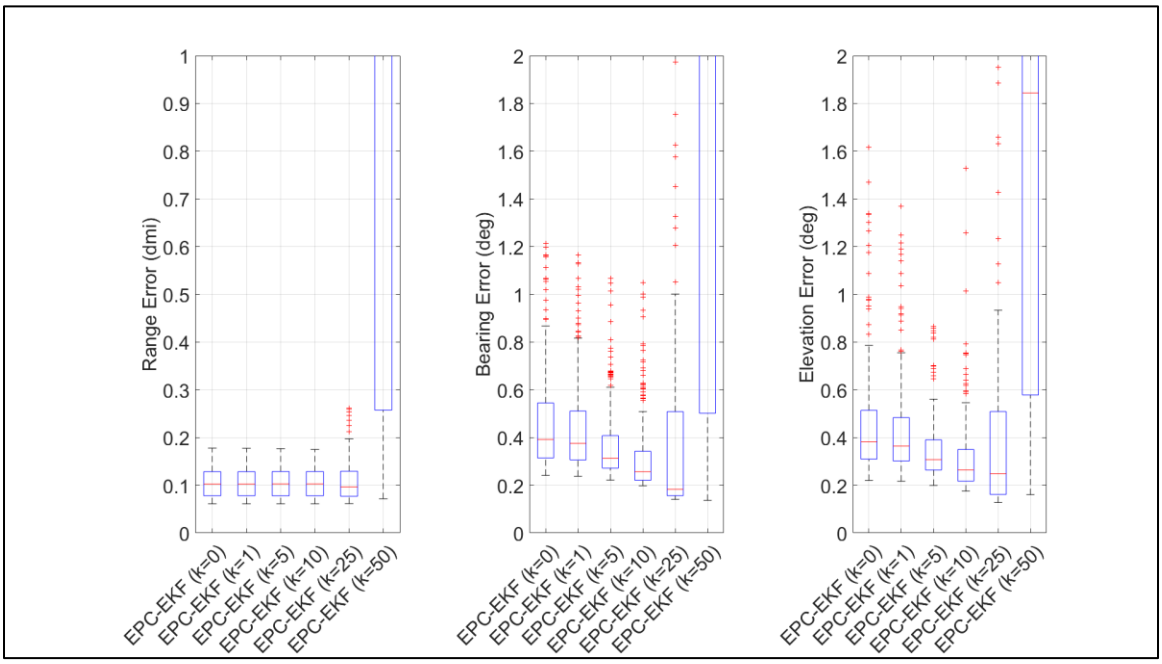

Figure 20: Position error box plot for PN constant tuning study

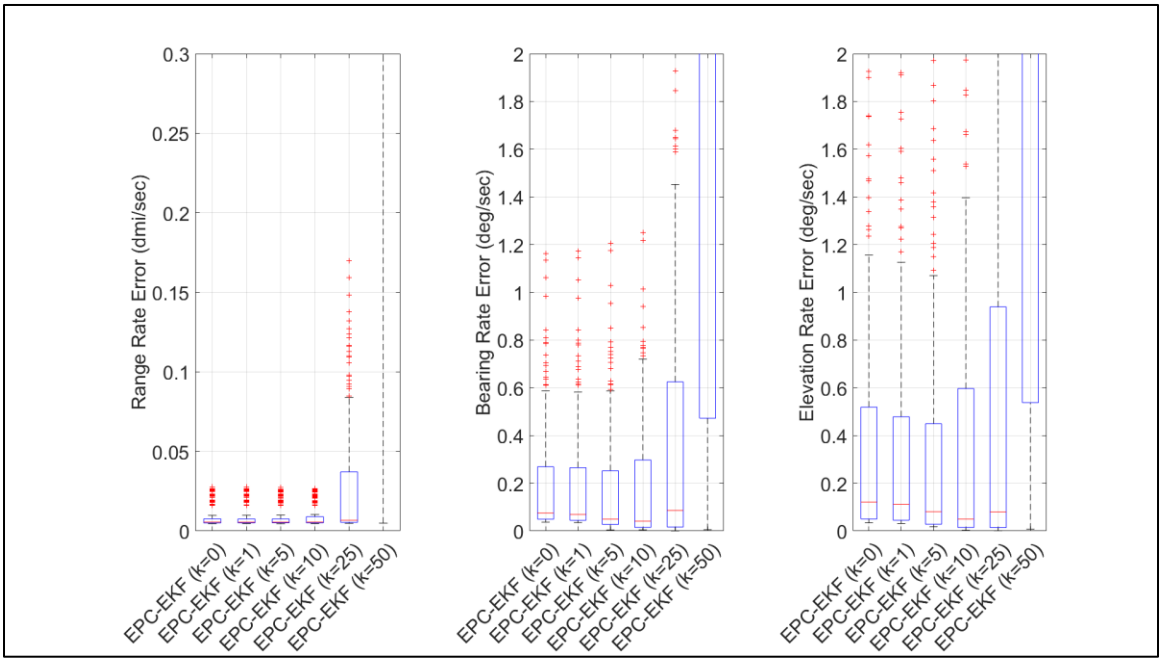

Figure 21: Velocity error box plot for PN constant tuning study

EPC filters featuring high PN constant values (50 and above) resulted in diverging tracks early on in the target trajectory. The resultant errors cannot be contained within the box plots in Figure 20 and Figure 21. A PN constant of 25 features a smaller median error value for the bearing and elevation, but there is a greater spread of errors, including significantly more outliers across all positions and velocity states. Lower PN constant values, such as k=1 and k=5, have a comparable number of outliers to k=10, but their median error values are notably higher, and the spread of all errors is greater. Based on these results, it was confirmed that the PN constant (k=10) offers the best balance of errors and should be used for the EPC EKF.

The overall RMS error performance was evaluated using all errors for all track state updates in the scenario. For the PN constant values whose tracks did not diverge, it is clear that a value of k=10 provides superior performance for range, bearing, and elevation, while suffering only marginal performance losses in the velocity components of these states. The results are shown in Table 6, with the best-performing filter for each dimension highlighted in green.

Table 6: PN constant selection study – overall RMSE table

Filter Name	RMS Error Overall					
	Range (dmi)	Range Rate (dmi/s)	Bearing (deg)	Bearing Rate (deg/s)	Elevation (deg)	Elevation Rate (deg/s)
EPC-EKF (k=0)	0.1058	0.0948	0.4646	0.1947	0.4548	0.3660
EPC-EKF (k=1)	0.1057	0.0947	0.4451	0.1884	0.4312	0.3574
EPC-EKF (k=5)	0.1056	0.0946	0.3772	0.1712	0.3495	0.3296
EPC-EKF (k=10)	0.1053	0.0952	0.3260	0.1848	0.3108	0.3487
EPC-EKF (k=25)	0.1103	0.1665	0.7268	0.6271	0.7955	0.7794

2.6.EPC EKF and Legacy Filter Performance Study

2.6.1. EPC EKF Comparison Study Goals

The primary goal of developing the EPC EKF was to show that the modified kinematic model at the core of the filter offers superior tracking performance compared to the C-EKF and S-EKF. In this case, superior tracking performance refers to reduced track-state errors, particularly in the bearing and elevation dimensions. NRI targets tend to exhibit maneuvers that cause the system track to propagate in the wrong direction between the radar measurement updates. This results in significant errors perpendicular to the radial path of the target. If it can be shown that the EPC EKF has lower angular errors while offering a tolerable range, range rate, bearing rate, and elevation rate errors, it can be said that the EPC EKF experiment is a success.

The performance results against the maneuvering targets introduced in Section 1.3 are covered in the following sections. For the two NRI targets (radially inbound and Dutch roll), the performance will be discussed with respect to the performance gains offered by the EPC EKF. For non-NRI targets, observations shall be shared, which motivated further study into EPC filter improvements.

2.6.2. Radially Inbound Target

The radially inbound trajectory is the scenario that is most accurately modeled using the EPC kinematic model. The trajectory does not deviate from the radial path; therefore, it is expected that the EPC EKF will outperform the C-EKF and S-EKF for this case. It is not surprising that the bearing (Figure 22) and elevation (Figure 23) RMSE performance of the EPC EKF is significantly better than that of the legacy filters. The

errors at each measurement update are lower than those of the legacy filter, and the propagated states do not deviate from the target truth as much as the C-EKF and S-EKF track states do. This occurs because legacy filters allow their tracks to propagate away from the radial path more easily than the EPC EKF.

The box plots featured in Figure 24 and Figure 25 add further evidence that the EPC EKF is the superior filter for this test case. The EPC EKF suffers no noticeable losses in range or range rate RMSE performance, while significantly outperforming the legacy filters in angles and angle rates, with perhaps one or two outlier errors falling beyond the smallest outlier for the S-EKF. With the radially inbound scenario as a baseline, it is time to evaluate a more challenging NRI target.

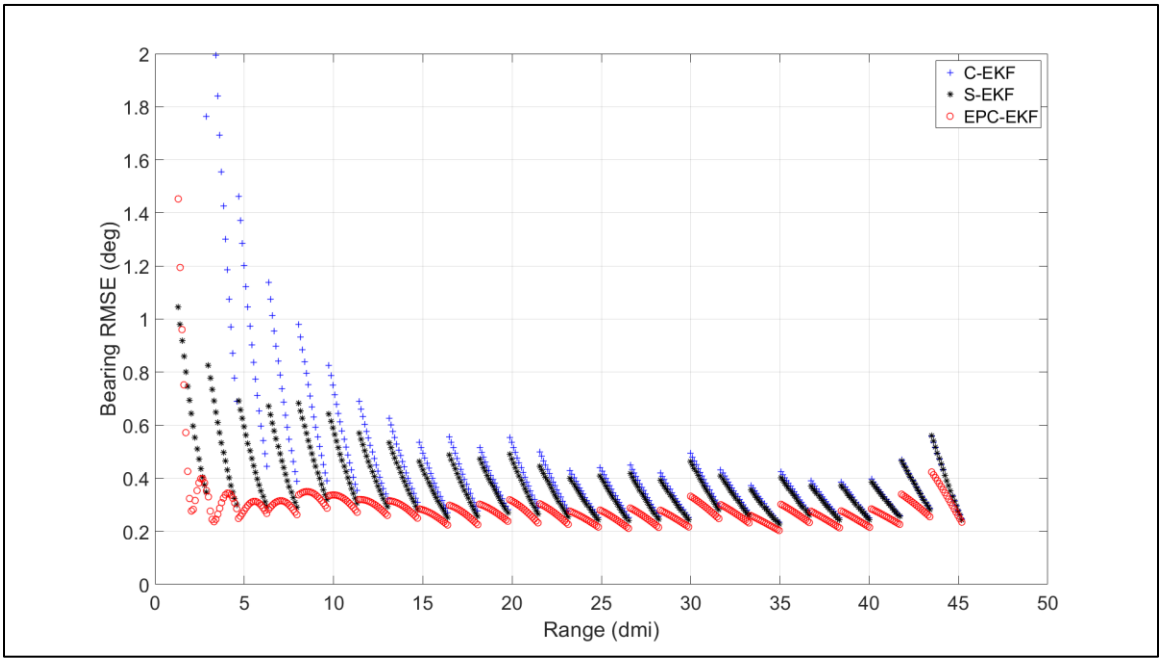

Figure 22: EPC EKF vs legacy filters – bearing RMSE for radially inbound scenario

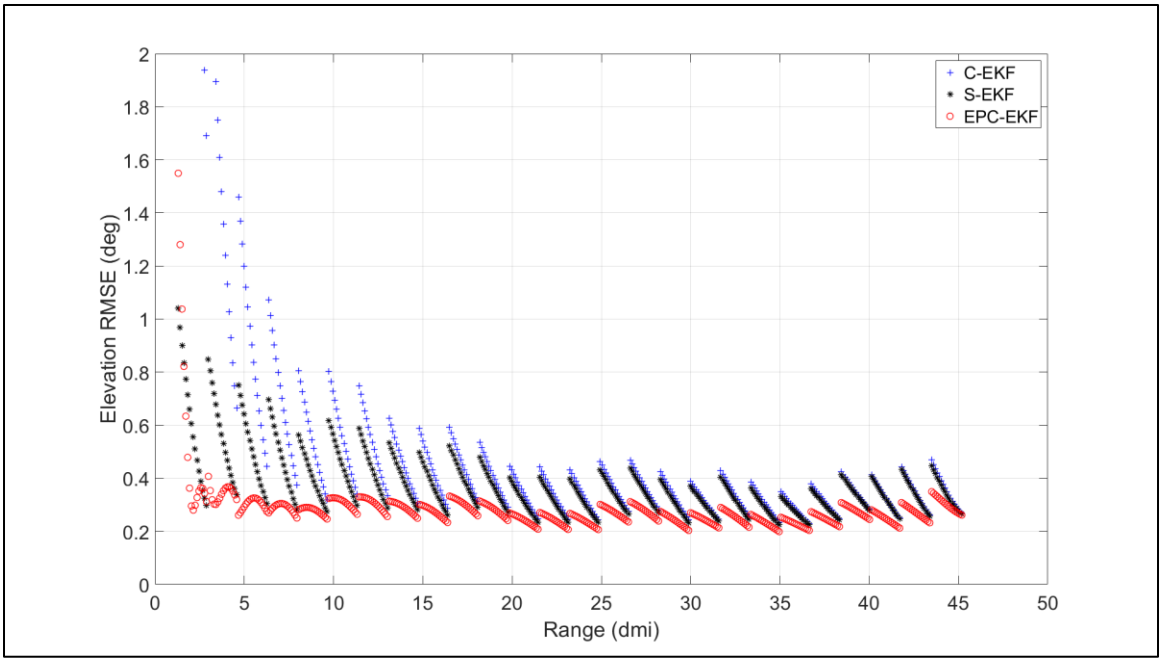

Figure 23: EPC EKF vs legacy filters – elevation RMSE for radially inbound scenario

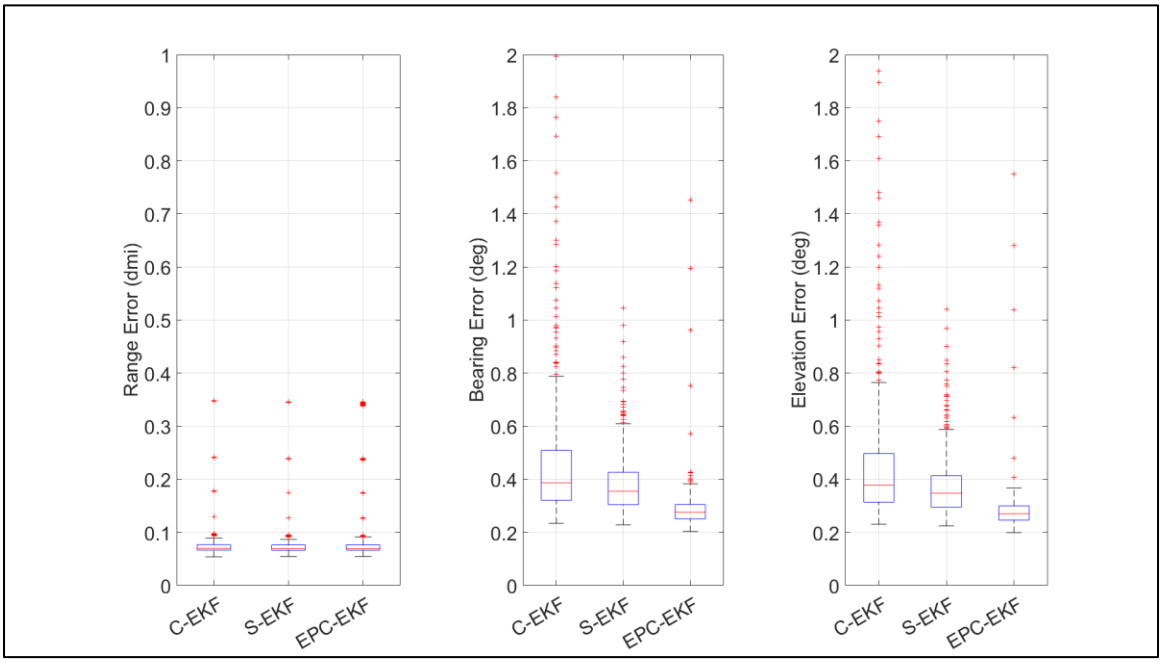

Figure 24: EPC EKF vs legacy filters – position error box plots for radially inbound scenario

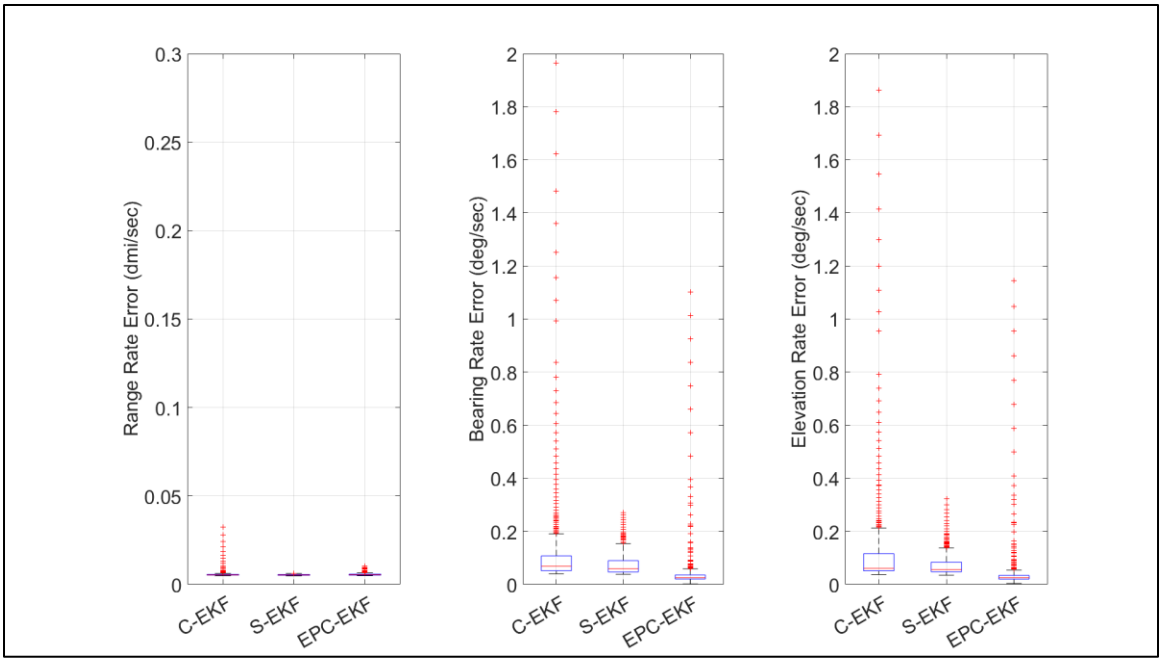

Figure 25: EPC EKF vs legacy filters – velocity error box plots for radially inbound scenario

2.6.3. Dutch Roll Target

For the Dutch roll scenario, the target approaches the sensor origin from a range of 25 dmi, and begins performing its weaving maneuver at approximately 12.5 miles range to go (RTG). During the radial inbound phase of the trajectory, the error performance closely resembled that of the radially inbound target. After the weave begins, the errors for all filters increase, with the EPC EKF featuring the lowest error growth and maintaining its advantage over the legacy filters throughout the maneuver period. The angle RMSE performance of the EPC EKF is shown in Figure 26 and Figure 27. The position and velocity box plots in Figure 28 and Figure 29 tell a similar story regarding the overall angle error statistics. The EPC EKF has by far the best overall bearing and elevation error performance, while remaining competitive with the S-EKF in terms of range, range rate, and angular velocities.

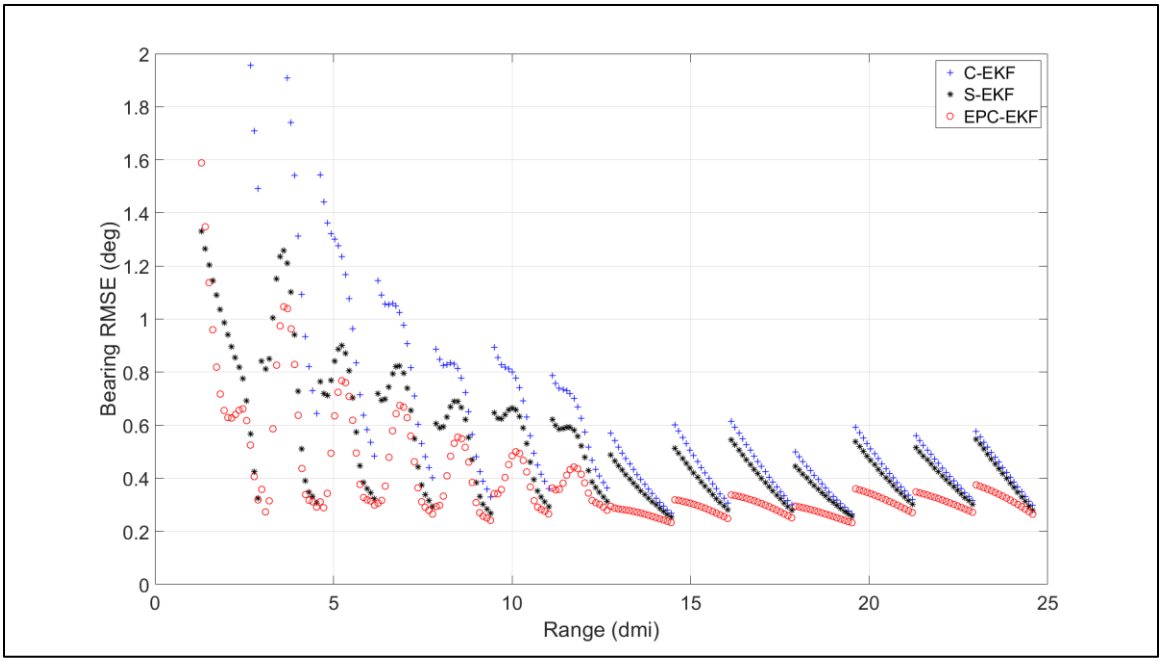

Figure 26: EPC EKF vs legacy filters – bearing RMSE for Dutch roll scenario

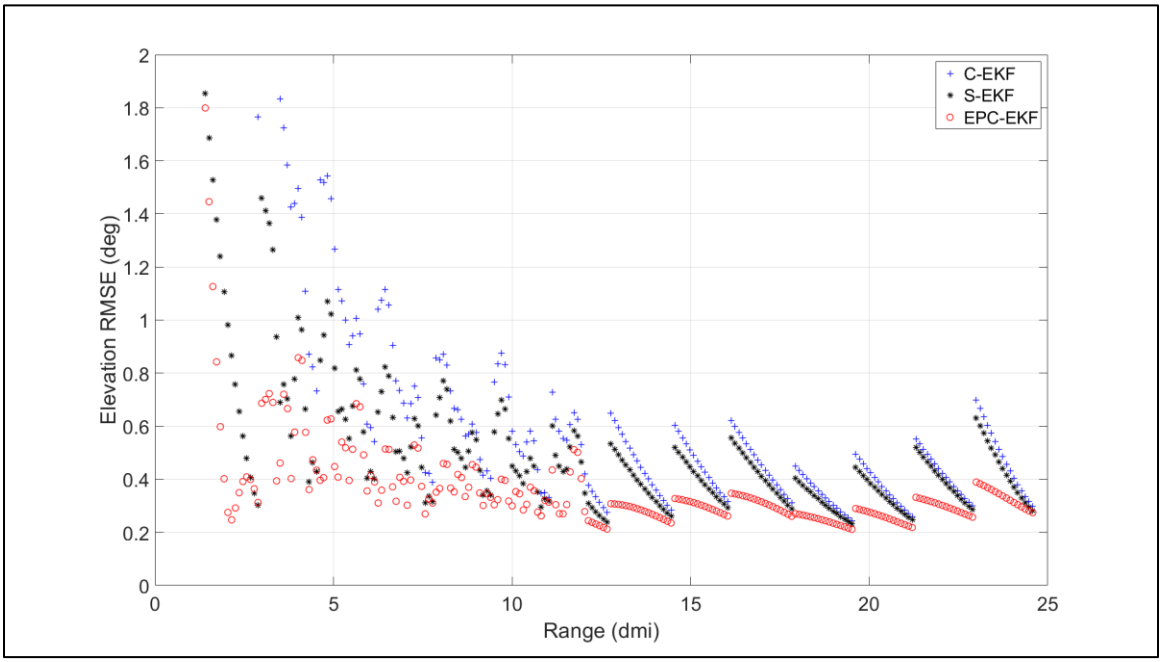

Figure 27: EPC EKF vs legacy filters – elevation RMSE for Dutch roll scenario

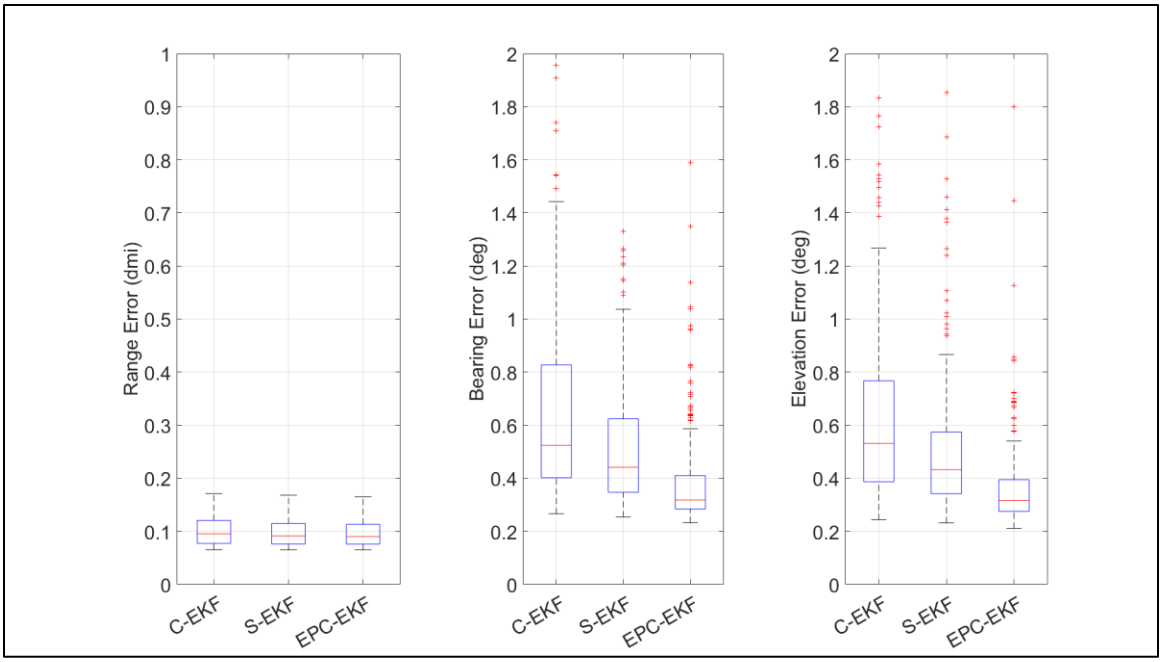

Figure 28: EPC EKF vs legacy filters – position error box plots for Dutch roll scenario

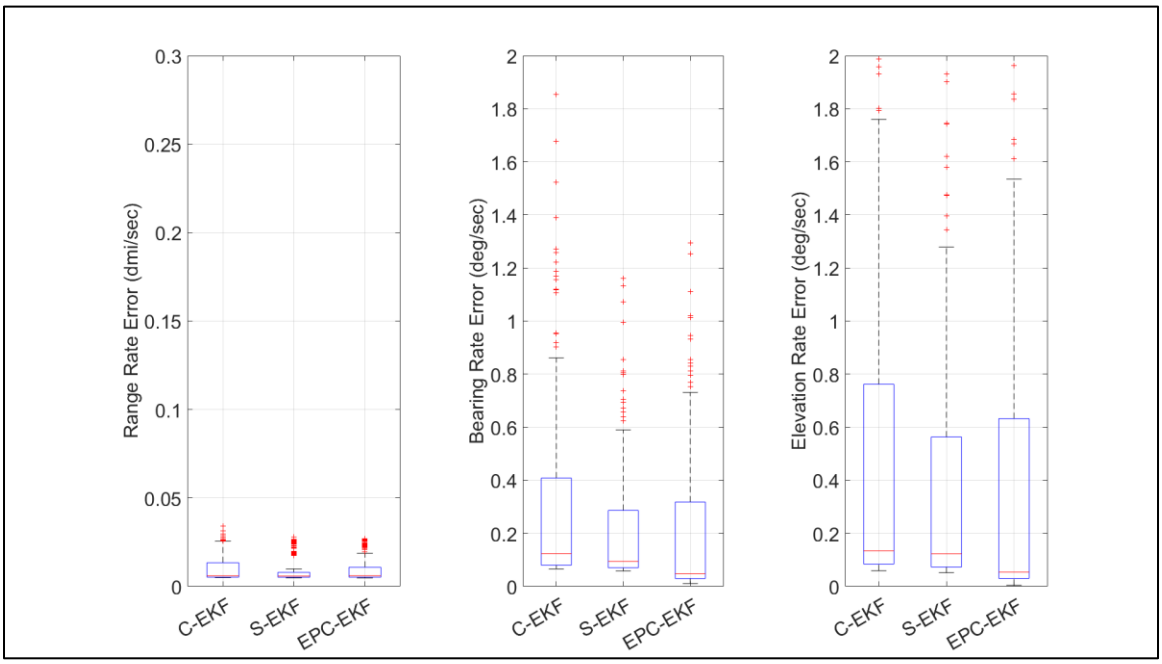

Figure 29: EPC EKF vs legacy filters – velocity error box plots for Dutch roll scenario

2.6.4. High-Diver Target

The high-diver's trajectory is not NRI for its full duration. The high-diver performs an abrupt turn in the vertical crossrange down toward the sensor origin at

approximately 10 dmi RTG. Prior to the downward turn, the high-diver target is not radially inbound because it would overshoot the sensor origin by 6 dmi if it maintained its initial trajectory. Consequently, the EPC filter failed to properly model the motion of the target. This failure is reflected in degradation of the elevation RMSE (Figure 31), starting at approximately 20 dmi RTG. Notably, the bearing RMSE (Figure 30) is not negatively affected by the mis-modeling of the target motion.

The box plots in Figure 32 and Figure 33 further emphasize the performance gains in bearing and performance losses in elevation for the EPC EKF. These gains and losses are carried over to the bearing rate and elevation rate errors, respectively. The range and range rate errors are in family with the S-EKF and C-EKF. Based on this evidence, the S-EKF is the preferred filter for the high-diver target until modifications to the EPC EKF can be made to address the performance degradation.

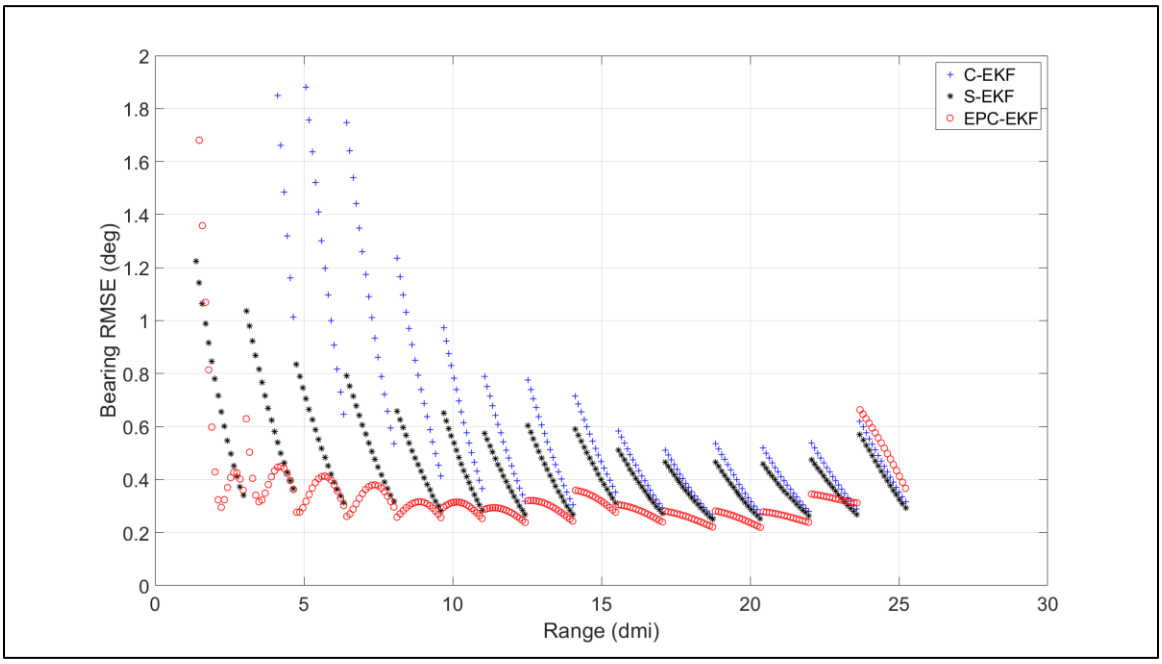

Figure 30: EPC EKF vs legacy filters – bearing RMSE for high-diver scenario

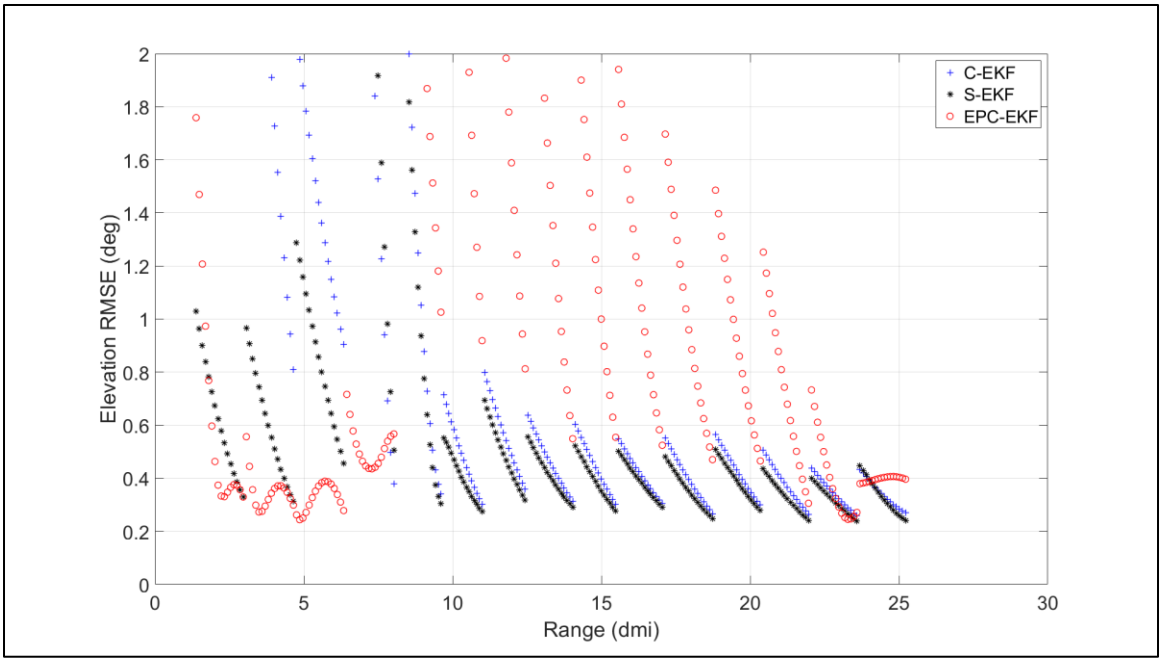

Figure 31: EPC EKF vs legacy filters – elevation RMSE for high-diver scenario

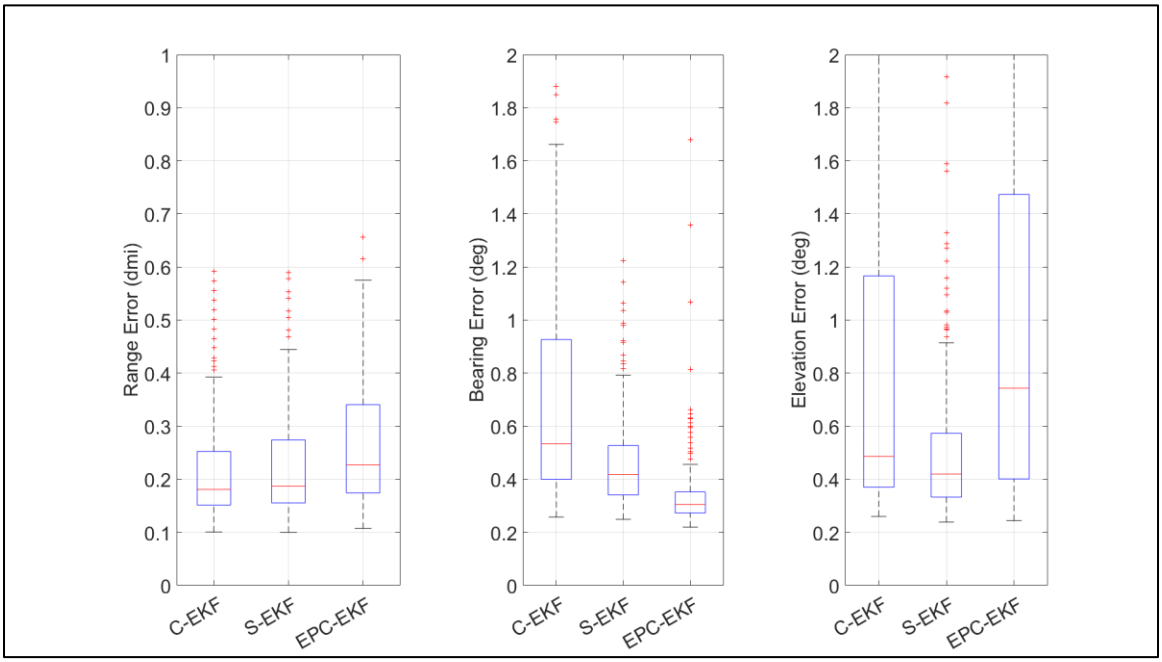

Figure 32: EPC EKF vs legacy filters – position error box plots for high-diver scenario

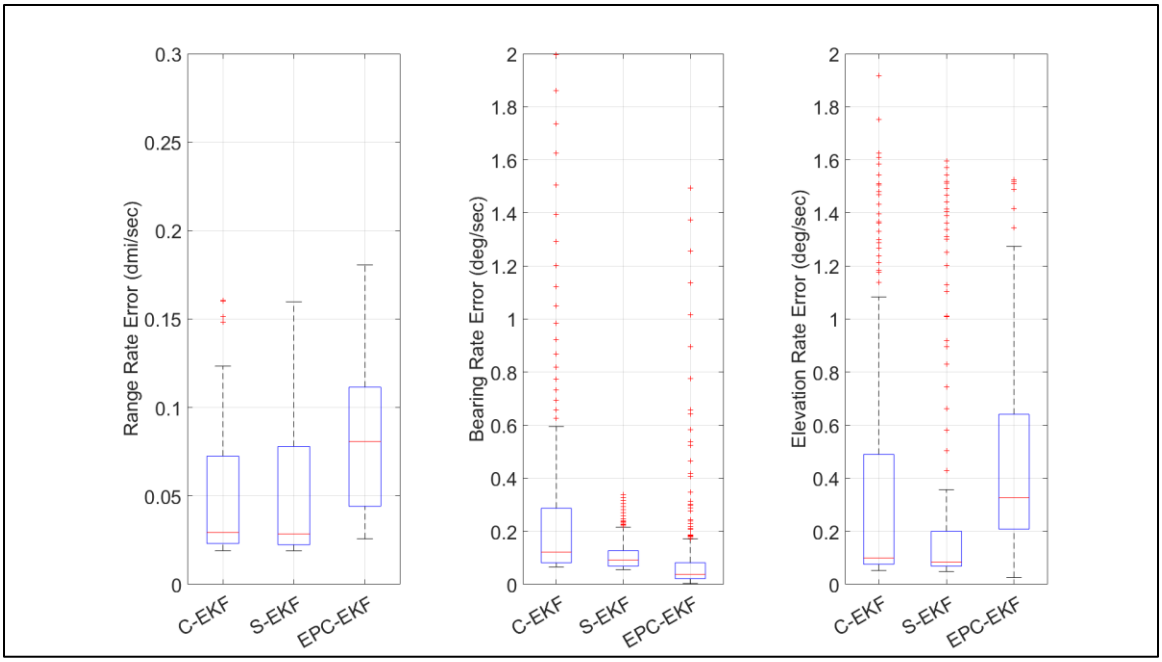

Figure 33: EPC EKF vs legacy filters – velocity error box plots for high-diver scenario

2.6.5. Dogleg Target

The dogleg trajectory is similar to the high-diver trajectory in that a single maneuver is performed that changes the target from a non-NRI trajectory into an NRI trajectory. The error performance essentially trades the error gains and losses in the bearing and elevation covered in Section 2.6.4. The dogleg scenario features a maneuver in horizontal crossrange relative to the sensor; therefore, the bearing RMSE performance of the EPC EKF is degraded until the turn occurs at approximately 6 dmi RTG as shown in Figure 34. The elevation RMSE performance does not suffer the same fate, and the EPC EKF outperforms the legacy filters, as shown in Figure 35. The EPC EKF is not able to fully recover from its diverging track behavior owing to the maneuver occurring late in the scenario.

The box plots (Figure 36-Figure 37) for the dogleg scenario further emphasize the loss of the bearing error performance and the gains in the elevation error performance.

For the bearing and bearing rate, the S-EKF is the superior filter, whereas for the elevation and elevation rate, the EPC EKF offers better error performance. The range and range rate errors are in family across all three filters.

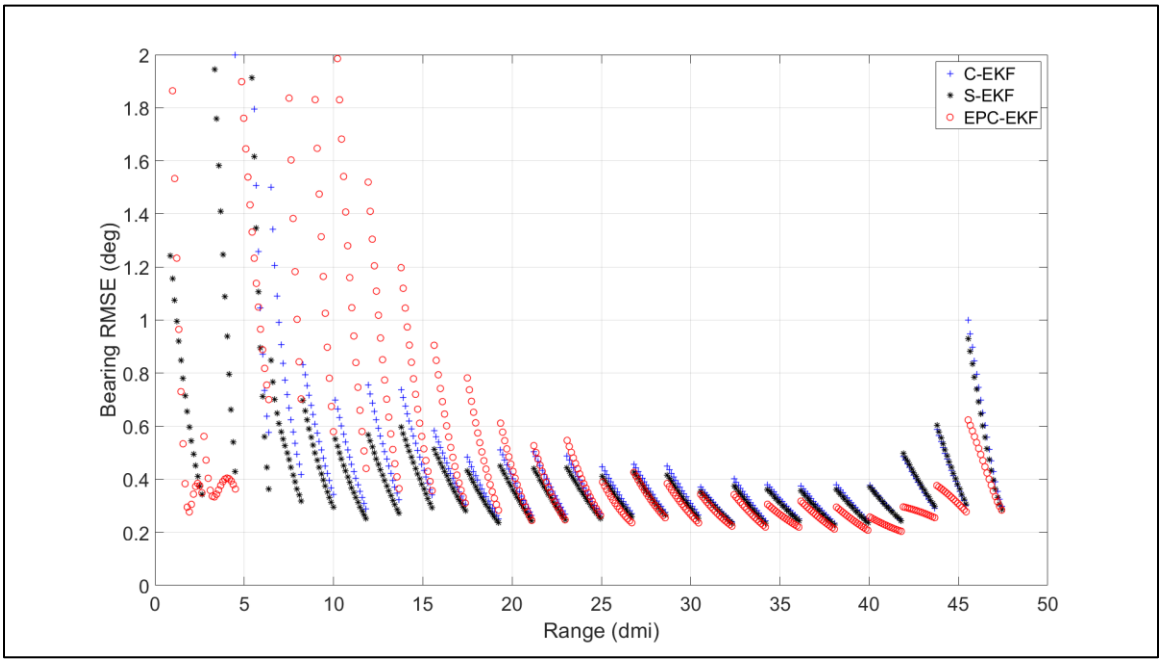

Figure 34: EPC EKF vs legacy filters – bearing RMSE for dogleg scenario

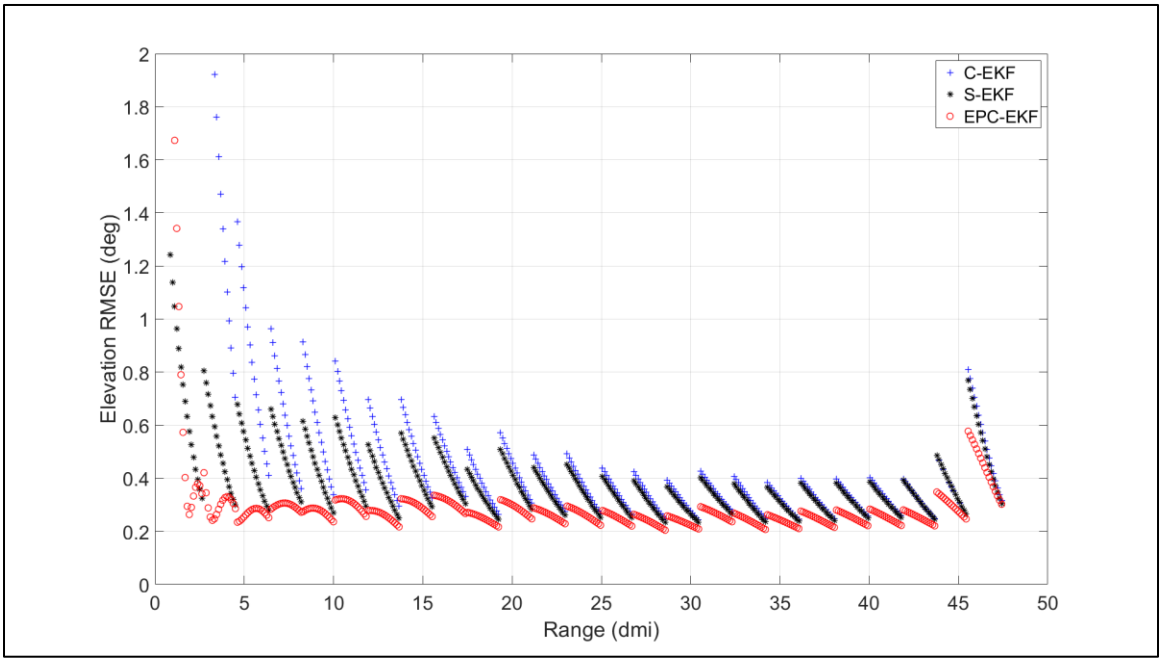

Figure 35: EPC EKF vs legacy filters – elevation RMSE for dogleg scenario

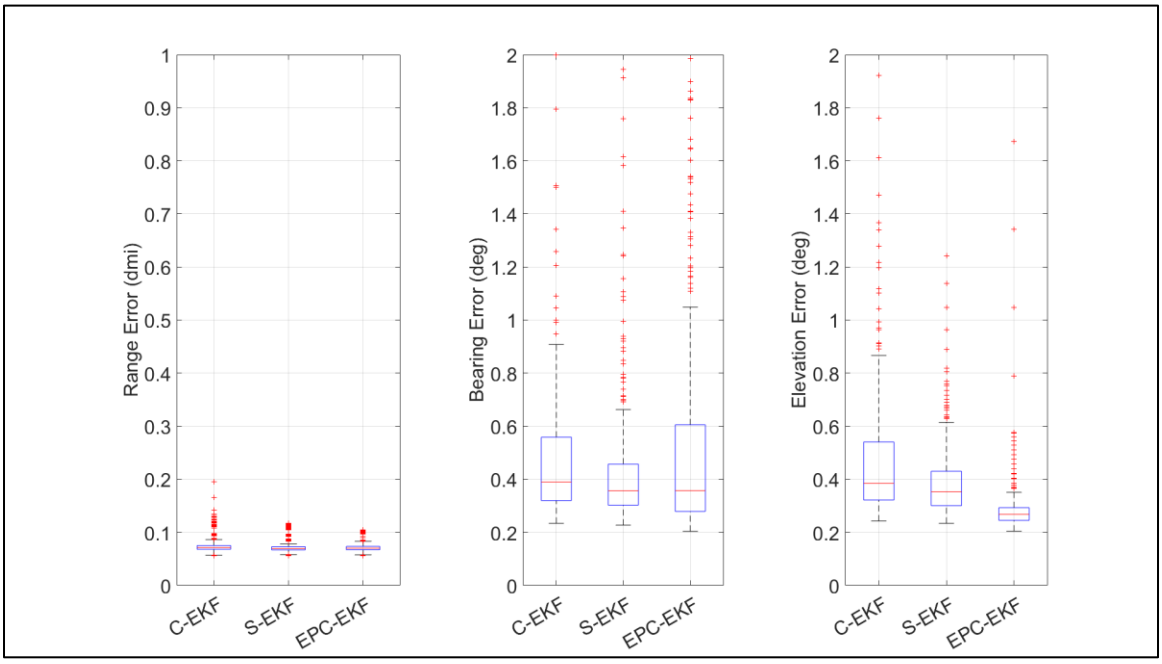

Figure 36: EPC EKF vs legacy filters – position error box plots for dogleg scenario

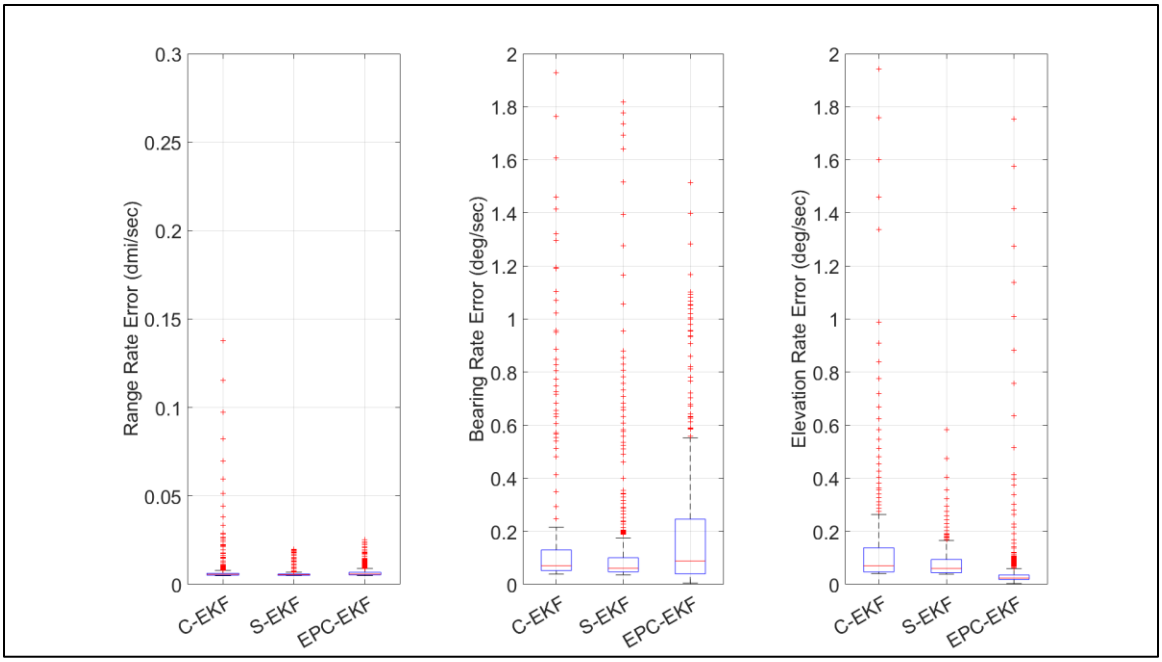

Figure 37: EPC EKF vs legacy filters – velocity error box plots for dogleg scenario

2.6.6. EPC vs Legacy Filter Overall Performance Assessment

The results of the EPC EKF study reaffirmed the initial hypothesis regarding EPC filter performance for NRI targets, while revealing that the EPC model is not suitable for all trajectory types. The EPC EKF delivers better error performance against NRI targets while suffering significant performance losses against non-NRI targets, where the EPC model fails to properly model the target maneuvers. In these non-NRI cases, the S-EKF delivered the best results among the three filters tested. This result offers further support for the findings in Section 2.2.3 as well as those of Haug and Williams [46]. The overall RMSE performance for each scenario is presented in Table 7. The best error performance for each dimension is highlighted in green for each trajectory.

Table 7: EPC EKF vs legacy filters – overall RMSE performance

Scenario	Filter Name	RMS Error Overall					
		Range (dmi)	Range Rate (dmi/s)	Bearing (deg)	Bearing Rate (deg/s)	Elevation (deg)	Elevation Rate (deg/s)
Radial Inbound	C-EKF	0.0943	0.0765	0.6422	0.1848	0.6186	0.1760
	S-EKF	0.0935	0.0737	0.3845	0.0744	0.3759	0.0738
	EPC-EKF	0.0934	0.0751	0.2875	0.0516	0.2829	0.0518
Dutch Roll	C-EKF	0.1009	0.1012	0.8950	0.3580	0.9486	0.5417
	S-EKF	0.0980	0.0961	0.5168	0.2130	0.5154	0.3851
	EPC-EKF	0.0969	0.0974	0.3931	0.2070	0.3741	0.3743
High-diver	C-EKF	0.3171	0.2457	1.2010	0.4091	1.1943	0.4726
	S-EKF	0.3224	0.2459	0.4589	0.1099	0.6674	0.2374
	EPC-EKF	0.3547	0.2900	0.3461	0.1064	1.1161	0.4378
Dogleg	C-EKF	0.0748	0.0920	1.0790	0.4317	0.7392	0.2438
	S-EKF	0.0720	0.0776	0.5016	0.1535	0.3881	0.0798
	EPC-EKF	0.0720	0.0843	0.5999	0.2077	0.2958	0.0645

Although the RMSE performance varied from scenario to scenario, the timing analysis did not. All filters were found to be real-time algorithms in this analysis, as listed

in Table 8. This means that the combat system and any downstream consumers of the track data are not hindered by slow computation times.

Table 8: EPC EKF vs legacy filters – runtime performance table

Scenario	Filter Name	Scenario Duration (sec)	Average Runtime (sec)	Runtime Ratio	Real Time?
Radial Inbound	C-EKF	117	0.00174	0.0015%	Yes
	S-EKF	117	0.00171	0.0015%	Yes
	EPC-EKF	117	0.00165	0.0014%	Yes
Dutch Roll	C-EKF	69	0.00124	0.0018%	Yes
	S-EKF	69	0.00088	0.0013%	Yes
	EPC-EKF	69	0.00084	0.0012%	Yes
High-diver	C-EKF	73	0.00115	0.0016%	Yes
	S-EKF	73	0.00109	0.0015%	Yes
	EPC-EKF	73	0.00124	0.0017%	Yes
Dogleg	C-EKF	108	0.00184	0.0017%	Yes
	S-EKF	108	0.00172	0.0016%	Yes
	EPC-EKF	108	0.00172	0.0016%	Yes

The EPC EKF was found to offer superior error performance across all state dimensions aside from the range rate for the radially inbound and Dutch roll scenarios. For the radially inbound target, the EPC EKF delivered overall angle RMSE performance improvements of 25.2% in bearing and 24.7% in elevation when compared to the S-EKF. Such large performance gains are largely owing to the use of a well-matched model for a perfectly radially inbound trajectory. The most encouraging result is that similar gains are observed in the Dutch roll scenario, with a 23.9% improvement in bearing RMSE and a 27.4% improvement in elevation RMSE. The S-EKF outperformed the C-EKF across all metrics and is the preferred filter for non-NRI scenarios based on this study. As an additional bonus, the EPC modification to the S-EKF adds a negligible amount of

computation time. However, not all targets are NRI and further development is required before the EPC EKF can be fielded in a combat system. In the next three chapters, we will apply the EPC model to a larger array of tracking algorithms to determine if they can improve upon the EPC EKF's NRI performance and, more importantly, mitigate the performance losses for non-NRI targets.

3. Alternative Discrete-Time EPC Filters

3.1.EPC Unscented Kalman Filter

3.1.1. UKF Background

Chapter 2 focused on the development of EKF algorithms to solve the NRI maneuvering target tracking problem. The filter dynamic and observation models were expanded in a Taylor polynomial about the state estimate to allow linearized predictions of the state and observation data. Although they are popular in the target tracking domain [71], EKFs are suboptimal filters and may not be the best algorithms for solving the NRI tracking problem. When the time between measurement updates is long, the EKF's linearized approximation degrades, which may result in performance losses or filter instability [62]. For problems in which the dynamic or observation transition functions are highly nonlinear, second-order terms can be included in the EKF at the expense of calculating Hessian matrices [43]. Another method for achieving second-order approximations in a nonlinear filter without using second order linearization methods is to use an unscented Kalman filter.

Unscented Kalman filters (UKFs) are a subclass of sigma point filters that can offer a nonlinear filter designer improved tracking performance compared with EKFs with the same dynamic models [61]. While EKFs use numerical differentiation to linearize the dynamics of the problem, UKFs use sigma point numerical integration to achieve a higher-order representation of the system [43][55][56][57]. As an added bonus, UKFs do not require the computation of complicated Jacobian or Hessian matrices, which further simplifies the filter development process. UKFs have been applied in

nonlinear control applications requiring full-state feedback, and have been shown to achieve higher accuracy than the EKF at a comparable level of complexity [109].

During the filter prediction step, the estimated state and covariance are used to generate a set of sigma points, which are used to approximate the Gaussian distribution that they describe. Figure 38 shows a set of sigma points (blue) for the EPC model that are then propagated forward in time for a target closing on the sensor origin. These sigma points are surrounded by a covariance ellipse, which describes the Gaussian distribution that they attempt to approximate.

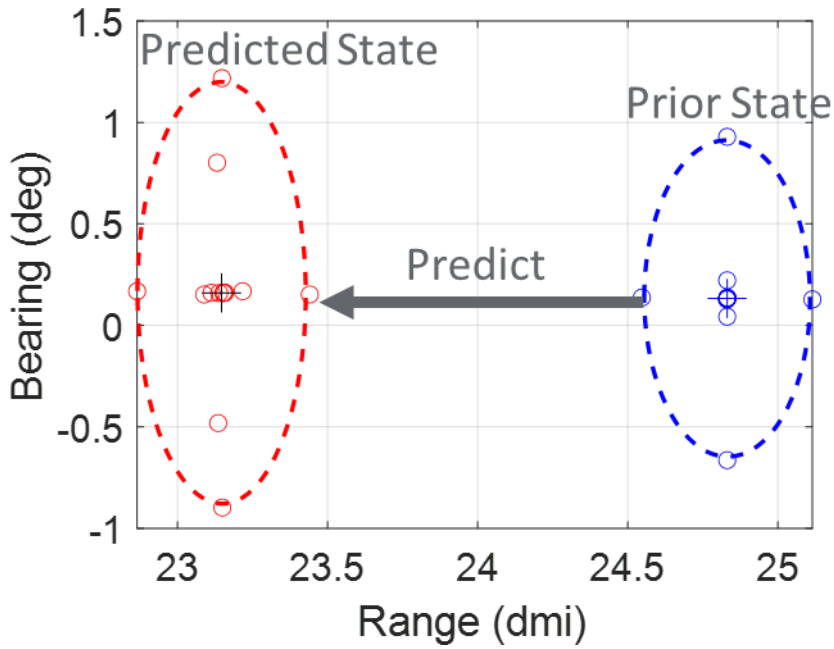

Figure 38: Sigma point prediction example in range and bearing

After the set of prior sigma points is propagated, process noise is added, and a new set of predicted sigma points is generated. The set of predicted sigma points are used to estimate the observation and state-observation covariance matrices. It is the generation, propagation, and maintenance of these sigma points that allow the UKF to achieve the

second-order approximation without linearization. The EPC UKF algorithm is discussed in detail in the next section.

3.1.2. EPC UKF Development

The EPC UKF requires the computation of the sigma points related to its estimated distribution. Kalman filter designers are most interested in the first two moments of this distribution, the mean and covariance, and the sigma points easily lend themselves to the calculation of these estimates. The affine transformation of the distribution when combined with sigma point methods makes it possible to compute the mean and covariance with relative ease [43].

The prior state update sigma points at time *n*-1 conditioned on all past observations ($\boldsymbol{\chi}^{(j)}_{n-1|n-1}$) are calculated using the affine transformation of the predicted state distribution and are found to be

$$\boldsymbol{\chi}^{(j)}_{n-1|n-1} = \hat{\boldsymbol{x}}_{n-1|n-1} + \sqrt{\boldsymbol{P}^{xx}_{n-1|n-1}}\boldsymbol{c}^{(j)}, \tag{3.1}$$

where $\boldsymbol{c}^{(j)}$ represents the vector points required to generate sigma points. The generation of sigma points is covered in detail in [43][56][57][61]. In the simplest form, this vector represents a set of orthogonal, 6x1 unit vectors, each extending one unit in the positive or negative direction for each dimension, then scaled by the sigma point weights. The vector $\boldsymbol{c}^{(j)}$ at the central point (j=0) is a 6 × 1 vector of zeros. Thus, for a six-dimensional filter, there are a total of thirteen sigma points.

The computation of state and covariance predictions at time *n* given all observations up to time *n*-1, becomes

$$\hat{\boldsymbol{x}}_{\mathrm{n|n\text{-}1}} = \sum_{\mathrm{j}=1}^{12} w_{\mathrm{j}} \mathbf{f}_{\mathrm{EPC}} \left(\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})} \right), \tag{3.2}$$

where $\mathbf{f}_{\mathrm{EPC}}$ is the same EPC dynamic transition function defined in (2.140) without including the noise vector $\boldsymbol{v}$. The covariance predictions are as follows:

$$\boldsymbol{c}^{(i)} = \sqrt{\frac{N_x}{1-w_0}} \boldsymbol{r}^{(i)} \tag{3.3}$$

$$\boldsymbol{P}_{\mathrm{n|n\text{-}1}}^{\boldsymbol{xx}} = \boldsymbol{Q} + \sum_{j=0}^{12} w_j \left[\mathbf{f}_{\mathrm{EPC}}(\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})}) - \hat{\boldsymbol{x}}_{n|n-1} \right] \left[\mathbf{f}_{\mathrm{EPC}}(\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})}) - \hat{\boldsymbol{x}}_{n|n-1} \right]^{\mathrm{T}}. \tag{3.4}$$

The sigma point weights depend on the user-defined free parameter w_0 and are given by

$$w_{\mathrm{j}} = \begin{cases} w_0, & j = 0 \\ (1-w_0)/12, & j = 1,2,...12. \end{cases} \tag{3.5}$$

For the simulation results presented in Section 3.3 the central weight value $w_0 = 0.25$.

For the next step of the algorithm, the set of predicted sigma points must be generated using the same process as in Equation (3.1), except that the central point is the state prediction $\hat{\boldsymbol{x}}_{\mathrm{n|n\text{-}1}}$ and the covariance whose square root is taken will be the predicted state covariance $\boldsymbol{P}_{\mathrm{n|n\text{-}1}}^{\boldsymbol{xx}}$. After computing the propagated sigma points, the observation prediction becomes

$$\hat{\boldsymbol{z}}_{\mathrm{n|n\text{-}1}} = \sum_{\mathrm{j}=0}^{12} w_j \boldsymbol{H} \boldsymbol{\chi}_{\mathrm{n|n\text{-}1}}^{(\mathrm{j})}. \tag{3.6}$$

Note that matrix H is described in Equation (2.113). The predicted observation and state-observation covariances become

$$\boldsymbol{P}_{\mathrm{n|n\text{-}1}}^{\boldsymbol{zz}} = \boldsymbol{R} + \sum_{\mathrm{j}=1}^{12} w_{\mathrm{j}} \left[\boldsymbol{H} \boldsymbol{\chi}_{\mathrm{n|n\text{-}1}}^{(\mathrm{j})} - \hat{\boldsymbol{z}}_{\mathrm{n|n\text{-}1}} \right] \left[\boldsymbol{H} \boldsymbol{\chi}_{\mathrm{n|n\text{-}1}}^{(\mathrm{j})} - \hat{\boldsymbol{z}}_{\mathrm{n|n\text{-}1}} \right]^{\mathrm{T}} \tag{3.7}$$

and

$$\boldsymbol{P}_{n|n-1}^{xz} = \sum_{j=1}^{12} w_j \left[\mathbf{f}_{\mathrm{EPC}}\left(\boldsymbol{\chi}_{\mathrm{n|n\text{-}1}}^{(\mathrm{j})} \right) - \hat{\boldsymbol{x}}_{n|n-1} \right] \left[\boldsymbol{H}\boldsymbol{\chi}_{\mathrm{n|n\text{-}1}}^{(\mathrm{j})} - \hat{\boldsymbol{z}}_{\mathrm{n|n\text{-}1}} \right]^{\mathrm{T}}, \tag{3.8}$$

respectively. The remainder of the UKF algorithm, including the Kalman gain and update steps, follows the formulation given in (2.119)–(2.121).

While the computation of the sigma points can be costly for higher-dimensional state vectors, the UKF for this EPC tracking problem requires only thirteen points. This causes no significant reduction in the speed-based performance compared to the EKF. The results showing how the UKF algorithm performs in tracking maneuvering targets are demonstrated in Section 3.3.

3.1.3. EPC UKF Algorithm

The layout of the EPC UKF algorithm differs from that of the EKFs studied in Chapter 2 until it reaches the Kalman gain calculations. The algorithm, including the generation of sigma points, is presented in Table 9. The algorithm used to obtain the square root of matrices is described in the MATLAB documentation [82]. Because all covariance matrices handled in this filtering algorithm are real and nonsingular, the method described by Higham [47] was used for the EPC UKF.

Table 9: EPC UKF Algorithm

Generate Prior Sigma Points	$\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})} = \hat{\boldsymbol{x}}_{\mathrm{n\text{-}1|n\text{-}1}} + \sqrt{\boldsymbol{P}_{\mathrm{n\text{-}1|n\text{-}1}}^{xx}}\,\boldsymbol{c}^{(j)},$	(3.9)
State Prediction	$\hat{\boldsymbol{x}}_{\mathrm{n|n\text{-}1}} = \sum_{j=0}^{12} w_j \mathbf{f}_{\mathrm{EPC}}\left(\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})} \right)$	(3.10)
State Covariance Prediction	$\boldsymbol{P}_{\mathrm{n|n\text{-}1}}^{xx} = \boldsymbol{Q} + \sum_{j=0}^{12} w_j \left[\mathbf{f}_{\mathrm{EPC}}(\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})}) - \hat{\boldsymbol{x}}_{n|n-1} \right] \left[\mathbf{f}_{\mathrm{EPC}}(\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})}) - \hat{\boldsymbol{x}}_{n|n-1} \right]^{\mathrm{T}}.$	(3.11)
Generate Predicted Sigma Points	$\boldsymbol{\chi}_{\mathrm{n\text{-}1|n\text{-}1}}^{(\mathrm{j})} = \hat{\boldsymbol{x}}_{\mathrm{n|n\text{-}1}} + \sqrt{\boldsymbol{P}_{\mathrm{n|n\text{-}1}}^{xx}}\,\boldsymbol{c}^{(j)},$	(3.12)

Observation Prediction	$\hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}} = \sum_{j=0}^{12} w_j \boldsymbol{H}\boldsymbol{\chi}^{(\mathrm{j})}_{\mathrm{n\text{-}1\|n\text{-}1}}$	(3.13)
Observation Covariance	$\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n\|n\text{-}1}} = \boldsymbol{R} + \sum_{\mathrm{j}=1}^{12} w_{\mathrm{j}} \left[\boldsymbol{H}\boldsymbol{\chi}^{(\mathrm{j})}_{\mathrm{n\|n\text{-}1}} - \hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}} \right] \left[\boldsymbol{H}\boldsymbol{\chi}^{(\mathrm{j})}_{\mathrm{n\|n\text{-}1}} - \hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}} \right]^{\mathrm{T}}$	(3.14)
State-Observation Covariance	$\boldsymbol{P}^{\boldsymbol{xz}}_{n\|n-1} = \sum_{j=1}^{12} w_j \left[\mathbf{f}_{\mathrm{EPC}} \left(\boldsymbol{\chi}^{(\mathrm{j})}_{\mathrm{n\|n\text{-}1}} \right) - \hat{\boldsymbol{x}}_{n\|n-1} \right] \left[\boldsymbol{H}\boldsymbol{\chi}^{(\mathrm{j})}_{\mathrm{n\|n\text{-}1}} - \hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}} \right]^{\mathrm{T}},$	(3.15)
Kalman Gain	$\boldsymbol{K}_{\mathrm{n}} = \boldsymbol{P}^{\boldsymbol{xz}}_{\mathrm{n\|n\text{-}1}} \left[\boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n\|n\text{-}1}} \right]^{-1}$	(3.16)
State Update	$\hat{\boldsymbol{x}}_{\mathrm{n\|n}} = \hat{\boldsymbol{x}}_{\mathrm{n\|n\text{-}1}} + \boldsymbol{K}_{\mathrm{n}} (\boldsymbol{z}_{\mathrm{n}} - \hat{\boldsymbol{z}}_{\mathrm{n\|n\text{-}1}})$	(3.17)
State Covariance Update	$\boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n}} = \boldsymbol{P}^{\boldsymbol{xx}}_{\mathrm{n\|n\text{-}1}} - \boldsymbol{K}_{\mathrm{n}} \boldsymbol{P}^{\boldsymbol{zz}}_{\mathrm{n\|n\text{-}1}} \boldsymbol{K}^{\mathrm{T}}_{\mathrm{n}}$	(3.18)

3.2.EPC Particle Filter

3.2.1. Particle Filtering Background

Although the EKF and UKF algorithms offer the nonlinear filter designer a means to estimate the first two moments of the underlying distribution, directly estimating the distribution itself also has significant value. Sequential importance sampling (SIS) particle filters are a class of nonlinear filters which generate samples or "particles" and propagate them through the dynamic model, weighting them based on their respective "importance," or likelihoods [43]. Particle filters (PFs) are particularly useful for estimating stochastic systems when the underlying distributions are non-Gaussian or if the system itself is highly nonlinear.

PFs have been used in a variety of applications and continue to be a heavily researched topic. In bioinformatics, a PF was used to process gene expression measurements, classify single-cell trajectories, and predict the likelihood of developing a disease (such as cancer) [41]. PFs are also used in a variety of nonlinear filtering problems found in economics and finance [26]. Several target tracking algorithms have

also been developed that leverage a PF, including [54], although they do not use the EPC model.

SIS particle filters offer a way to perform recursive estimation of the target state probability distribution.

$$\mathrm{p}\left(\boldsymbol{x}_{\mathrm{n}} \mid \boldsymbol{z}_{1:\mathrm{n}}\right) = \sum_{i=1}^{N_{\text{samples}}} w_{\mathrm{n}}^{\mathrm{i}} \delta\left(\boldsymbol{x}_{\mathrm{n}} - \boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})}\right), \tag{3.19}$$

where N_{samples} is the number of samples used in the particle filter, $w_{\mathrm{n}}^{\mathrm{i}}$ is the set of particle weights, and $\delta(\boldsymbol{x}_{\mathrm{n}} - \boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})})$ is the delta function evaluated at the location of each particle. The PF weights are based on the ratio of the probability density function of the desired distribution to the probability density function of the distribution used.

The PF's particles are generated as

$$\boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})} = \mathbf{f}_{\mathrm{EPC}}\left(\boldsymbol{x}_{\mathrm{n-1}}^{(\mathrm{i})}\right) + \boldsymbol{v}_{\mathrm{n}}^{(\mathrm{i})}, \tag{3.20}$$

where $\boldsymbol{v}_{\mathrm{n}}$ is a vector of the samples drawn from the estimated distribution. These propagated particles are weighted based on their likelihoods and then used to compute the next iteration of the estimated distribution. The unnormalized updated particle weights $\tilde{w}_{\mathrm{n}}^{(\mathrm{i})}$ are computed using the prior particle weights $w_{\mathrm{n-1}}^{(\mathrm{i})}$,

$$\tilde{w}_{\mathrm{n}}^{(\mathrm{i})} = w_{\mathrm{n-1}}^{(\mathrm{i})} \frac{\mathrm{p}\left(\boldsymbol{z}_{\mathrm{n}} \mid \boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})}\right)\mathrm{p}\left(\boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})} \mid \boldsymbol{x}_{\mathrm{n-1}}^{(\mathrm{i})}\right)}{\mathrm{q}\left(\boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})} \mid \boldsymbol{x}_{\mathrm{n-1}}^{(\mathrm{i})}, \boldsymbol{z}_{\mathrm{n}}\right)}. \tag{3.21}$$

The term $p\left(\boldsymbol{z}_n \mid \boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})}\right)$ is the observation probability density conditioned on each particle and is known, $\mathrm{p}\left(\boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})} \mid \boldsymbol{x}_{\mathrm{n-1}}^{(\mathrm{i})}\right)$ is the state transition probability density, and $\mathrm{q}\left(\boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})} \mid \boldsymbol{x}_{\mathrm{n-1}}^{(\mathrm{i})}, \boldsymbol{z}_{\mathrm{n}}\right)$ is the importance density from which the particles are drawn. This weight calculation can be simplified in the bootstrap particle filtering algorithm [39] by assuming

$$p\left(\boldsymbol{x}_n^{(i)} \mid \boldsymbol{x}_{n-1}^{(i)}\right) = q\left(\boldsymbol{x}_n^{(i)} \mid \boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{z}_n\right). \tag{3.22}$$

This establishes that that the incremental weights do not depend on the past trajectory of the particles [43].

To extract the desired state and error covariance information provided by the EKFs and UKF, one needs only to compute the sample mean and sample error covariance using the estimated distribution. The estimated distribution is obtained by normalizing the weights set out in Equation (3.21). Once these crucial data are obtained and sent to the combat system, the set of updated particles replaces the old prior particles and is used to start the recursive filter cycle again.

SIS particle methods are particularly useful when the underlying distributions in the system under evaluation are non-Gaussian or highly nonlinear. However, they can also add significant computational cost to the filter algorithm, depending on the number of particles required to accurately represent the target state distribution [30]. Any gains in RMSE performance will have to be balanced against the potential increase in computation time. In addition, SIS particle filters can also suffer from sample degeneracy [43], wherein particles based on posterior density diverge from the true posterior density, concentrating the weights on only a few central particles. Resampling and regularization procedures can be used to re-diversify the particles, and these methods will be covered in the development of the EPC BPF.

3.2.2. The EPC Bootstrap Particle Filter

For the comparison studies covered in this chapter, we implemented the bootstrap particle filter (BPF) first introduced by Gordon et al. [39]. The BPF uses the transition

density $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$ as the importance density [43]. As a result, the computation of the unnormalized importance weights

$$\tilde{w}_n^{(i)} = \tilde{w}_{n-1}^{(i)} p(\boldsymbol{z}_n \mid \boldsymbol{x}_n^{(i)}) \tag{3.23}$$

where

$$p\left(\boldsymbol{z}_n \mid \boldsymbol{x}_n^{(i)}\right) \sim N\left(\boldsymbol{z}_n; \boldsymbol{H}\boldsymbol{x}_n^{(i)}, \boldsymbol{H}\boldsymbol{P}_{n-1}^{xx}\boldsymbol{H}^T + \boldsymbol{R}\right) \tag{3.24}$$

simplifies to

$$\tilde{w}_n^{(i)} = w_{n-1}^{(i)} p\left(\boldsymbol{z}_n \mid \boldsymbol{x}_n^{(i)}\right). \tag{3.25}$$

Normalization of the weights yields

$$w_n^{(i)} = \frac{\tilde{w}_n^{(i)}}{\sum_{i=1}^{N_{samples}} \tilde{w}_n^{(i)}}. \tag{3.26}$$

The effective number of samples, a number representing the relative diversity of the sample weights, must now be computed as

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_{samples}} \left(w_n^{(i)}\right)^2}. \tag{3.27}$$

If N_{eff} is significantly smaller than $N_{samples}$, which is the number of samples or particles used in the algorithm, then resampling and regularization processes must take place to ensure particle diversity. For the studies performed in this dissertation, the rule of thumb from [43] is used, where resampling and regularization are performed if

$$N_{eff} \leq \frac{N_{samples}}{10}. \tag{3.28}$$

The EPC BPF algorithm uses stratified resampling, as Hol et al. [49] found that it delivers better results when compared to random sampling and comparable results when compared to systematic sampling.

Stratified sampling is based on the ideas used in survey sampling, where the interval (0,1] is pre-partitioned into an N_{samples} uniform adjoint strata [49]. Within each stratum, a separate uniform random number is generated such that for the *i*th strata,

$$u_{\text{i}} \sim \text{U}\left(\frac{i}{N_{\text{samples}}}, i + \frac{1}{N_{\text{samples}}} \right). \tag{3.29}$$

The sample distribution was then reconstructed using the inverse transform method [43]. Effectively, this means taking the cumulative sum of the uniform samples to recover particles. The resampled particles are then assigned equal weights to reduce the need for further resampling. To reduce the likelihood that any of the newly resampled particles share a common weight, regularization is also performed.

Regularization prevents repeated particles from for any given weight [43]. To prevent this "sample impoverishment," a kernel density estimate of the particle density can be used to resample the particles a second time. In this process, each new particle is selected from the original resampled particle based on a draw from a uniform distribution, and the sample point is dithered based on a draw from the local individual kernel. The regularization process tends to concentrate the particles in the region with the highest probability and separates them in a random fashion. Kernel sample generation can be performed using the processes described in [98] and [99].

Once resampling and regularization have been performed, the empirical mean

$$\hat{\boldsymbol{x}}_{\text{n}} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \tilde{\boldsymbol{x}}_{\text{n}}^{(\text{i})} \tag{3.30}$$

and empirical covariance

$$\boldsymbol{P}_{\mathrm{n}}^{xx} = \boldsymbol{\Sigma}_{\mathrm{s}} = \frac{1}{N_{\mathrm{samples}}} \sum_{\mathrm{i}=1}^{N_{\mathrm{samples}}} w_{\mathrm{n}}^{(\mathrm{i})} \left(\tilde{\boldsymbol{x}}_{\mathrm{n}}^{(\mathrm{i})} - \hat{\boldsymbol{x}}_{\mathrm{n}} \right) \left(\tilde{\boldsymbol{x}}_{\mathrm{n}}^{(\mathrm{i})} - \hat{\boldsymbol{x}}_{\mathrm{n}} \right)^{\mathrm{T}} \tag{3.31}$$

of the particles must be computed using the resampled particles $\tilde{\boldsymbol{x}}_{\mathrm{n}}^{(\mathrm{i})}$. The particles are then propagated forward in time and the weights are reset to

$$w_{\mathrm{n}}^{(\mathrm{i})} = \frac{1}{N_{\mathrm{samples}}}. \tag{3.32}$$

If N_{eff} is sufficiently large, the algorithm simply moves forward by a single time step and assigns the latest particles as prior particles and the latest normalized weights as the prior weights for the next cycle of the algorithm. As in the case with less particle diversity, the particles can be used to compute the empirical mean and covariance for track estimation.

3.2.3. EPC BPF Algorithm

The BPF algorithm's performance depends on many tunable parameters but is especially sensitive to the sample count. For the results presented in Section 3.3, a sample count (N_{samples}) of 10 000 was used. The full EPC BPF algorithm is laid out in Table 10, with the sections specifying whether the effective number of samples criterion (see Equation (3.28)) is satisfied. The algorithm is also illustrated in Figure 39.

Table 10: EPC BPF Algorithm

Generate sample noise	$\boldsymbol{v}_{\mathrm{n}}^{(\mathrm{i})} \sim \mathrm{N}\left(\boldsymbol{0}, \boldsymbol{Q}_{\mathrm{n}}\right)$	(3.33)
Draw new samples	$\boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})} = \mathbf{f}_{\mathrm{EPC}}(\boldsymbol{x}_{\mathrm{n-1}}^{(\mathrm{i})}) + \boldsymbol{v}_{\mathrm{n}}^{(\mathrm{i})}$	(3.34)
Generate unnormalized weights	$\tilde{w}_{\mathrm{n}}^{(\mathrm{i})} = \tilde{w}_{\mathrm{n-1}}^{(\mathrm{i})} \mathrm{p}(\boldsymbol{z}_{\mathrm{n}} \mid \boldsymbol{x}_{\mathrm{n}}^{(\mathrm{i})})$	(3.35)
Normalize weights	$w_{\mathrm{n}}^{(\mathrm{i})} = \tilde{w}_{\mathrm{n}}^{(\mathrm{i})} \Big/ \sum_{\mathrm{i}=1}^{N_{\mathrm{samples}}} \tilde{w}_{\mathrm{n}}^{(\mathrm{i})}$	(3.36)

Calculate N_{eff}	$N_{\text{eff}} = 1 \Big/ \sum_{\text{i}=1}^{N_{\text{samples}}} \left(w_{\text{n}}^{(\text{i})} \right)^2$	(3.37)
IF: $N_{\text{eff}} << N_{\text{samples}}$:	Resample and regularize…	
Calculate empirical mean	$\boldsymbol{\mu}_{\text{s}} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \boldsymbol{x}_{\text{n}}^{(\text{i})}$	(3.38)
Calculate empirical covariance	$\boldsymbol{\Sigma}_{\text{s}} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \left(\boldsymbol{x}_{\text{n}}^{(\text{i})} - \boldsymbol{\mu}_{\text{s}} \right) \left(\boldsymbol{x}_{\text{n}}^{(\text{i})} - \boldsymbol{\mu}_{\text{s}} \right)^{\text{T}}$	(3.39)
Resample particles	Use empirical mean and covariance to resample.	
Regularize particles	Use process described in Section 3.2.2	
Calculate state update using resampled particles	$\hat{\boldsymbol{x}}_{\text{n}} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \tilde{\boldsymbol{x}}_{\text{n}}^{(\text{i})}$	(3.40)
Calculate covariance update using resampled particles	$P_{\text{n}}^{xx} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \left(\tilde{\boldsymbol{x}}_{\text{n}}^{(\text{i})} - \boldsymbol{\mu}_{\text{s}} \right) \left(\tilde{\boldsymbol{x}}_{\text{n}}^{(\text{i})} - \boldsymbol{\mu}_{\text{s}} \right)^{\text{T}}$	(3.41)
Time step: resampled particles become prior particles	$\tilde{\boldsymbol{x}}_{\text{n}}^{(\text{i})} \rightarrow \boldsymbol{x}_{\text{n-1}}^{(\text{i})}$	(3.42)
Time step: weights are re-set to be equal for all particles	$w_{\text{n}}^{(\text{i})} \rightarrow \frac{1}{N_{\text{samples}}}$	(3.43)
ELSE:	Use current particles to update track state and covariance	
Calculate covariance update using resampled particles	$\hat{\boldsymbol{x}}_{\text{n}} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \boldsymbol{x}_{\text{n}}^{(\text{i})}$	(3.44)
Calculate covariance update using resampled particles	$\boldsymbol{P}_{\text{n}}^{xx} = \frac{1}{N_{\text{samples}}} \sum_{\text{i}=1}^{N_{\text{samples}}} w_{\text{n}}^{(\text{i})} \left(\boldsymbol{x}_{\text{n}}^{(\text{i})} - \hat{\boldsymbol{x}}_{\text{n}} \right) \left(\boldsymbol{x}_{\text{n}}^{(\text{i})} - \hat{\boldsymbol{x}}_{\text{n}} \right)^{\text{T}}$	(3.45)
Time step: latest particles become prior particles	$\boldsymbol{x}_{\text{n}}^{(\text{i})} \rightarrow \boldsymbol{x}_{\text{n-1}}^{(\text{i})}$	(3.46)
Time step: latest weights become prior weights	$w_{\text{n}}^{(\text{i})} \rightarrow w_{\text{n-1}}^{(\text{i})}$	(3.47)

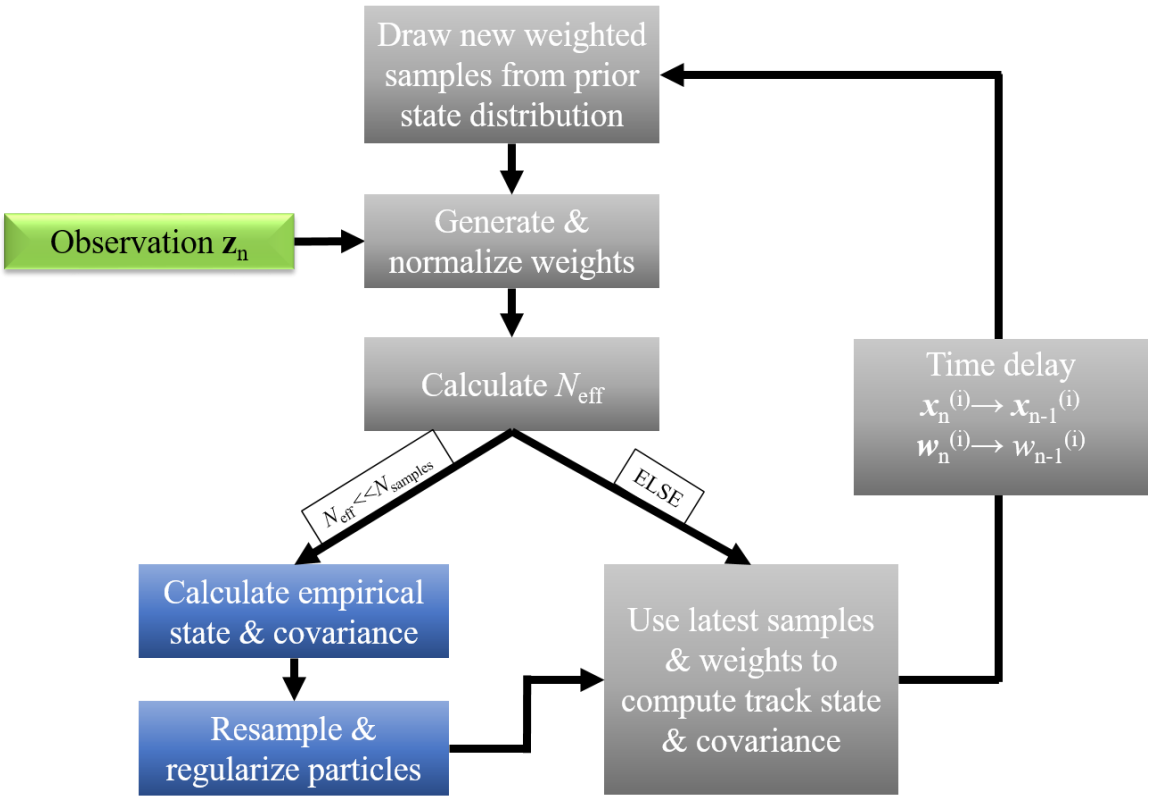

Figure 39: BPF algorithm flowchart

3.3. Performance Study of EPC Filters

3.3.1. Purpose of Study

In Sections 3.1 and 3.2, two more discrete-discrete (D-D) filters using the EPC model were developed as alternatives to the EPC EKF. These filters were compared with the EPC EKF using the same metrics used to evaluate the EPC EKF in Section 2.6. The goal of this study was to establish whether the tracking error performance for each target can be improved by implementing the EPC model in the UKF and BPF algorithms. At the outset of this study, it was expected that these filters, with their greater ability to handle highly nonlinear systems, may also ameliorate the non-NRI target tracking performance degradation described in Section 2.6. If either the EPC UKF or EPC BPF can offer better

performance than the EPC EKF in both NRI and non-NRI cases, then the winner will clearly be the superior filter.

Although large performance gains against NRI targets were established in Section 2.6, it is worth noting that error performance improvements of even one percent have value in target tracking. In such a heavily mined field, a target tracker with a few percent or a few tenths of a percent improved performance could prove valuable to a combat system. However, if the same target tracking filter fails against commonly used non-NRI targets, it may be necessary to implement that filter as part of a larger algorithm to compensate. This study also identifies candidates for a multiple-model EPC algorithm to address a broad set of trajectories if the EPC UKF or EPC BPF fails to address the challenges faced by the EPC EKF in Section 2.6.

3.3.2. Radially Inbound Target

For the radially inbound target, the angle errors of the EPC filters demonstrated a clear pattern. The EPC BPF holds the edge over the other two filters throughout the trajectory for both the bearing (Figure 40) and elevation (Figure 41) RMSE, consistently delivering the lowest RMSE for both angles. The EPC UKF is also superior to the EPC EKF. In the box plots (Figure 42–Figure 43), the EPC BPF exhibited the best performance in terms of bearing, bearing rate, elevation, and elevation rate, despite some outliers. However, this improved angle error performance comes at the cost of a slightly degraded range and a range rate error performance. Based on this first experiment, the EPC UKF and EPC BPF appear to be promising alternatives to the EPC EKF.

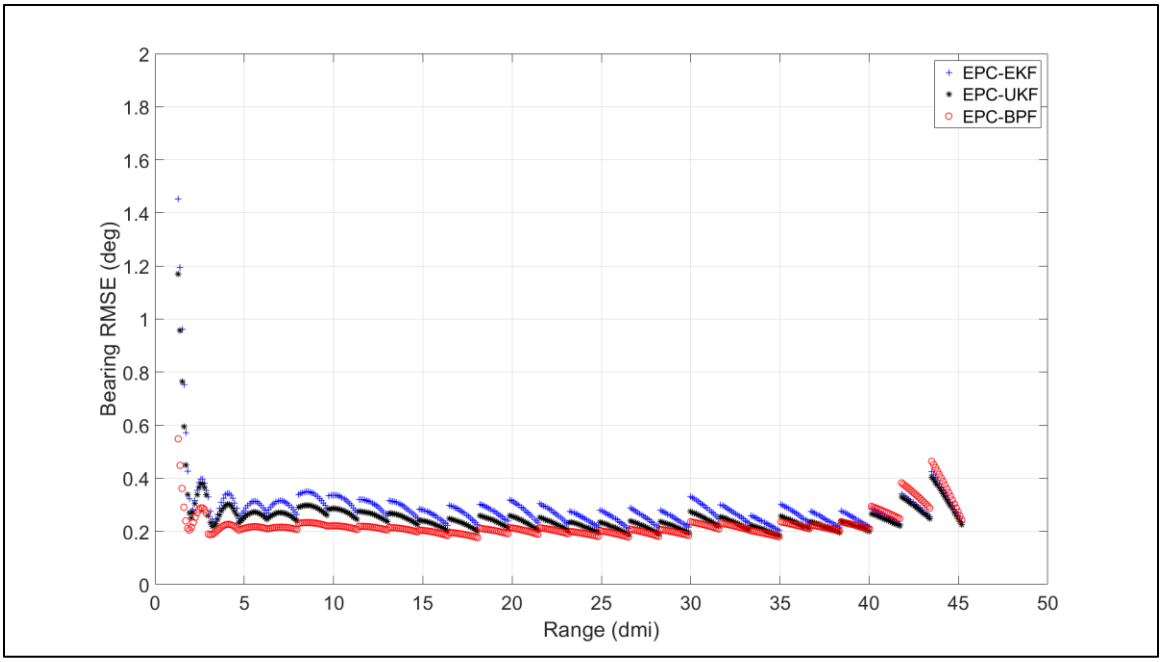

Figure 40: EPC filter comparison study – bearing RMSE for radially inbound scenario

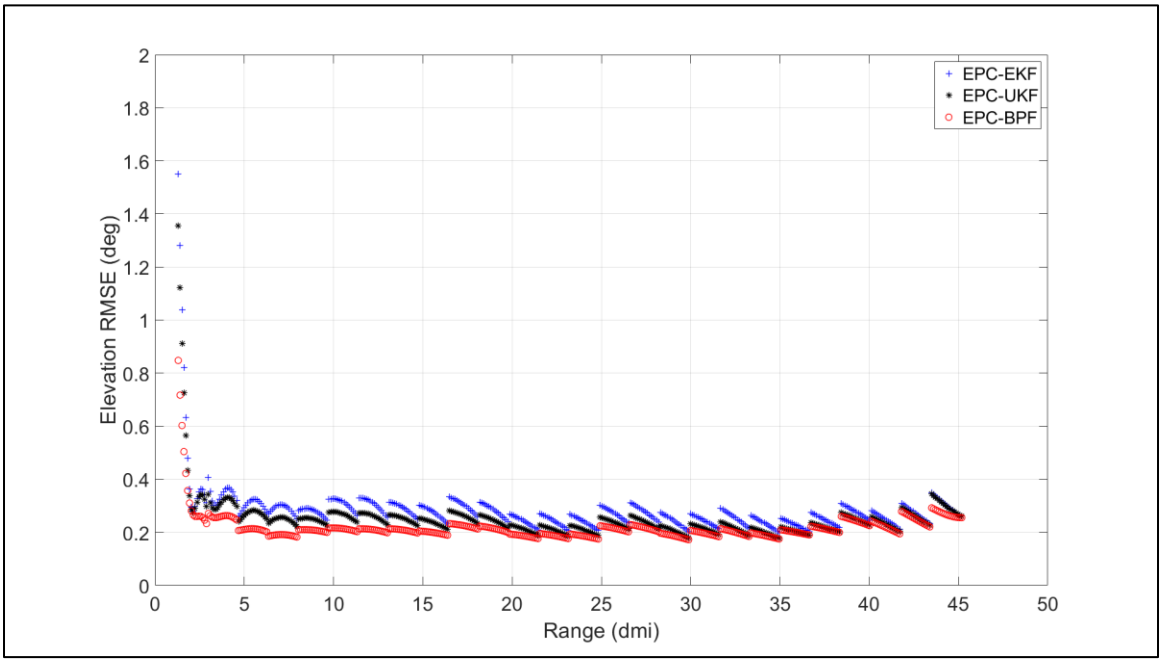

Figure 41: EPC filter comparison study – elevation RMSE for radially inbound scenario

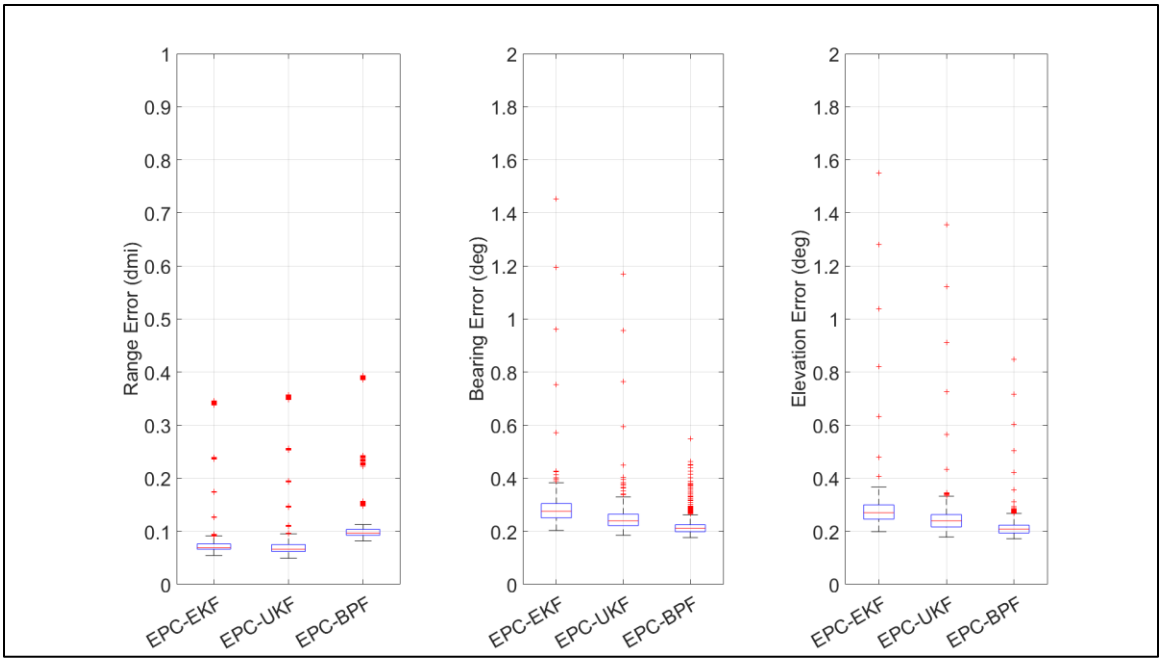

Figure 42: EPC filter comparison study – position error box plots for radially inbound scenario

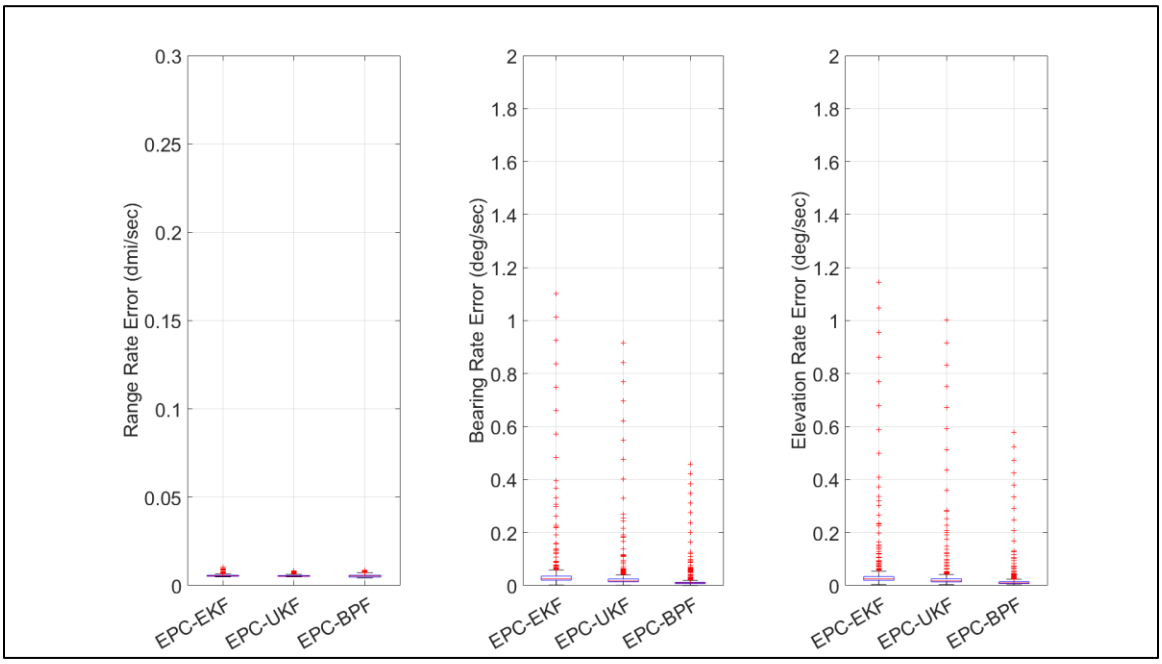

Figure 43: EPC filter comparison study – velocity error box plots for radially inbound scenario

3.3.3. Dutch Roll Target

For the Dutch roll scenario, the EPC BPF continues to have a performance advantage over the EPC UKF and EPC EKF. The bearing RMSE performance gains from the BPF over the other two filters continue throughout the target's trajectory, and the BPF retains its advantage throughout the weaving maneuver, as shown in Figure 44. The EPC BPF was especially dominant during the fully radially inbound portion of the trajectory. The elevation errors are similarly better for the EPC BPF, as shown in Figure 45. The EPC UKF also performed better than the EPC EKF, but not by the same margin. The box plots in Figure 46 and Figure 47 for this scenario bear out the findings from the RMSE plots as a function of the range. The EPC BPF displayed superior RMSE performance in all dimensions, including the range and range rate.

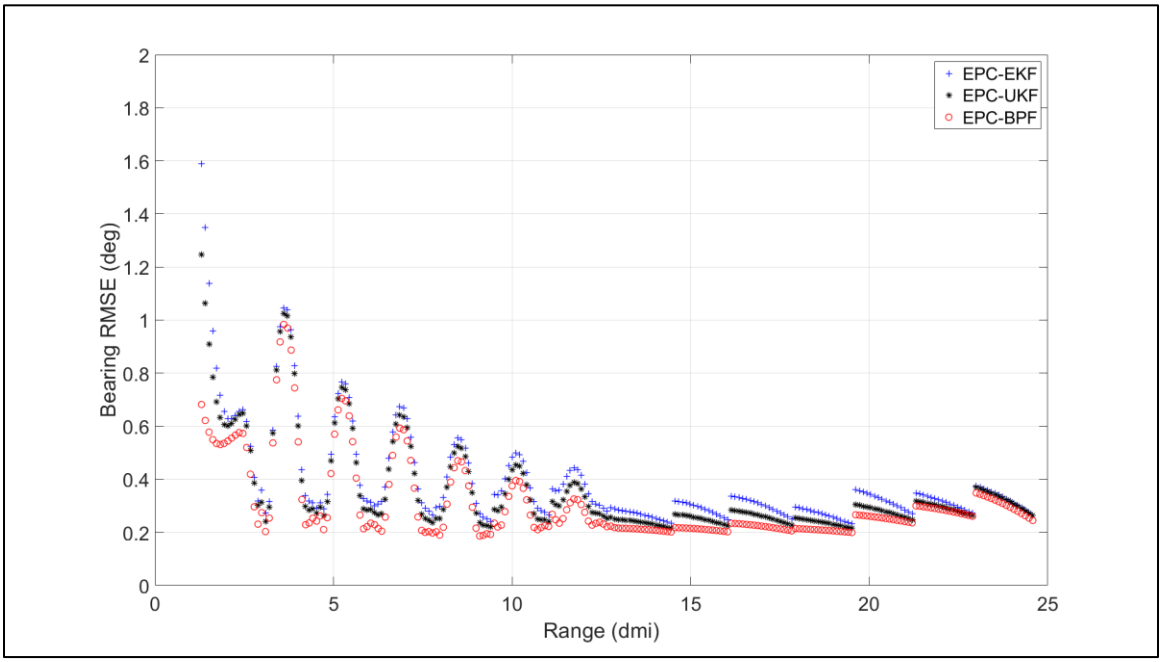

Figure 44: EPC filter comparison study – bearing RMSE for Dutch roll scenario

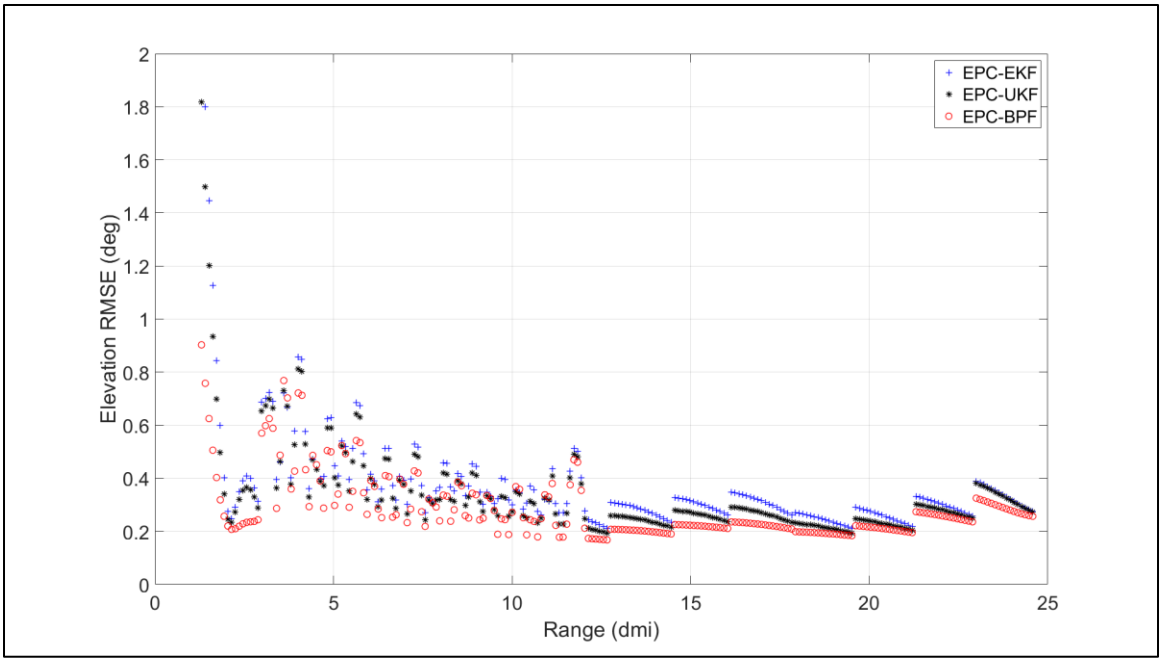

Figure 45: EPC filter comparison study – elevation RMSE for Dutch roll scenario

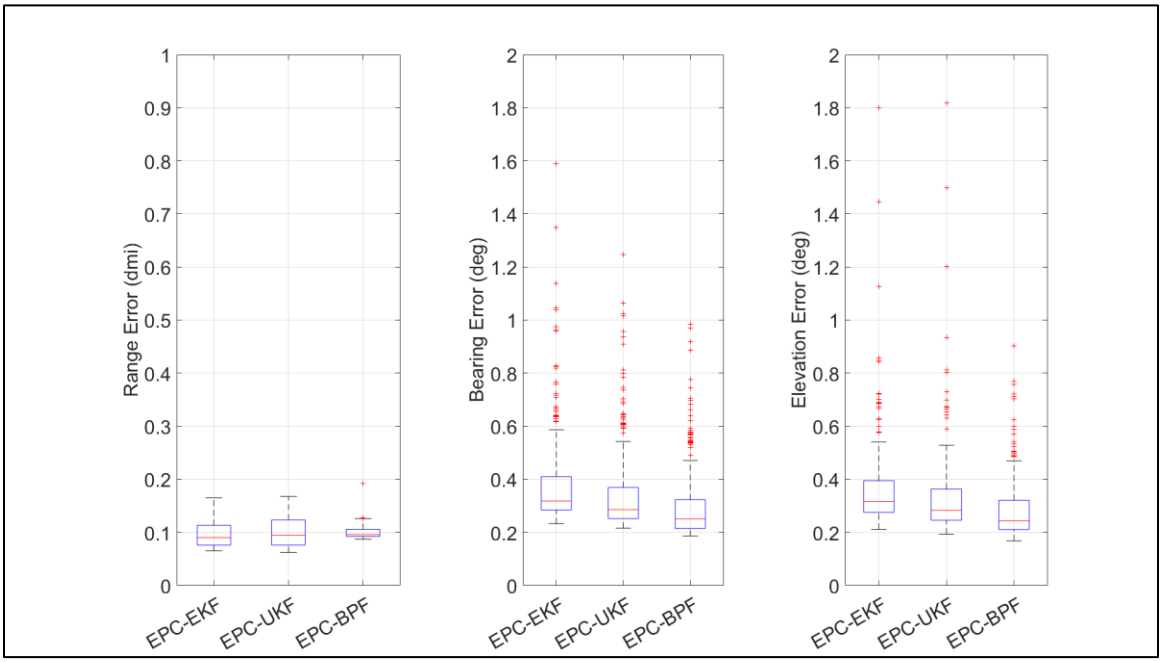

Figure 46: EPC filter comparison study – position error box plots for Dutch roll scenario

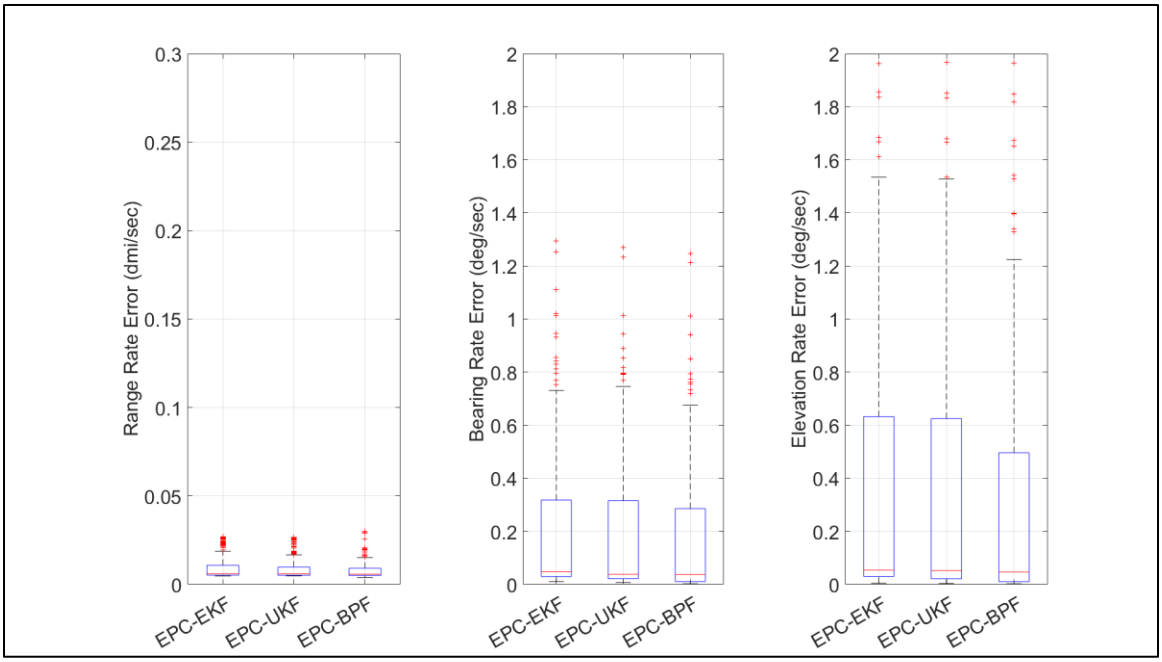

Figure 47: EPC filter comparison study – velocity error box plots for Dutch roll scenario

3.3.4. High-diver Target

Any expectation for good tracking performance against non-NRI targets was broken after evaluating the high-diver target. All three filters struggled to properly track the target. Apparently, the ability to handle higher levels of nonlinearity is not sufficient to overcome the model mismatch presented by this non-NRI target. Figure 48 and Figure 49 clearly demonstrate that the angular tracks diverge as the target motion becomes more less and less NRI. The RMSE statistics gathered in Figure 50 and Figure 51 show that the EPC BPF, which performed well for NRI targets, is also the worst filter when applied to non-NRI targets. The EPC BPF exhibited the worst performance across all position and velocity states. Although the EPC EKF and EPC UKF deliver reasonable error levels when compared to the legacy filters examined in Section 2.6, it is clear that simply changing the filter type will not solve the non-NRI problem.

A notable finding in this analysis is that when the high-diver made its downward turn around 10 dmi RTG, both the EPC EKF and EPC UKF saw their elevation errors drop dramatically. The EPC BPF struggles the most with this scenario and features significant track divergence. For this scenario, the RMSE plots as a function of range (Figure 48 and Figure 49) emphasize the changeover from non-NRI to NRI behavior and the resulting degradation of performance, especially in the period leading up to the dive maneuver.

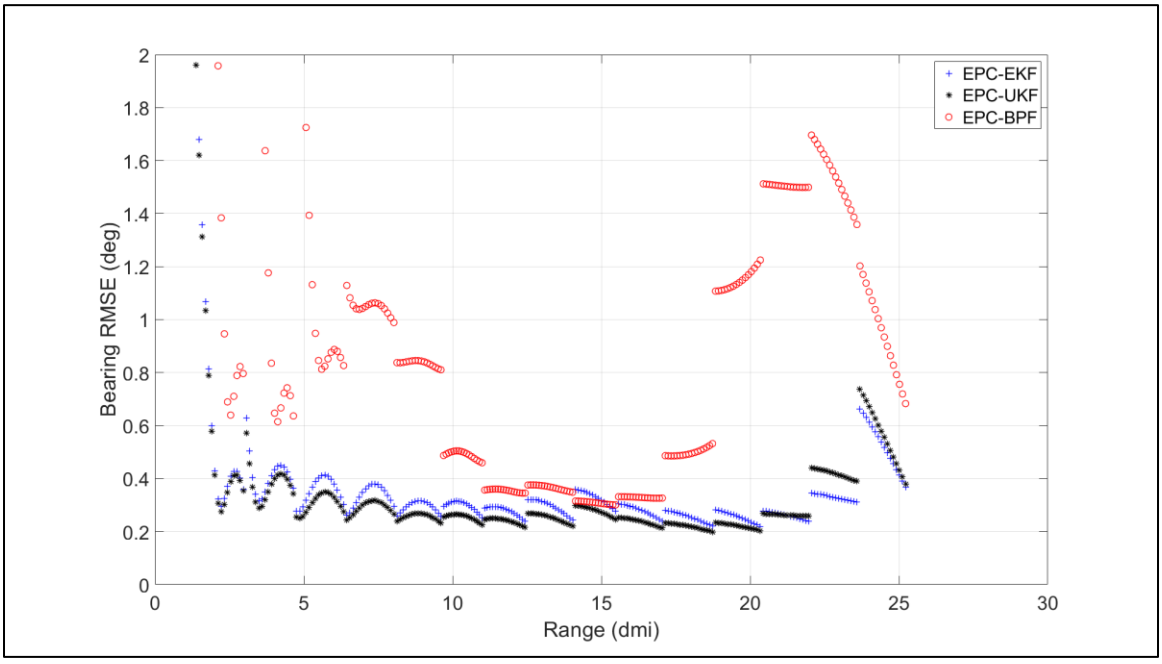

Figure 48 EPC filter comparison study – bearing RMSE for high-diver scenario

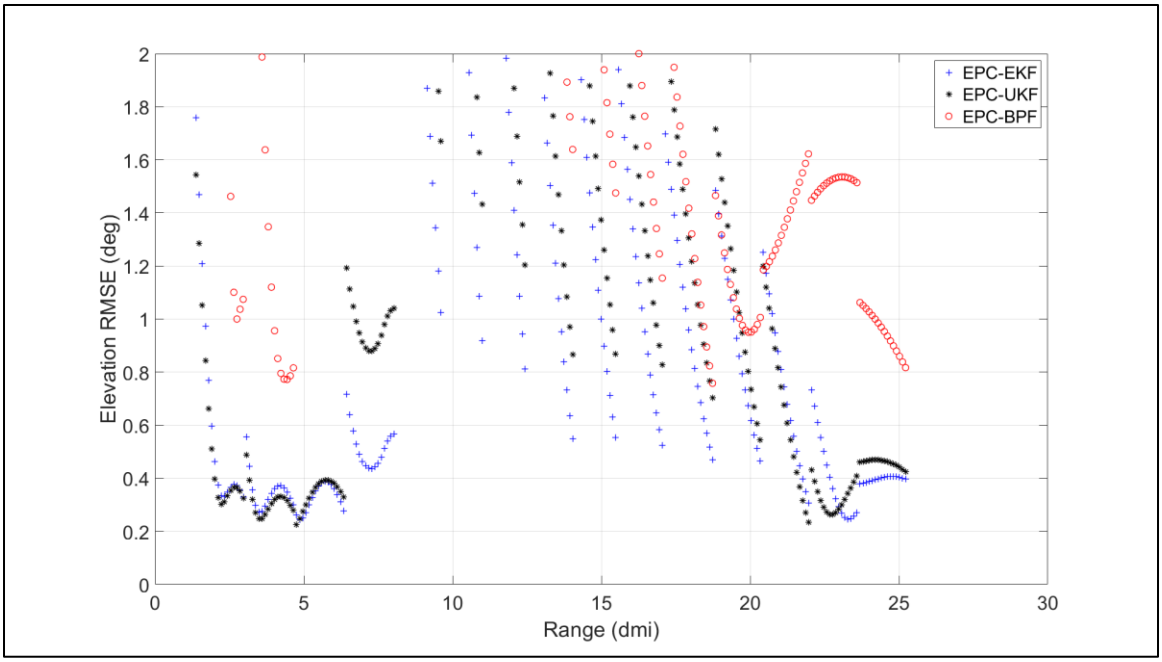

Figure 49: EPC filter comparison study – elevation RMSE for high-diver scenario

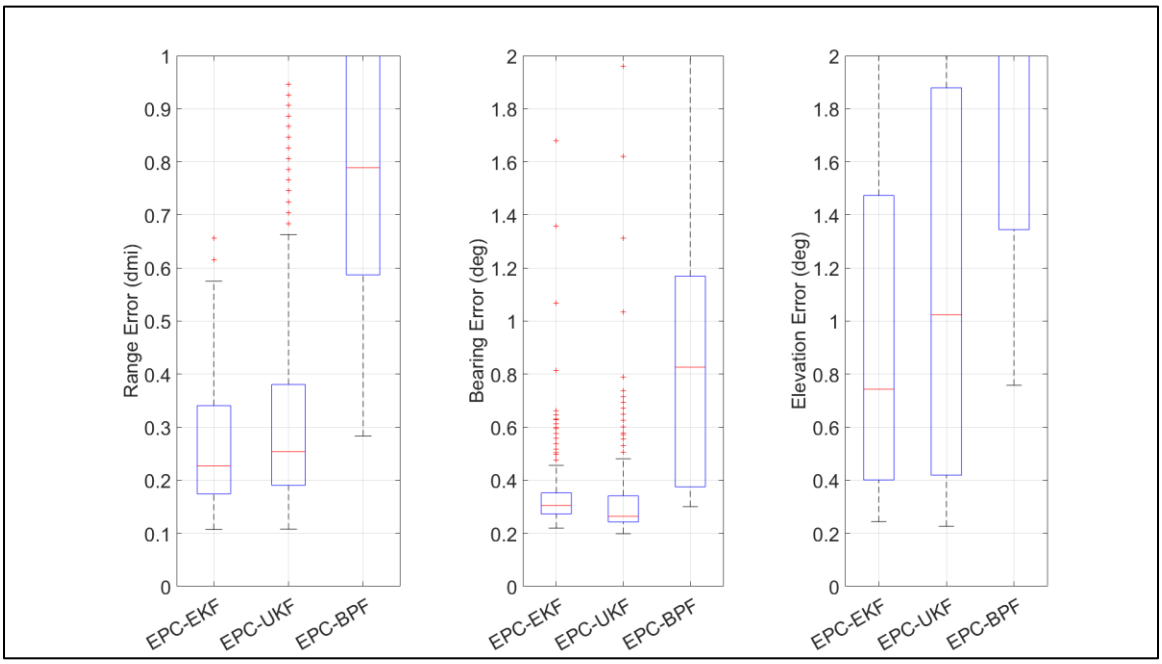

Figure 50: EPC filter comparison study – position error box plots for high-diver scenario

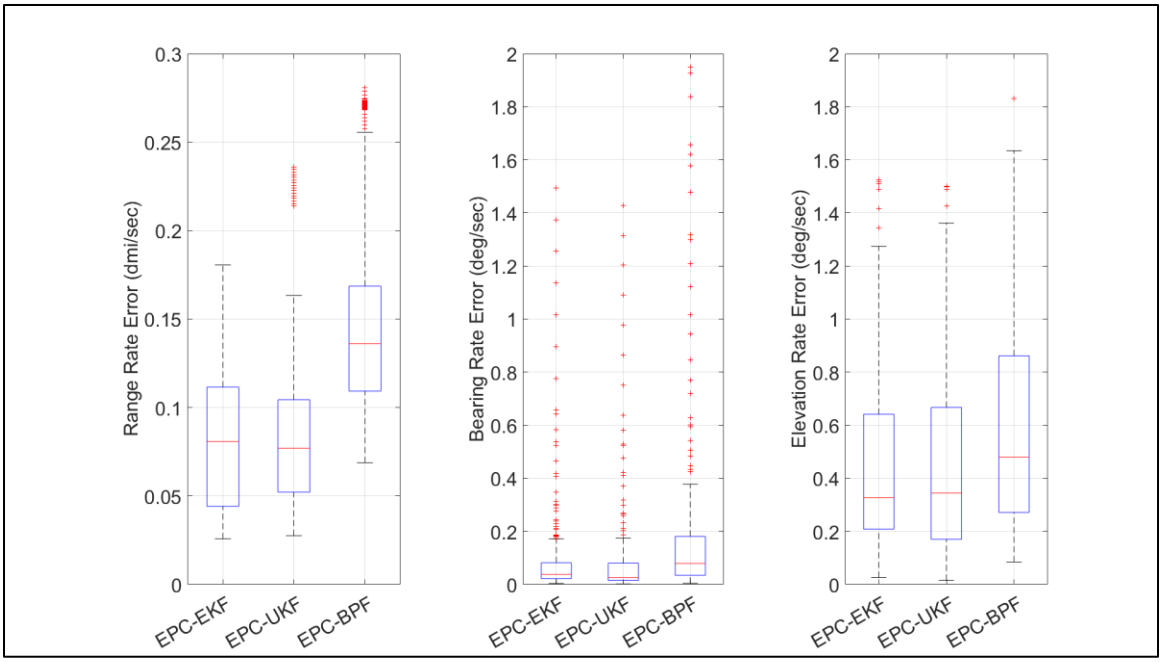

Figure 51: EPC filter comparison study – velocity error box plots for high-diver scenario

3.3.5. Dogleg Target

Similar to the high-diver, the dogleg target is an example where non-NRI target motion has an impact on the EPC filter tracking performance. In this case, the bearing error performance was affected by the non-NRI behavior in the horizontal plane. This was again because of the model mismatch introduced by the EPC. For the dogleg target, the EPC EKF had the best bearing RMSE performance (see Figure 52), although it was previously found that this performance was poor compared to the legacy C-EKF and S-EKF. The EPC BPF features the best elevation RMSE performance, as shown in Figure 53, which is consistent with its performance against other targets. The EPC UKF track error performance was between those of the other two EPC filters.

All filters examined in this study diverged at approximately 10 dmi RTG for the dogleg scenario as the target became more non-NRI. The box plots (Figure 54-Figure 55) tell a similar story, with all three filters featuring poor performance in bearing and

bearing rate but otherwise acceptable performance (still superior to the legacy C-EKF and S-EKF) in the other dimensions. After considering the high-diver target, it appears that a more flexible solution is required to improve the tracking performance against non-NRI targets.

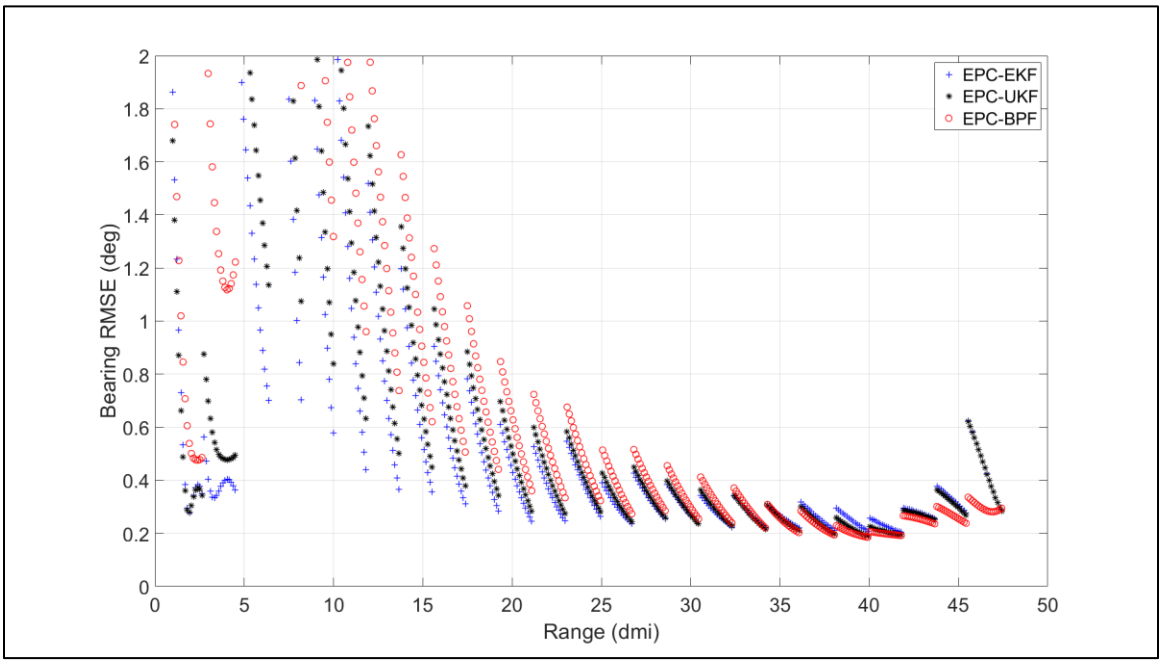

Figure 52: EPC filter comparison study – bearing RMSE for dogleg scenario

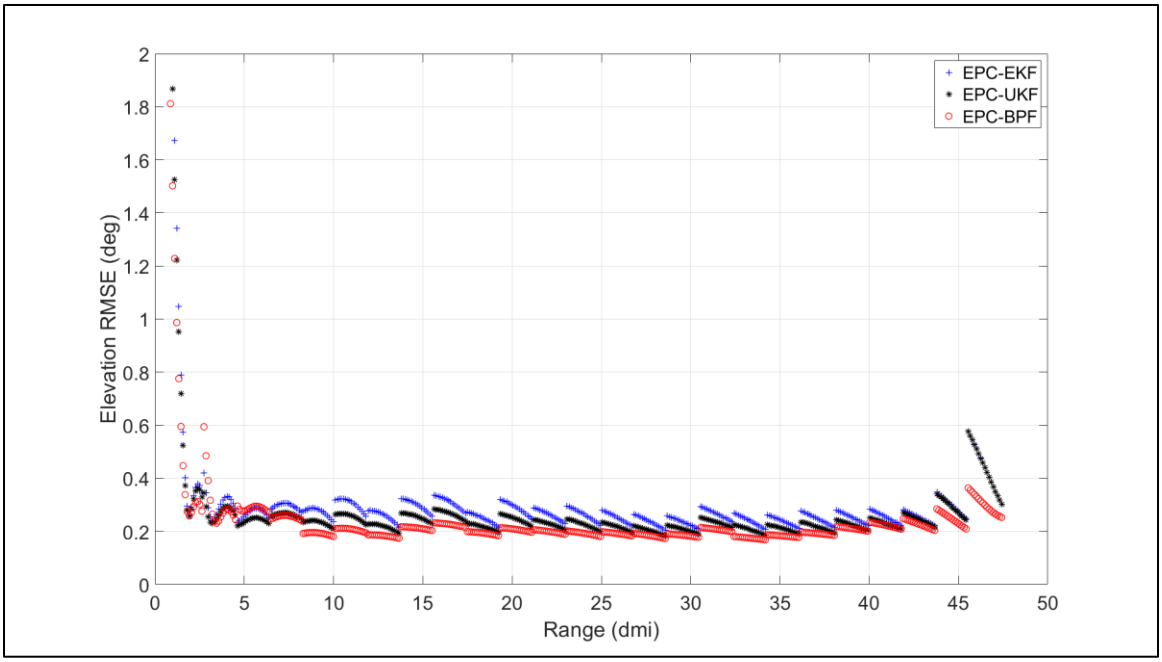

Figure 53: EPC filter comparison study – elevation RMSE for dogleg scenario

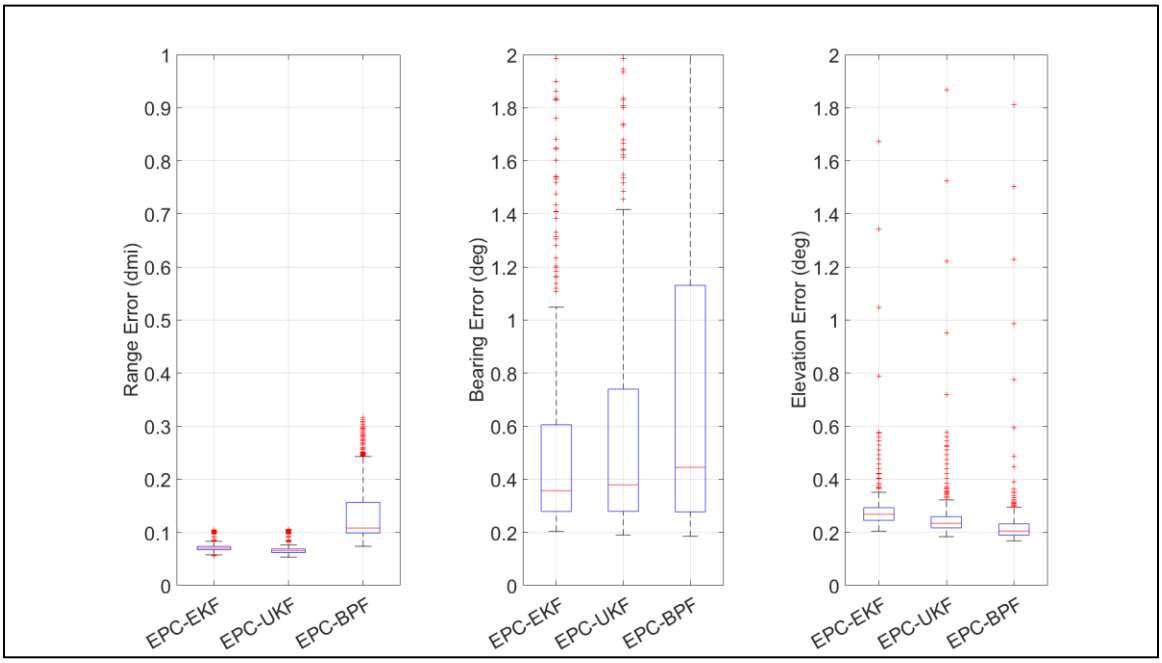

Figure 54: EPC filter comparison study – position error box plots for dogleg scenario

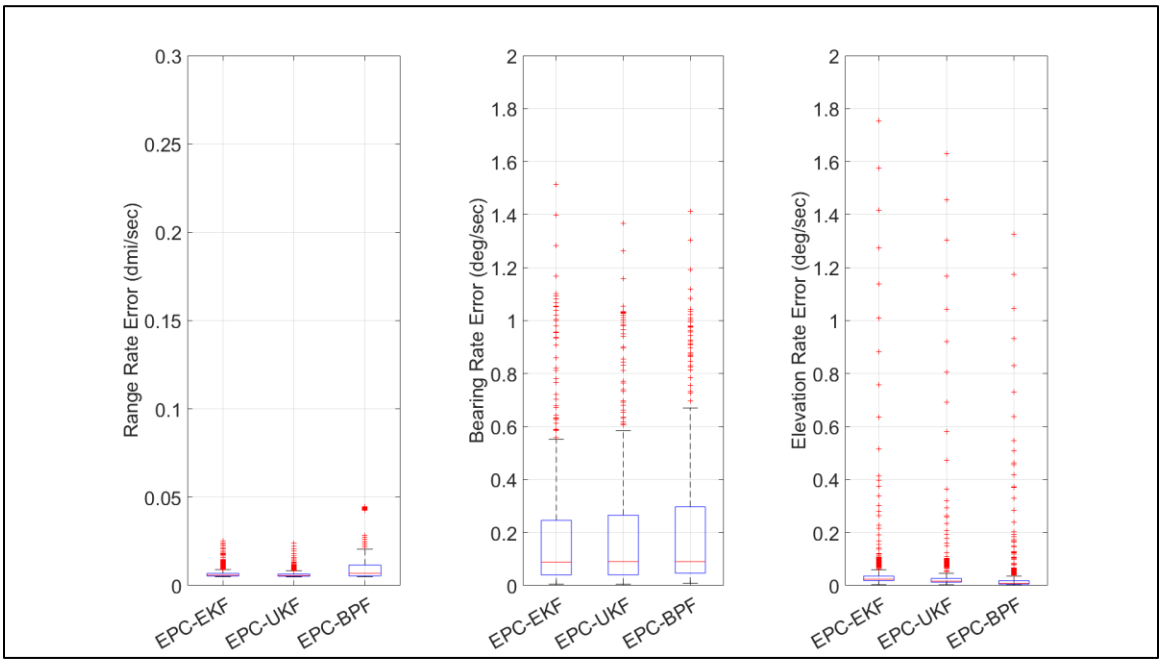

Figure 55: EPC filter comparison study – velocity error box plots for dogleg scenario

3.3.6. EPC Filter Comparison Overall Performance Assessment

In the second major performance study, it is important to consider both the track error performance and run time when declaring the superior filter. When considering both RMSE (see Table 11) and algorithm execution speed, the EPC UKF is the winner despite the EPC BPF delivering the best error performance for the two NRI targets. The EPC EKF was previously found to be a real-time algorithm in Section 2.6, and the EPC UKF adds only a small amount to the computation time. The EPC BPF, however, was much slower, clocking in 9.24% of the scenario time for the radially inbound target (see Table 12). This means that for each radar observation (every 4 seconds), the system requires about 0.37 seconds to update the track. For the scenarios examined, a sample count of N_{samples}=10 000 was required to achieve the best RMSE performance for the EPC BPF. Using a smaller number of samples to speed up the algorithm would result in a loss of accuracy to the extent that the EPC BPF would lose its RMSE advantage over other filters.

Table 11: EPC filter comparison study – overall RMSE performance

Scenario	Filter Name	RMS Error Overall					
		Range (dmi)	Range Rate (dmi/s)	Bearing (deg)	Bearing Rate (deg/s)	Elevation (deg)	Elevation Rate (deg/s)
Radial Inbound	EPC EKF	0.09343	0.07513	0.28745	0.05160	0.28287	0.05184
	EPC UKF	0.09442	0.07398	0.25274	0.03906	0.25075	0.04044
	EPC BPF	0.11568	0.07552	0.22396	0.02111	0.21776	0.02343
Dutch Roll	EPC EKF	0.09689	0.09744	0.39311	0.20701	0.37408	0.37431
	EPC UKF	0.10068	0.09635	0.35663	0.19339	0.33925	0.35986
	EPC BPF	0.10548	0.08991	0.30942	0.17260	0.28714	0.32941
High-diver	EPC EKF	0.35472	0.29001	0.34609	0.10642	1.11607	0.43779
	EPC UKF	0.41911	0.29563	0.32329	0.09248	1.37374	0.44054
	EPC BPF	1.04987	0.39317	1.40074	0.94943	2.87045	0.85031
Dogleg	EPC EKF	0.07196	0.08432	0.59988	0.20765	0.29580	0.06454
	EPC UKF	0.06770	0.08067	0.68773	0.20666	0.26404	0.05360
	EPC BPF	0.13600	0.10030	0.91049	0.22682	0.23460	0.04659

Table 12: EPC filter comparison study – runtime performance table

Scenario	Filter Name	Scenario Duration (sec)	Average Runtime (sec)	Runtime Ratio	Real Time?
Radial Inbound	EPC-EKF	117	0.00191	0.0016%	Yes
	EPC-UKF	117	0.00956	0.0082%	Yes
	EPC-BPF	117	11.70725	10.0062%	No
Dutch Roll	EPC-EKF	69	0.00087	0.0013%	Yes
	EPC-UKF	69	0.00396	0.0057%	Yes
	EPC-BPF	69	6.37953	9.2457%	No
High-diver	EPC-EKF	73	0.00144	0.0020%	Yes
	EPC-UKF	73	0.00571	0.0078%	Yes
	EPC-BPF	73	7.58258	10.3871%	No
Dogleg	EPC-EKF	108	0.00146	0.0014%	Yes
	EPC-UKF	108	0.00680	0.0063%	Yes
	EPC-BPF	108	10.64164	9.8534%	No

While it is troublesome that the EPC UKF and EPC BPF did not solve all the issues discovered in the legacy filter study, the results are still quite valuable. The main

benefit is that we were able to determine which of these three EPC filters would function best as part of a multiple-model algorithm. Based on the collected data, the EPC UKF outperformed the EPC EKF in almost all dimensions for both NRI targets. However, the EPC BPF was the winner in terms of the overall performance of NRI targets. The real-time nature of the EPC UKF makes it a more attractive candidate for use in a multiple-model algorithm than the EPC BPF. Thus, owing to their similar speed performance and excellent NRI tracking performance, both the EPC EKF and EPC UKF will be used as core filters in the multiple model tracking algorithm. If implemented properly, these algorithms can track both types of target.

4. Robust NRI Target Tracking Using Multiple Models

4.1.Multiple Model Algorithm Background

Target maneuverability continues to grow alongside the measures intended to track and defeat them. The proportional navigation constant *k* effectively acts as a measure of the extent to which the NRI model suits a problem. Whereas a high *k* value denotes a target that closely follows a radially inbound trajectory, a *k* value of zero in the EPC EKF allows the recovery of the S-EKF with no EPC. In the event that a target is so maneuverable that part of its trajectory is *not* NRI, the EPC filters developed in Chapters 2 and 3 may not be preferable because the corrective accelerations in the dynamic model have been shown to increase the tracking errors. EPC performance degradation in the face of non-NRI target motion was demonstrated for the dogleg and high-diver trajectories evaluated in Sections 2.6 and 3.3. Implementing a multiple-model algorithm has been found to alleviate this problem by fusing the weighted outputs of a non-EPC filter, such as the S-EKF, with and EPC filter.

A plethora of algorithms have been developed to fuse weighted filter outputs. These algorithms include non-switching models, such as the autonomous multiple model algorithm (AMM), which does not feature any communication between models and performs a simple fusion based on probabilities calculated at the output of each filter. Non-switching models are fairly easy to implement but have been shown to have inferior performance compared with switching models [38]. Various switching models include generalized pseudo-Bayesian (GPB), interacting multiple model (IMM), and B-best multiple model (BMM) algorithms. A thorough study by Pitre et al. [91] found that the IMM offers an error performance comparable to that of other switching models with the

least computational complexity. Multiple models representing different target dynamics can be efficiently managed by an IMM [37]; hence, it is the algorithm used in this dissertation.

The basic algorithm structure for an IMM algorithm is shown in Figure 56. The bank of filters runs in parallel, and the model probabilities are updated based on the residual and residual covariance of each filter. The interaction algorithm then sends updated weights to the fusion portion of the algorithm, which combines the output of each filter to form a fused estimate of the state and covariance. IMMs have been used in various applications, including human motion tracking [33], air traffic control [69], and target tracking [37]. Both Gomes [38] and Mazor et al. [84] provided excellent IMM algorithm descriptions for execution in any application.

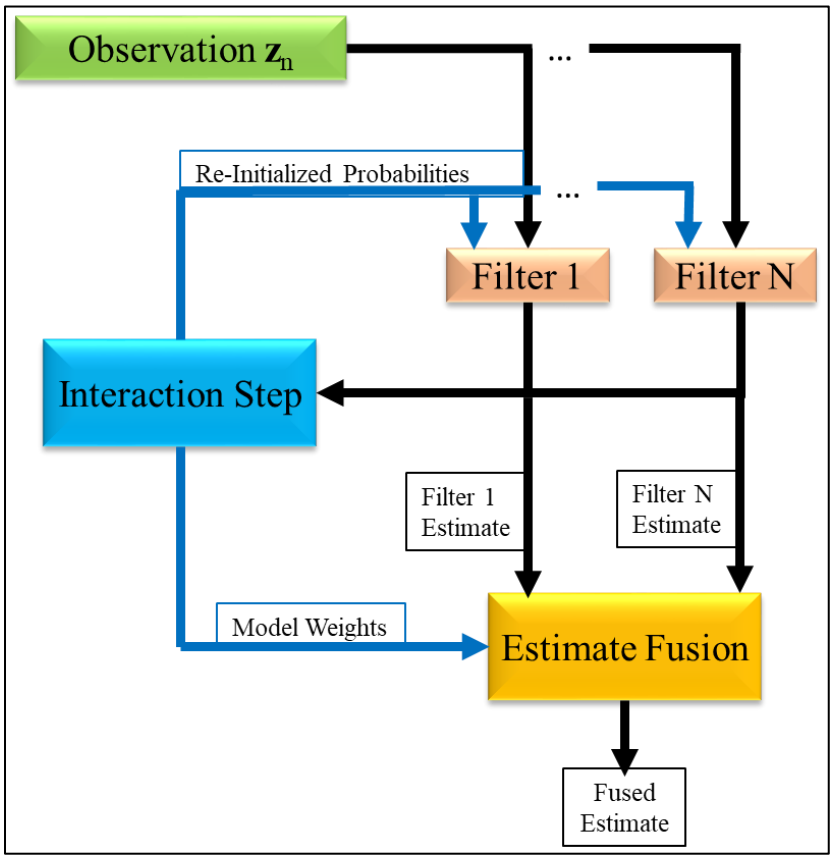

Figure 56: IMM algorithm block diagram

4.2. A Pair of EPC IMMs

4.2.1. EPC EKF IMM Implementation

An EPC IMM that utilizes a bank of EPC EKFs with different *k* values was developed as a first attempt to resolve the errors introduced by non-NRI target motion. By tuning this IMM properly, the algorithm can automatically switch between models to favor the model that best captures the target motion behavior. The basic IMM structure for each filter model is well known and is described in Table 13, with *i* superscripts denoting the outcome for the *i*th filter model computation. This structure was developed using methods described in [38] and [84].

The Markov state transition probability matrix (**$\boldsymbol{\pi}$**) is a key parameter of the IMM. This matrix is a design specification selected by the filter developer, and the indices (*i*,*j*) refer to the different models handled by the IMM. The matrix diagonals (π_{ii}) represent the probability of remaining in model state *i*, while the other elements in each row represent the probability of "switching", or transitioning, to that state from the *i*th state. The first EPC IMM to be tested in this dissertation features two filters, an EPC EKF with PN constant *k*=10, and an S-EKF, also represented by a PN constant *k*=0. For this tracking algorithm, the transition matrix is given by:

$$\boldsymbol{\pi} = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}. \tag{4.1}$$

Table 13: EPC IMM Algorithm

Predicted Model Probability	$\mu_{n\|n-1}^{(i)} = \sum_j \pi_{ji} \mu_{n-1}^{(j)}$	(4.2)
Mixing Probabilities	$\mu_{n-1}^{(j\|i)} = \pi_{ji} \mu_{n-1}^{(j)} / \mu_{n\|n-1}^{(i)}$	(4.3)
Mixing Estimate	$\bar{\boldsymbol{x}}_{n-1\|n-1}^{(i)} = \sum_j \hat{\boldsymbol{x}}_{n-1\|n-1}^{(i)} \mu_{n-1}^{(j\|i)}$	(4.4)
Mixing Covariance	$\bar{\boldsymbol{P}}_{n-1\|n-1}^{\boldsymbol{xx},(i)} = \sum_j \left[\boldsymbol{P}_{n-1\|n-1}^{\boldsymbol{xx},(i)} + \left(\bar{\boldsymbol{x}}_{n-1\|n-1}^{(i)} - \hat{\boldsymbol{x}}_{n-1\|n-1}^{(j)} \right) \left(\bar{\boldsymbol{x}}_{n-1\|n-1}^{(i)} - \hat{\boldsymbol{x}}_{n-1\|n-1}^{(j)} \right)^T \mu_{n-1}^{(j\|i)} \right]$	(4.5)
State Prediction	$\hat{\boldsymbol{x}}_{n\|n-1}^{(i)} = \boldsymbol{F} \hat{\boldsymbol{x}}_{n-1\|n-1}^{(i)} + \boldsymbol{G}\mathbf{u}_{\text{EPC}}(\hat{\boldsymbol{x}}_{n-1\|n-1}^{(i)})$	(4.6)
State Covariance	$\boldsymbol{P}_{n\|n-1}^{\boldsymbol{xx},(i)} = \hat{\boldsymbol{F}}_{\text{EPC}}(\hat{\boldsymbol{x}}_{n-1\|n-1}^{(i)}) \bar{\boldsymbol{P}}_{n-1\|n-1}^{\boldsymbol{xx},(i)} \left[\hat{\boldsymbol{F}}_{\text{EPC}}(\hat{\boldsymbol{x}}_{n-1\|n-1}^{(i)}) \right]^T + \boldsymbol{Q}(\hat{\boldsymbol{x}}_{n-1\|n-1}^{(i)})$	(4.7)
Observation Prediction	$\hat{\boldsymbol{z}}_{n\|n-1}^{(i)} = \boldsymbol{H} \hat{\boldsymbol{x}}_{n\|n-1}^{(i)}$	(4.8)
Observation Covariance	$\boldsymbol{P}_{n\|n-1}^{\boldsymbol{zz},(i)} = \boldsymbol{H} \boldsymbol{P}_{n\|n-1}^{\boldsymbol{xx},(i)} \boldsymbol{H}^T + \boldsymbol{R}$	(4.9)
State-Observation Covariance	$\boldsymbol{P}_{n\|n-1}^{\boldsymbol{xz},(i)} = \boldsymbol{P}_{n\|n-1}^{\boldsymbol{xx},(i)} \boldsymbol{H}^T$	(4.10)
Kalman Gain	$\boldsymbol{K}_n^{(i)} = \boldsymbol{P}_{n\|n-1}^{\boldsymbol{xz},(i)} \left[\boldsymbol{P}_{n\|n-1}^{\boldsymbol{zz},(i)} \right]^{-1}$	(4.11)
Model State Update	$\hat{\boldsymbol{x}}_{n\|n}^{(i)} = \hat{\boldsymbol{x}}_{n\|n-1}^{(i)} + \boldsymbol{K}_n^{(i)} (\boldsymbol{z}_n - \hat{\boldsymbol{z}}_{n\|n-1}^{(i)})$	(4.12)
Model State Covariance	$\boldsymbol{P}_{n\|n}^{\boldsymbol{xx},(i)} = \boldsymbol{P}_{n\|n-1}^{\boldsymbol{xx},(i)} - \boldsymbol{K}_n^{(i)} \boldsymbol{P}_{n\|n-1}^{\boldsymbol{zz}} \left[\boldsymbol{K}_n^i \right]^T$	(4.13)
Model Likelihood	$L_n^{(i)} \overset{assume}{=} \mathcal{N}(\boldsymbol{z}_n - \hat{\boldsymbol{z}}_{n\|n-1}^{(i)}; \boldsymbol{0}, \boldsymbol{P}_{n\|n-1}^{\boldsymbol{zz},(i)})$	(4.14)
Model Probability	$\mu_n^{(i)} = \mu_{n\|n-1}^{(i)} L_n^{(i)} / \left[\sum_j \mu_{n\|n-1}^{(j)} L_n^{(j)} \right]$	(4.15)
Fused State Estimate	$\hat{\boldsymbol{x}}_{n\|n} = \sum_i \hat{\boldsymbol{x}}_{n\|n}^{(i)} \mu_n^{(i)}$	(4.16)
Fused Covariance	$\boldsymbol{P}_{n\|n}^{\boldsymbol{xx}} = \sum_i \left[\boldsymbol{P}_{n\|n}^{\boldsymbol{xx},(i)} + (\hat{\boldsymbol{x}}_{n\|n} - \hat{\boldsymbol{x}}_{n\|n}^{(i)})(\hat{\boldsymbol{x}}_{n\|n} - \hat{\boldsymbol{x}}_{n\|n}^{(i)})^T \right] \mu_n^{(i)}$	(4.17)

The EPC IMM filter can switch between NRI (k=10) and non-NRI (k=0) models autonomously; therefore, a priori knowledge of the target's NRI-related behavior is not required. For NRI targets, the performance of the EPC EKF IMM is similar to that of the EPC EKF. Against the Dutch roll weaving target, the EPC IMM filter RMSE performance was virtually the same as that of the EPC EKF (with performance remaining

superior to the C-EKF and S-EKF). The RMSE plots appear nearly identical to those found in 2.6.4, Figure 30–Figure 31. All models were assigned equal initial probabilities. In Figure 57, the probability assigned to the model (k = 10) quickly increases, giving a similar performance throughout the track to the EPC EKF with the same k value.

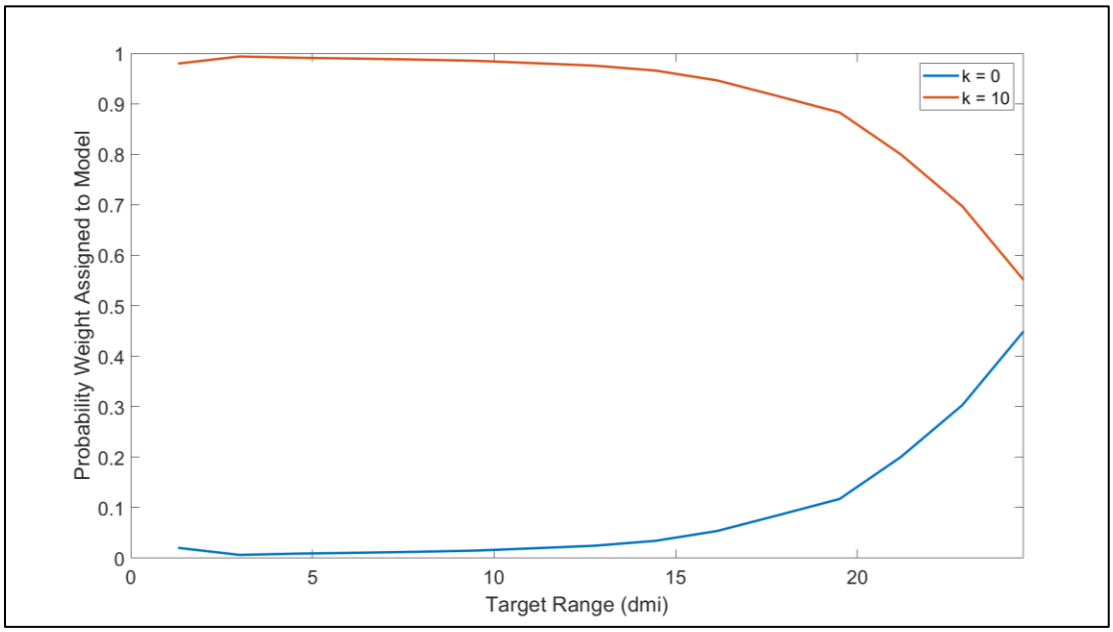

Figure 57: Plot of model probabilities for each k value for Dutch roll trajectory

A performance study with the EPC EKF IMM was conducted for the set of non-NRI targets to evaluate whether the EPC EKF IMM can leverage the EPC model benefits without diverging during the non-NRI portions of the target trajectory. In the next two sections, the EPC IMM is tested against high-diver and dogleg targets.

4.2.2. EPC EKF IMM Performance for the High-diver Target

In the final data miles prior to the high-diver's turn-down maneuver, it appears as though the target will fly over the sensor. For this region, the EPC filters have been shown to fail, since the "corrective" acceleration components are trying to force a non-NRI target back to the radial. As a result, the RMSE performance in some dimensions is

significantly worse for the EPC filter than for both the C-EKF and S-EKF as seen in the previous performance analyses in Sections 2.6 and 3.3.

Against a high-diving target, the EPC IMM starts by weighting its tracking estimates toward the EPC filter (k=10), as shown in Figure 58. At a long range, the motion of the high-diver still resembles an NRI behavior, but as the target gets closer to the sensor origin, it becomes clearer that it will fly over the sensor. Owing to the predicted flyover, the model switches to the k=0 model from RTG 20 dmi to 7.5 dmi. By executing this model change, the algorithm successfully averts the diverging track behavior that the k=10 model will introduce when the target is non-NRI. In addition, the algorithm switches back to the k=10 model shortly after the target executes its diving maneuver. This behavior is preferable because the IMM can take advantage of the smaller errors granted by the EPC model once the target is a fully NRI.

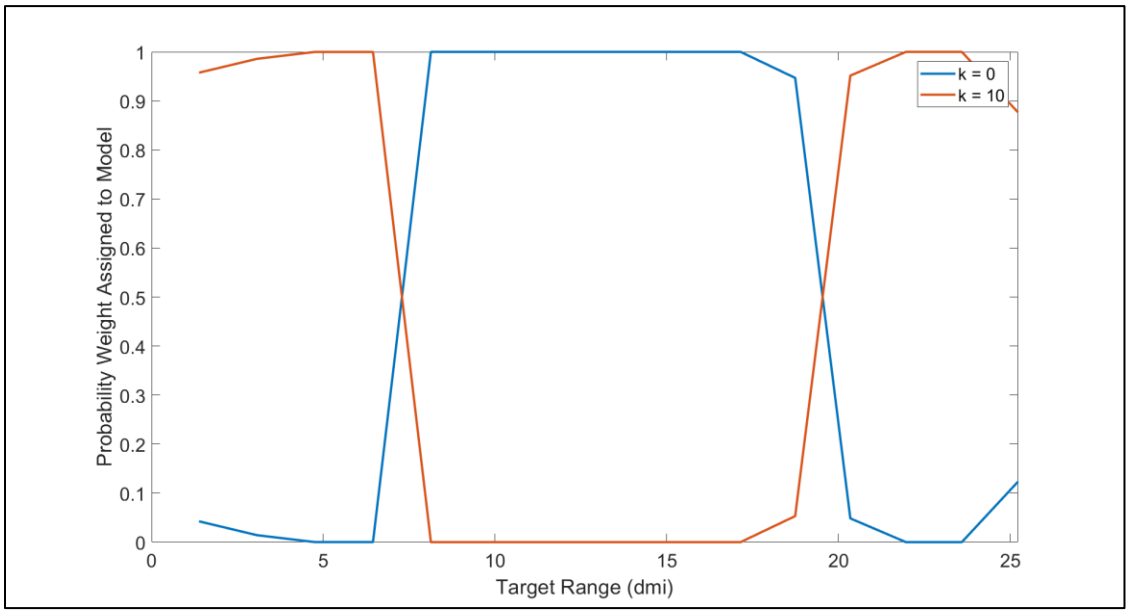

Figure 58: EPC IMM model probabilities for high-diver trajectory

The bearing RMSE performance of the EPC EKF for the high-diver trajectory is superior to that of the S-EKF, as shown in, but inferior to the EPC EKF acting on its own.

The negative tradeoff of this IMM algorithm is that some bearing RMSE performance is lost when the IMM favors the model with k=0 in the middle section of the scenario. However, the elevation RMSE performance of the EPC EKF was poor in the region where the target was not NRI (RTG = 7.5-28 dmi). This poor elevation RMSE performance is not observed in the EPC IMM. Thus, a positive tradeoff is found: the EPC IMM gains significant elevation RMSE performance at the cost of a modest degradation in bearing RMSE, as can be seen in Figure 59.

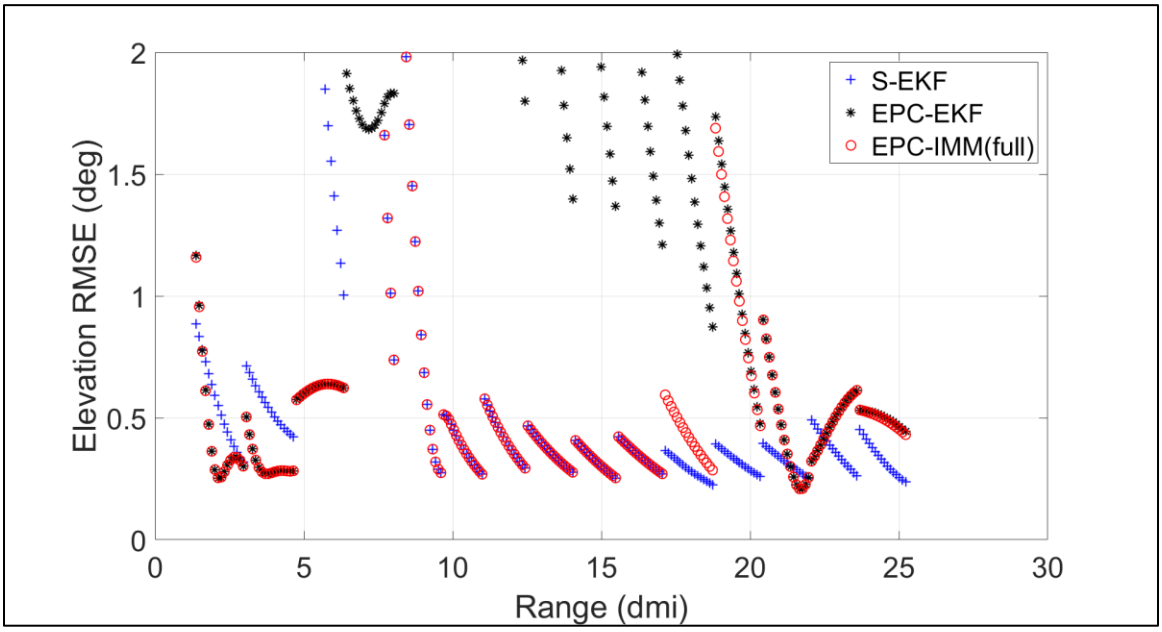

Figure 59: EPC EKF IMM elevation RMSE plot for non-NRI high-diver trajectory

4.2.3. EPC EKF IMM Performance for the Dogleg Target

The dogleg scenario was the second non-NRI target to be examined. In section 2.6.5 it was shown that for the latter portions of the dogleg trajectory, (between 30 dmi and 7.5 dmi RTG), the EPC EKF suffered a significant loss in bearing RMSE performance. The EPC IMM ameliorates this issue by allowing the PN constant k=0 model to be activated when the target's behavior is sufficiently non-NRI. All models start

with equal probability, but the EPC IMM is able to transition to the lower *k* value of approximately 18 dmi RTG as shown in Figure 60.

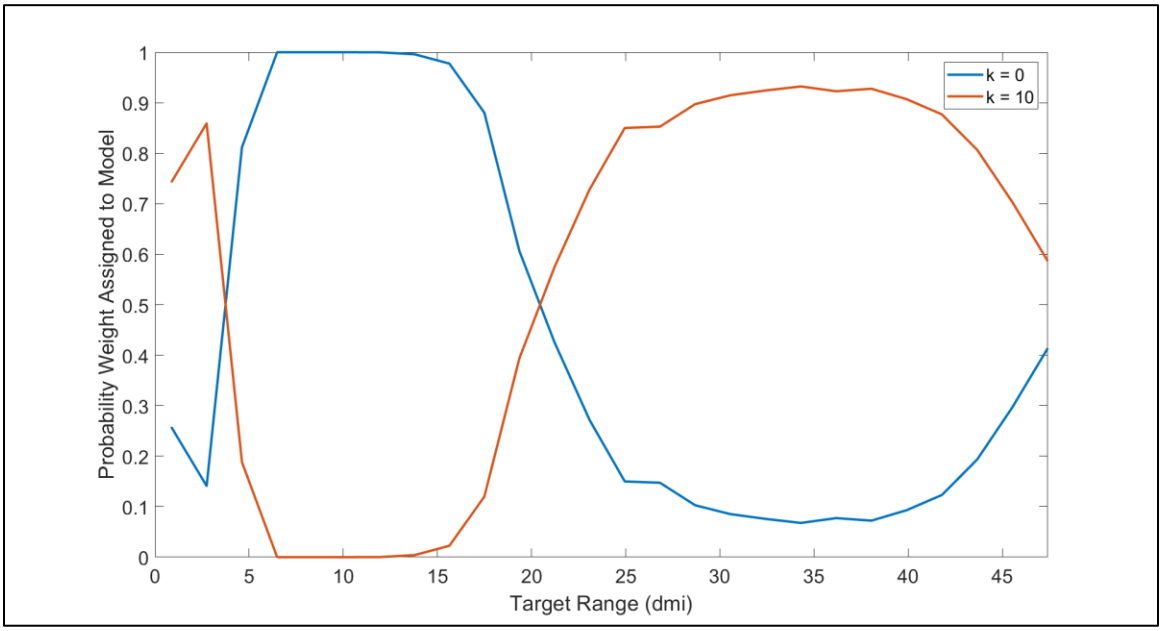

Figure 60: EPC EKF IMM model probabilities for high-diver scenario

The model switching shown in Figure 60 enables the EPC EKF IMM to maintain significantly lower bearing errors. Because this target is NRI for a smaller portion of its trajectory, the gains from the EPC filters are not as easy to observe, but the EPC EKF IMM outperforms the S-EKF at the beginning and end of this scenario. There is a small increase in the bearing RMSE (see Figure 61) in the 10 dmi period prior to the model changeover (30 dmi RTG to 20 dmi RTG). However, this is a reasonable tradeoff in exchange for the significant performance gains in the elevation RMSE shown in Figure 62. Note that for the heavily non-NRI portion of the scenario (20 dmi RTG to 7.5 dmi RTG), the EPC IMM matches the S-EKF's performance. While this is not necessarily a bad result, further error performance gains are possible, as shown in Section 4.2.4.

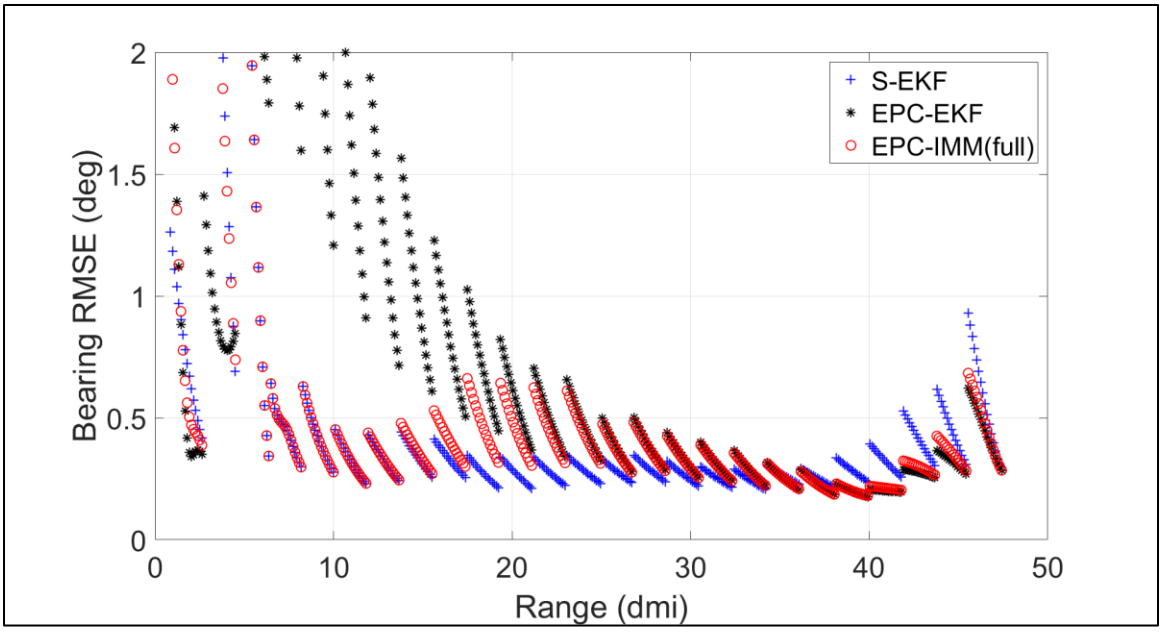

Figure 61: EPC EKF IMM bearing RMSE plot for non-NRI dogleg scenario

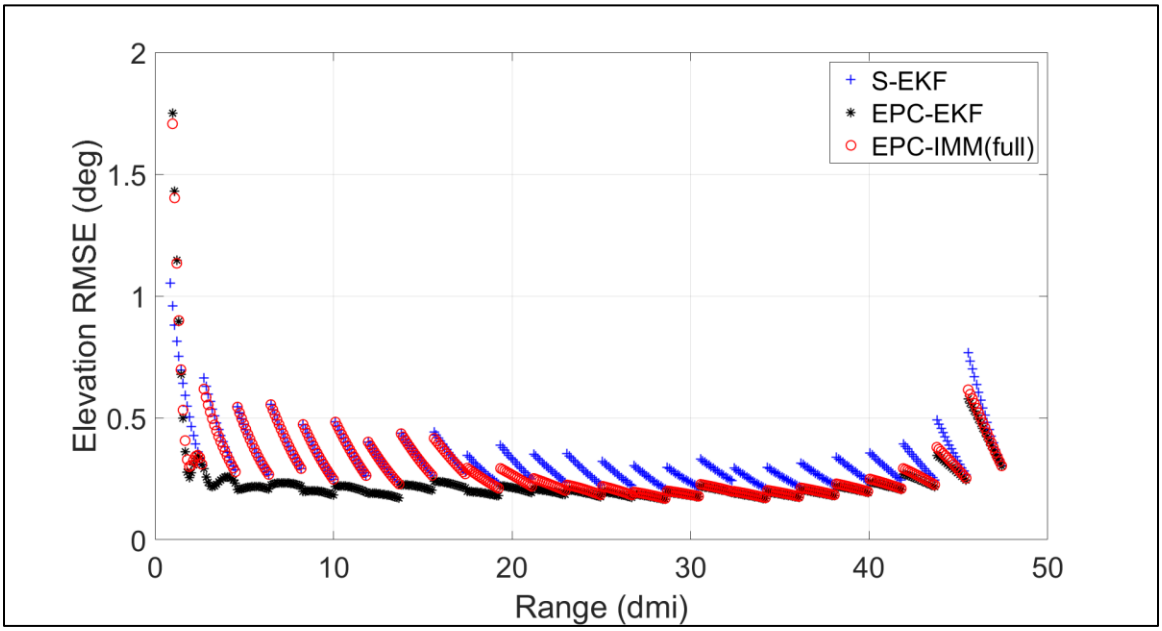

Figure 62: EPC EKF IMM elevation RMSE plot for non-NRI dogleg scenario

4.2.4. A More Flexible EPC IMM

After performing the studies in 4.2.2 and 4.2.3 it was found that it is possible to further tune the EPC models used in the IMM to exploit the fact that targets can be NRI

in some dimensions, but not in others. For example, a high-flying target that passes over the sensor is radially inbound in the bearing dimension, but not in the elevation dimension. A sea-skimming dogleg trajectory may be endpoint-constrained in elevation, but not in bearing, unless it turns toward the sensor origin. For this reason, an improved EPC IMM was developed by breaking up the PN constant k into three subcomponents k_R, k_θ, and k_φ. This necessitates a modification of $\mathbf{u}_{EPC}$ to take the following form:

$$\boldsymbol{u}_{\text{EPC}}(\boldsymbol{x}) = \begin{bmatrix} (1-k_R)R\left(\dot{\theta}^2\cos^2\varphi + \dot{\varphi}^2\right) \\ (2-k_\theta)\dot{\theta}\left(\dot{\varphi}\tan\varphi - \dfrac{\dot{R}}{R}\right) \\ (k_\varphi - 2)\dfrac{\dot{R}\dot{\varphi}}{R} + (k_\varphi - 1)\dot{\theta}^2\sin\varphi\cos\varphi \end{bmatrix}. \tag{4.18}$$

The flexible EPC IMM uses the same IMM algorithm process laid out in Table 13 with two notable changes. The first change is that the EKF steps in Equations (4.6)–(4.10) have been replaced with the EPC UKF algorithm steps. This modification was made in light of the findings from the study conducted in Section 3.3 which was documented in [34]. In those studies, it was shown that the EPC UKF provides superior RMSE performance compared to the EPC EKF, and can deliver that performance in real time. The second modification to the original EPC IMM algorithm is that the number of models is increased from two to four.

Among the four models comprising the flexible EPC IMM, the first sets all three PN constant values to zero, resulting in a spherical filter. The second and third models set all but k_θ and k_φ, respectively, to zero while changing the PN constants to ten. The final model used all k values set to ten, which is the original EPC UKF. The PN constant

values used for the various models are presented in Table 14. The performance results comparing the EPC IMM algorithms are presented in Section 4.3.

Table 14: Flexible IMM PN value table

Model Type:	k_R	k_θ	k_ϕ
Model 1 (No EPC)	0	0	0
Model 2 (EPC in Bearing)	0	10	0
Model 3 (EPC in Elevation)	0	0	10
Model 4 (Full EPC)	10	10	10

The Markov transition probability matrix $\boldsymbol{\pi}$ was modified compared with its debut in Equation (4.1). . The matrix diagonals (such as π_{ii}) represent the probability of remaining in model state *i*, while the other elements in each row represent the probability of "switching" to that state from the *i*th state. For the analysis, the transition matrix is given by

$$\boldsymbol{\pi} = \begin{bmatrix} 0.97 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.97 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.97 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.97 \end{bmatrix}. \tag{4.19}$$

Note that both the EPC EKF and EPC UKF can be used as core filters in the IMM. In the next section, the flexible EPC IMM is compared to the EPC IMM developed in Section 4.2.

4.3. EPC IMM Tracking Algorithm Performance Study

4.3.1. Purpose of Study

The goal of the EPC IMM algorithm development was to address the mis-modeling errors introduced by using the EPC with targets that were not NRI for their full duration. In Sections 4.2.2 and 4.2.3 it was shown that the EPC EKF IMM is capable of

delivering improved results for the non-NRI high-diver and dogleg targets. The EPC EKF IMM succeeded in ameliorating the track divergence caused by the targets' non-NRI behavior, but at a cost. The tradeoff is that while performance losses in the non-NRI dimension were removed (e.g., elevation for the high-diver target), so were performance gains in the NRI dimension. This occurred because only two models were available to the EPC EKF IMM; therefore, if the target was non-NRI in one dimension, all dimensions would lose the benefit of the EPC model when the IMM switched to the S-EKF. The flexible EPC UKF IMM eliminates this problem. To demonstrate this, the flexible EPC UKF IMM's performance was compared to that of the EPC EKF IMM, legacy S-EKF, and EPC EKF to show that it can offer robust tracking performance for both NRI and non-NRI targets.

4.3.2. Radially Inbound Target

For this scenario, both the EPC EKF IMM and flexible EPC UKF IMM converged to the full EPC (k=10 in all dimensions) model within the first few measurement updates, as shown in Figure 63 and Figure 64. At this point, each tracker is highly similar to their base EPC EKF and EPC UKF variants, as demonstrated in Figure 65-Figure 66. The overall track error performance results shown in Figure 67-Figure 68 follow suit, with both IMMs significantly outperforming the legacy S-EKF and largely matching the EPC EKF's results. The flexible EPC UKF IMM is clearly a superior filter for this scenario, as can be expected based on the findings in Section 3.3.

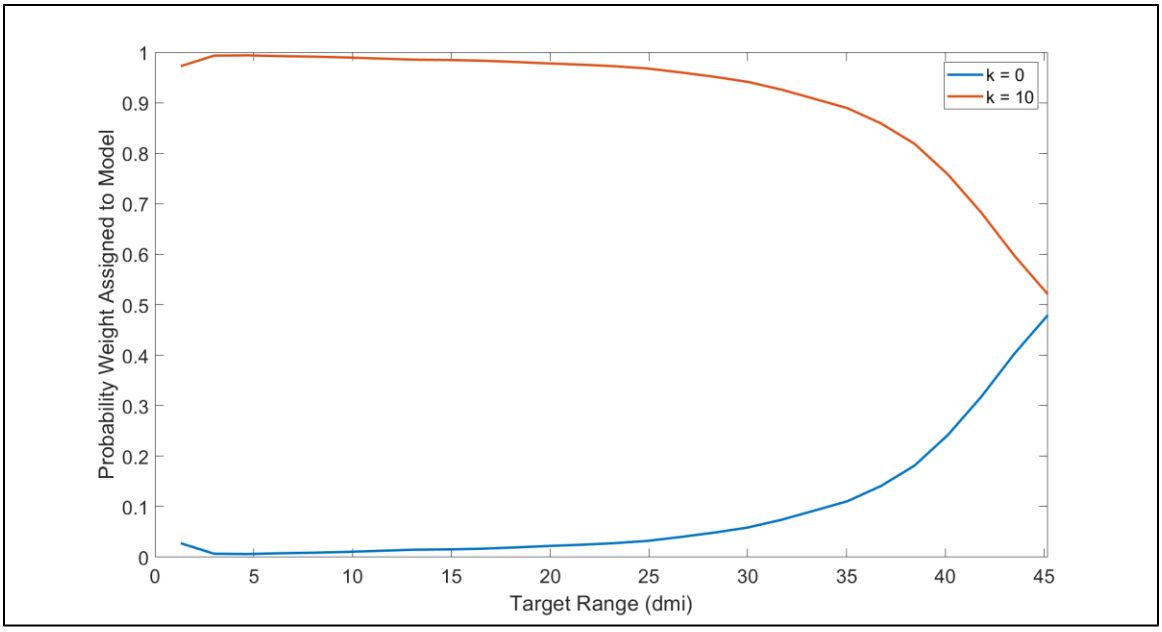

Figure 63: EPC IMM performance study – EPC EKF IMM probability weights for Dutch roll scenario

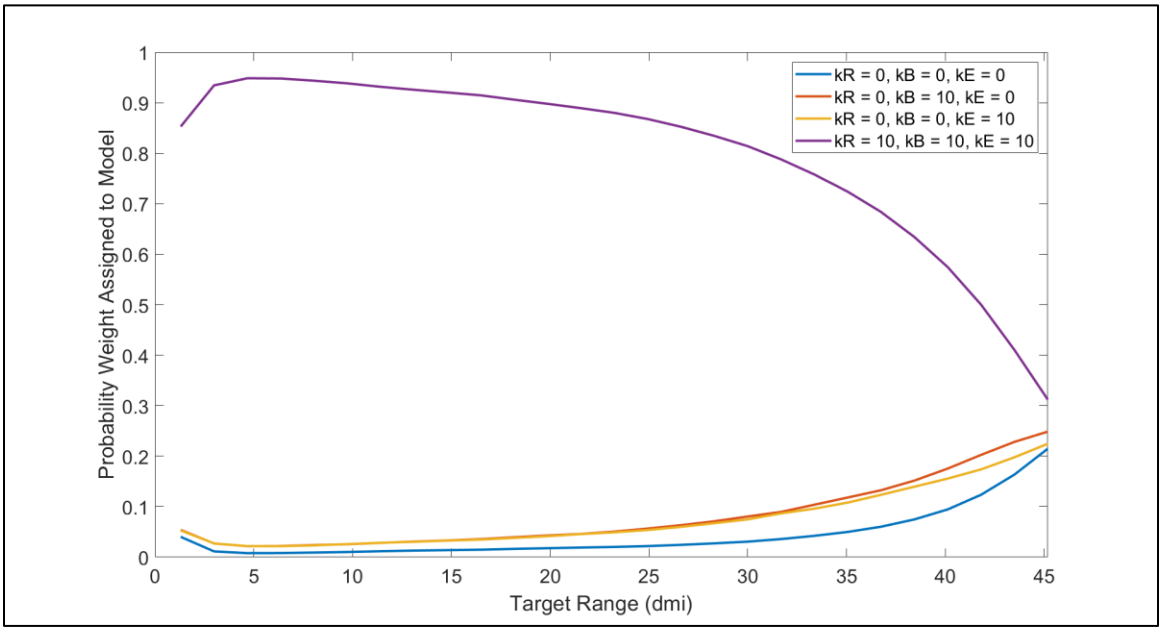

Figure 64: EPC IMM performance study – flexible EPC UKF IMM probability weights for Dutch roll scenario

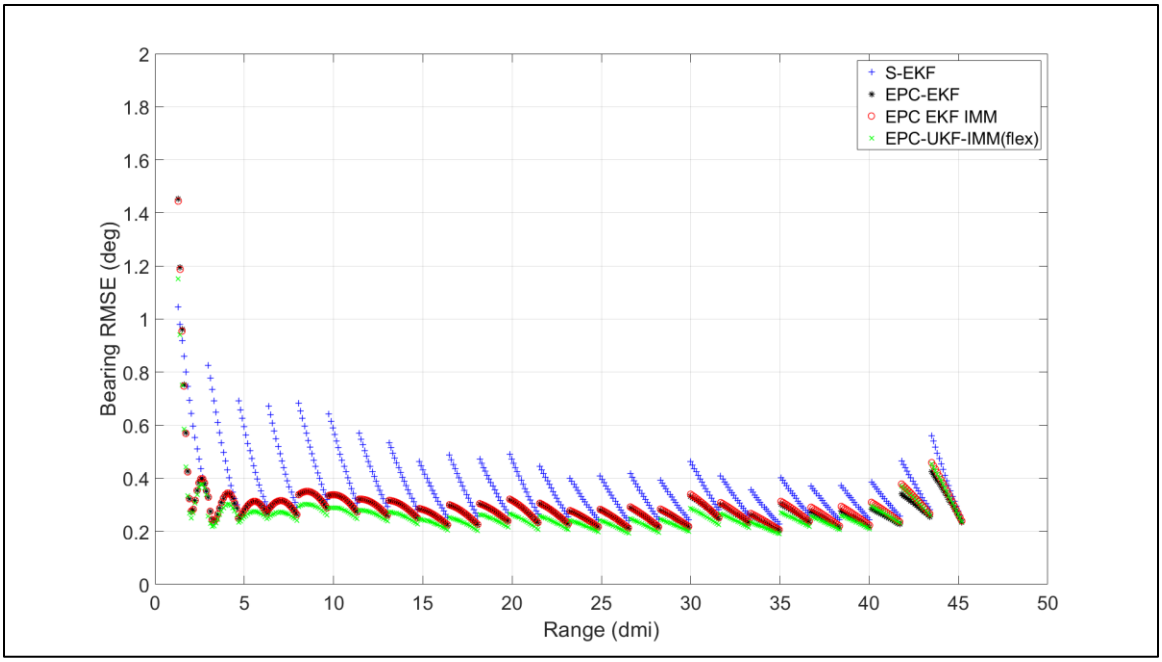

Figure 65: EPC IMM performance study – bearing RMSE for radially inbound scenario

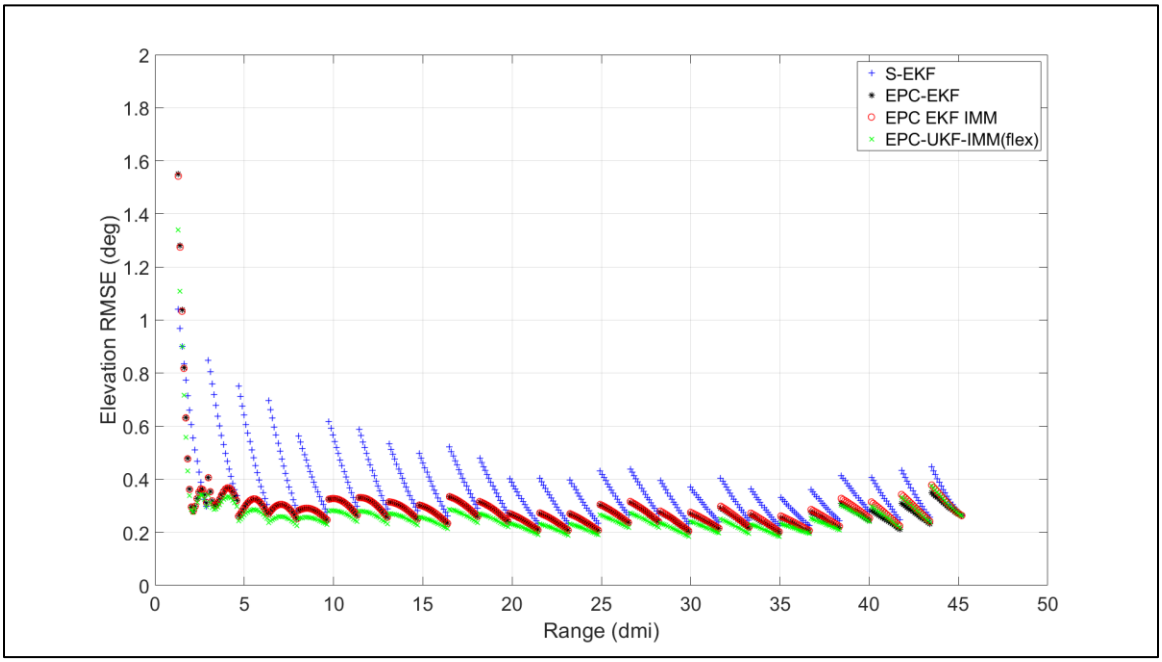

Figure 66: EPC IMM performance study – elevation RMSE for radially inbound scenario

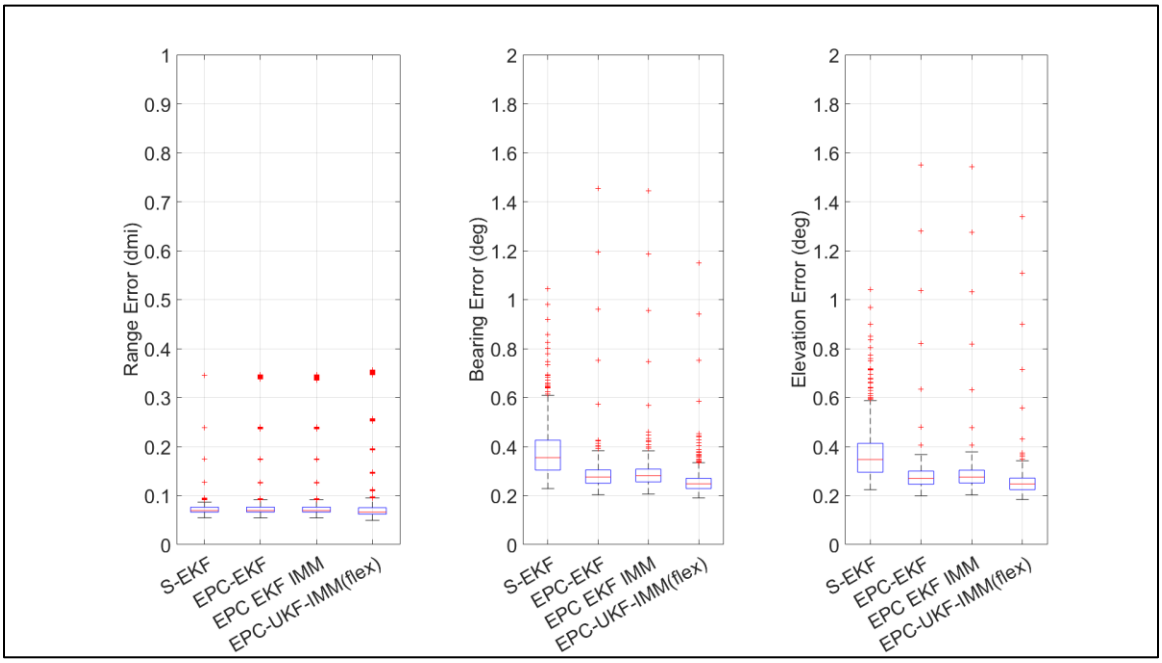

Figure 67: EPC IMM performance study – position error box plots for radially inbound scenario

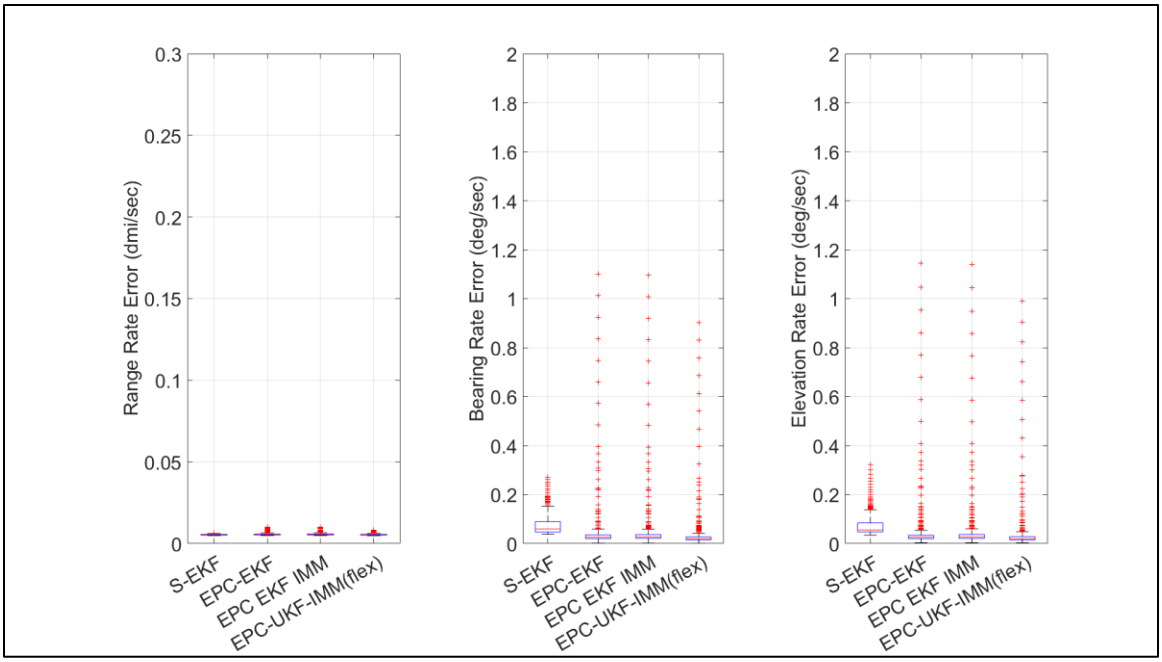

Figure 68: EPC IMM performance study – velocity error box plots for radially inbound scenario

4.3.3. Dutch Roll Target

For the Dutch roll scenario, the IMM error performance results were again similar to their non-IMM counterparts. However, the flexible EPC UKF IMM (Figure 70) algorithm took slightly longer than the EPC EKF IMM (Figure 69) to adjust its models and choose the full EPC model, as shown in Figure 70. Again, both trackers converge to the base EPC EKF and EPC UKF variants. It is clear that both tracking algorithms outperform the legacy S-EKF. Notably, the bearing and elevation RMSE (Figure 71-Figure 72) performance for both IMMs was slightly degraded at the beginning of the scenario. This degradation is due to the fact that the IMMs have not entirely settled into the full EPC model until later in the scenario. The other (less preferable) models still have some influence on the tracking solution until approximately 20 dmi RTG. Both IMMs roughly converged to the EPC EKF solution beyond that point. The contributions of these less preferable models to the overall error performance are minimal but still noticeable in the box plots in Figure 73-Figure 74.

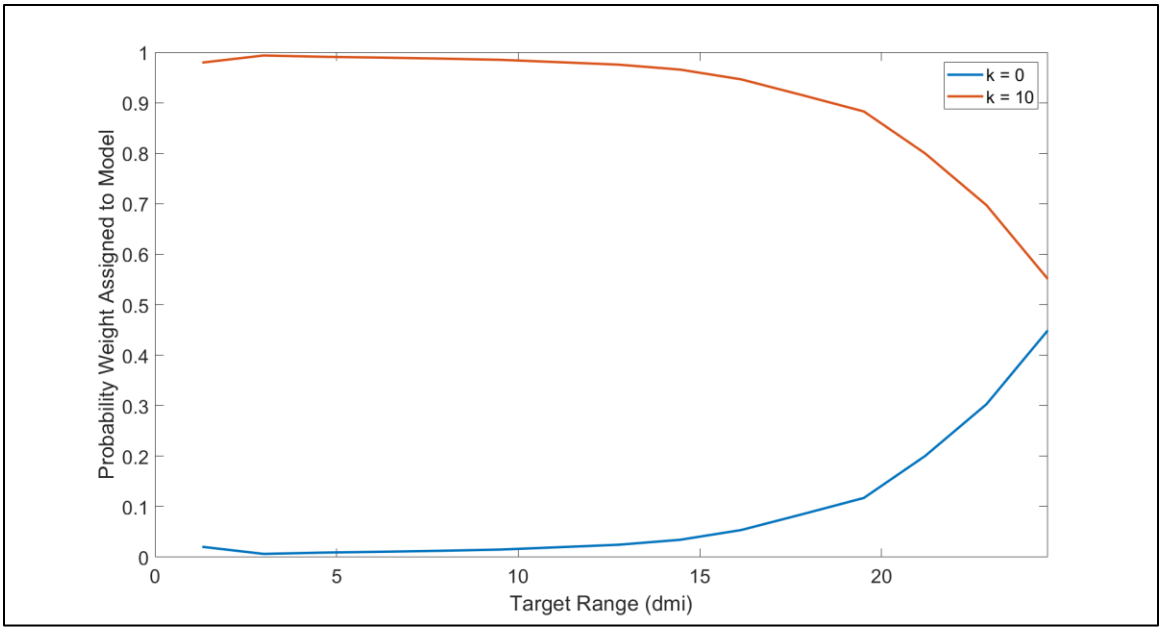

Figure 69: EPC IMM performance study – EPC EKF IMM probability weights for Dutch roll scenario

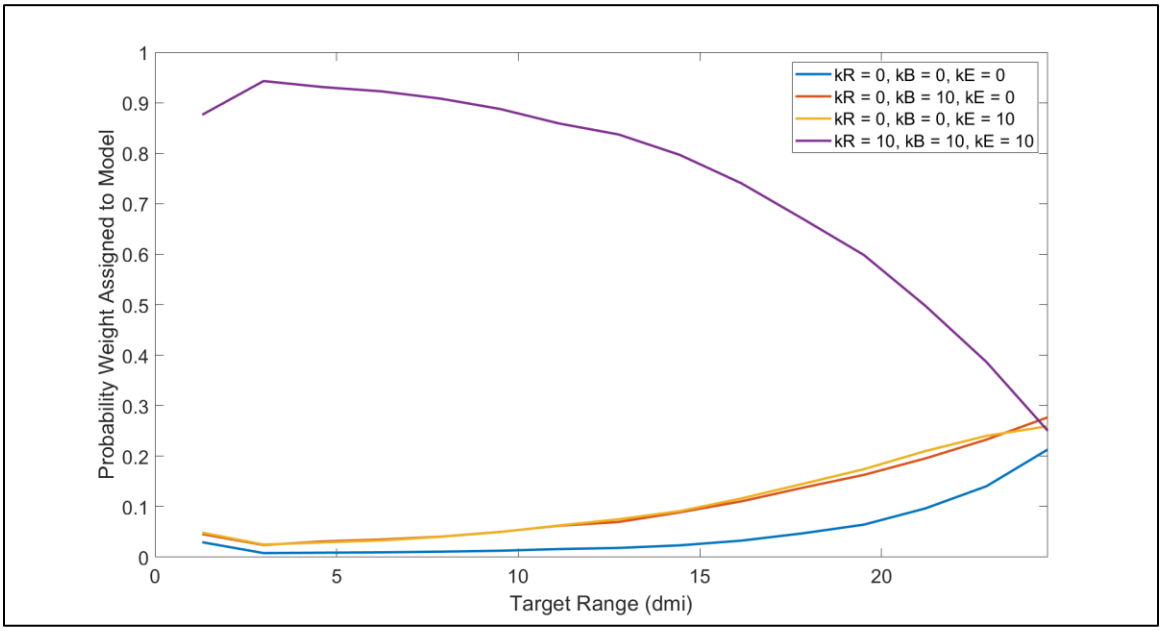

Figure 70: EPC IMM performance study – flexible EPC UKF IMM probability weights for Dutch roll scenario

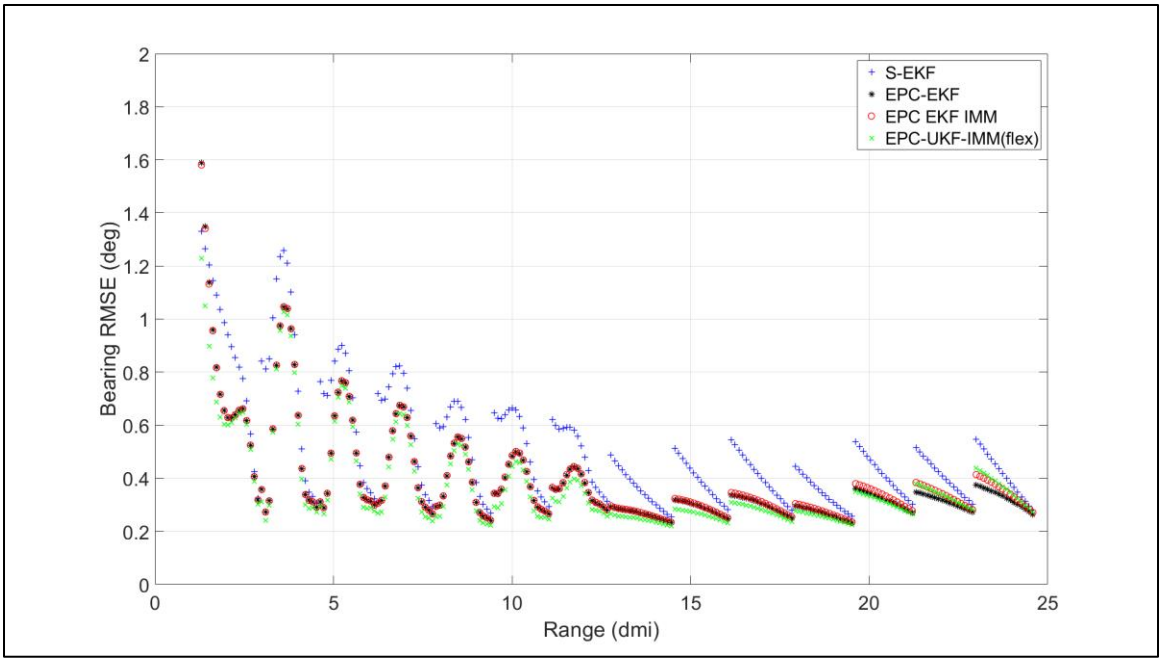

Figure 71: EPC IMM performance study – bearing RMSE for Dutch roll scenario

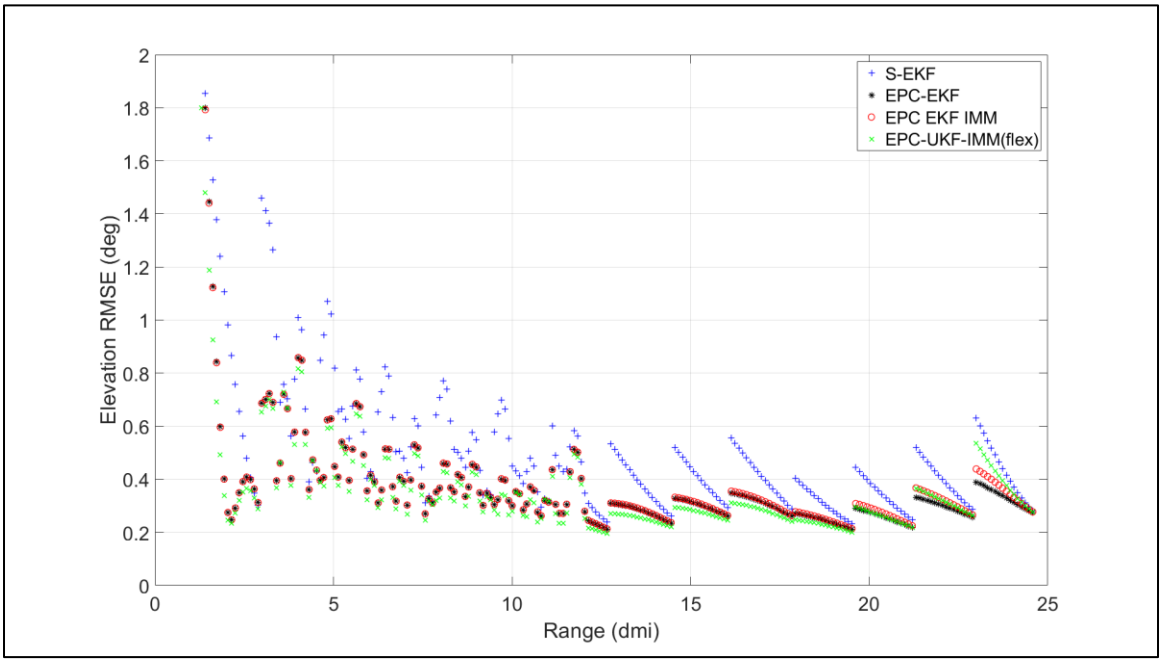

Figure 72: EPC IMM performance study – elevation RMSE for Dutch roll scenario

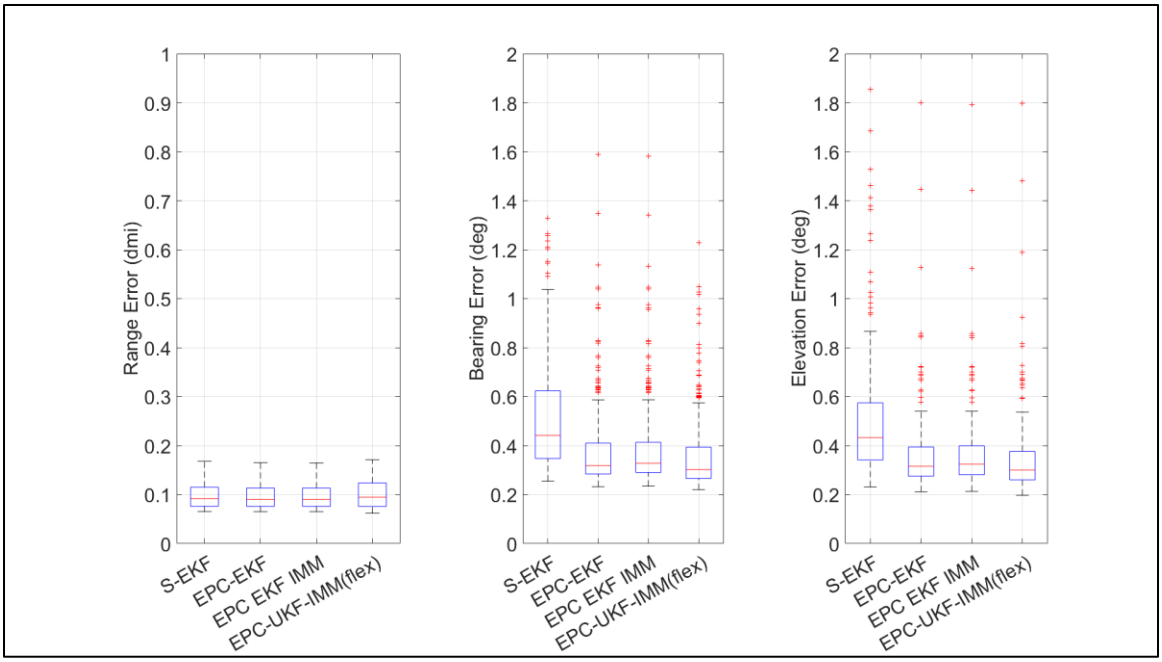

Figure 73: EPC IMM performance study – position error box plots for Dutch roll scenario

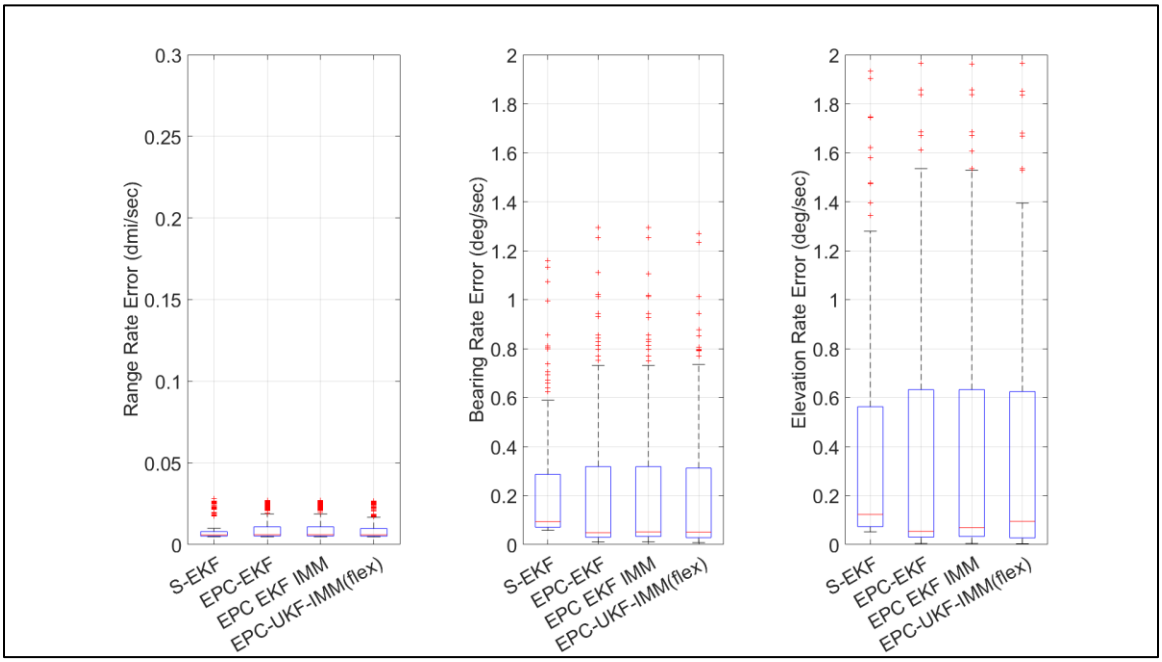

Figure 74: EPC IMM performance study – velocity error box plots for Dutch roll scenario

4.3.4. High-diver Target

The flexible EPC UKF IMM uses a full EPC (k=10 in all dimensions) at the beginning of the scenario when the target geometry is roughly NRI, owing to its long distance from the sensor origin, as shown by the probability weights over time in Figure 75. As the target closes in, the IMM switches to a model in which the EPC is applied only in the bearing dimension and not in elevation. This allows the flexible EPC UKF IMM to reduce bearing and elevation errors while the target is NRI in the horizontal plane but non-NRI in the vertical plane.

The transition between models in the IMMs is not instantaneous, so there is a small error spike for both IMM algorithms when the EPC EKF begins to diverge, but both algorithms are able to quickly adjust their model probability weights and get the errors back down again, as shown in Figure 76-Figure 77. The key benefit of the flexible EPC UKF IMM is its ability to keep the error levels down in all dimensions, regardless of the NRI status of the target. This leads to overall RMSE improvements in the bearing and bearing rate when compared to the EPC EKF IMM, as shown in Figure 78-Figure 79. The small bearing error performance losses compared to the EPC EKF are acceptable owing to the significant gains in the elevation error performance.

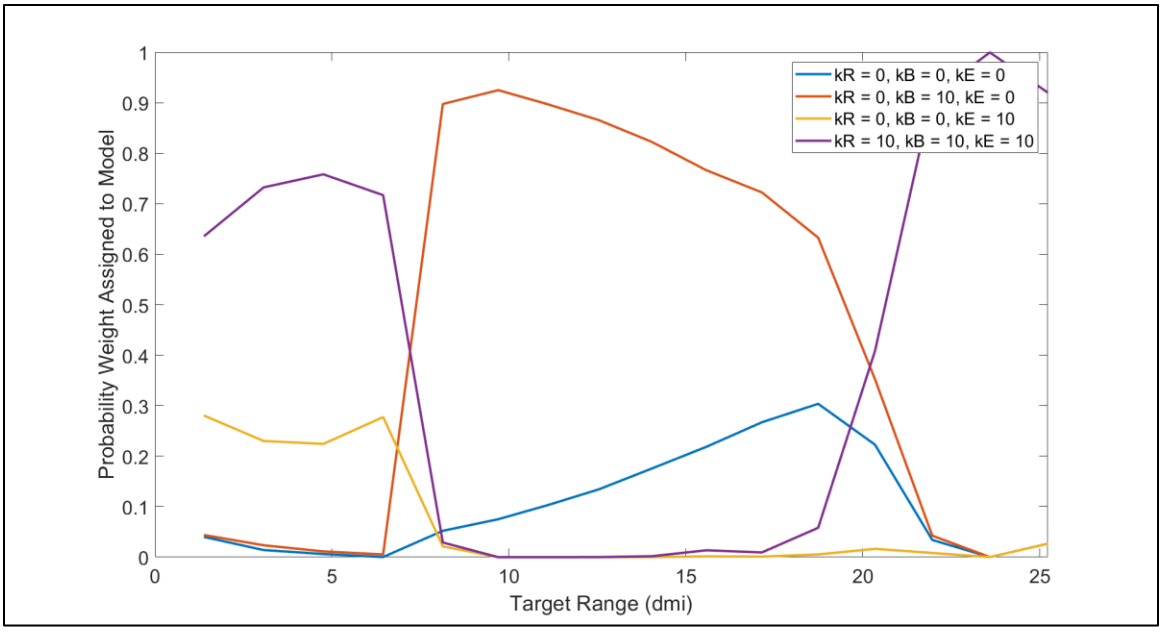

Figure 75: EPC IMM performance study – flexible EPC UKF IMM probability weights for dogleg scenario

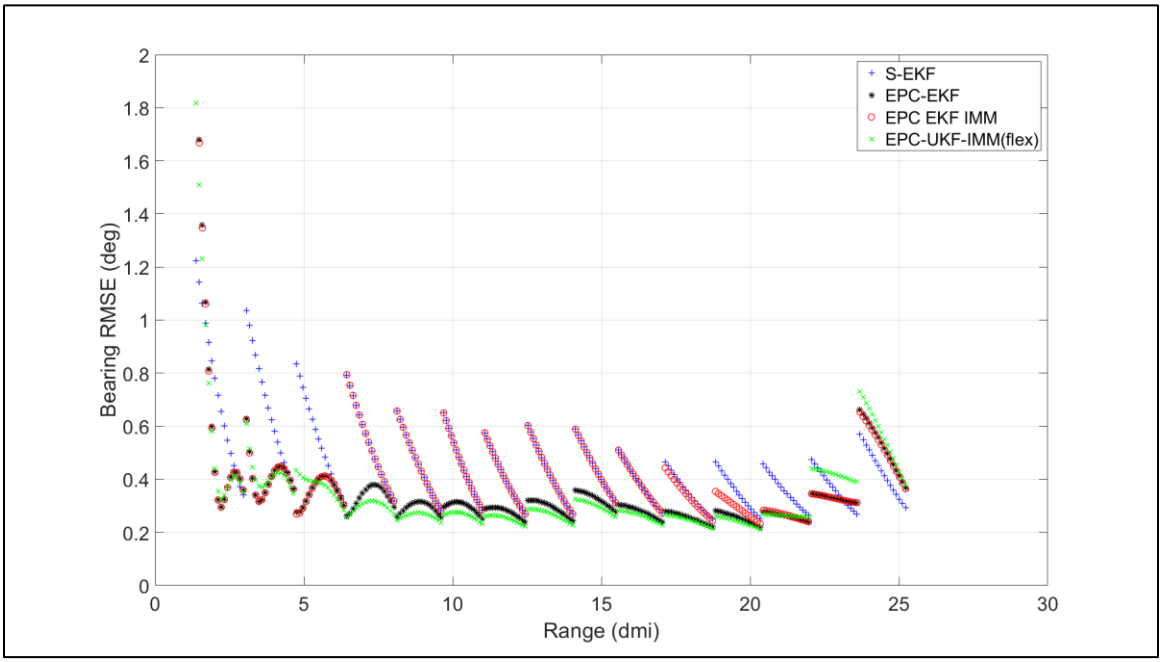

Figure 76: EPC IMM performance study – bearing RMSE for high-diver scenario

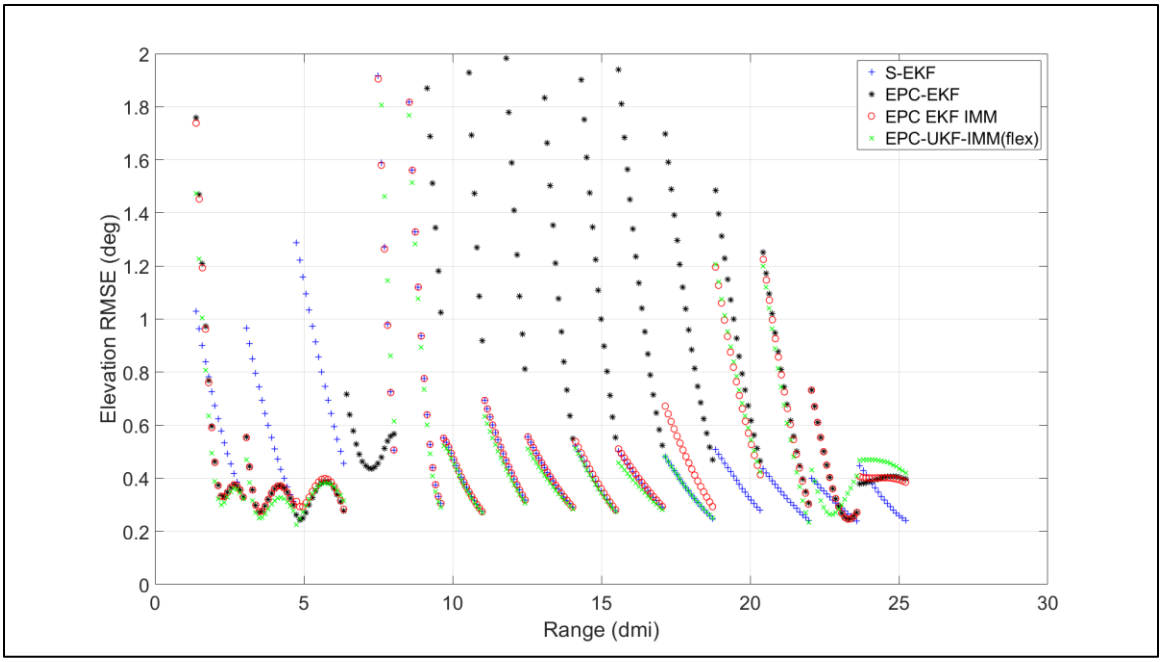

Figure 77: EPC IMM performance study – bearing RMSE for high-diver scenario

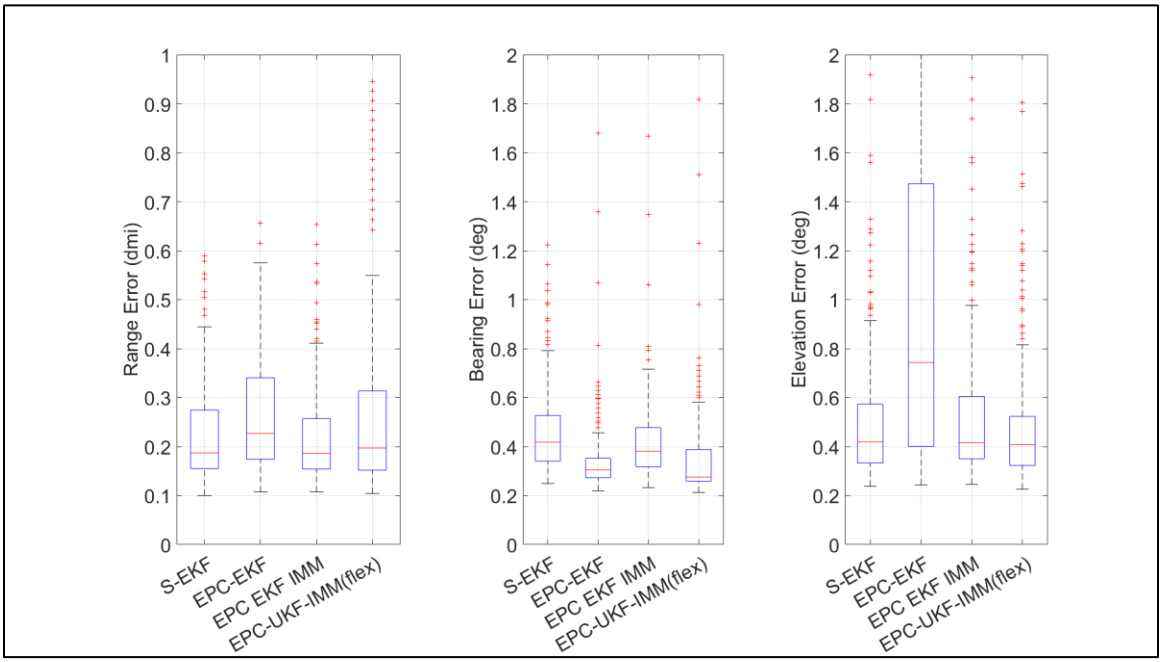

Figure 78: EPC IMM performance study – position error box plots for high-diver scenario

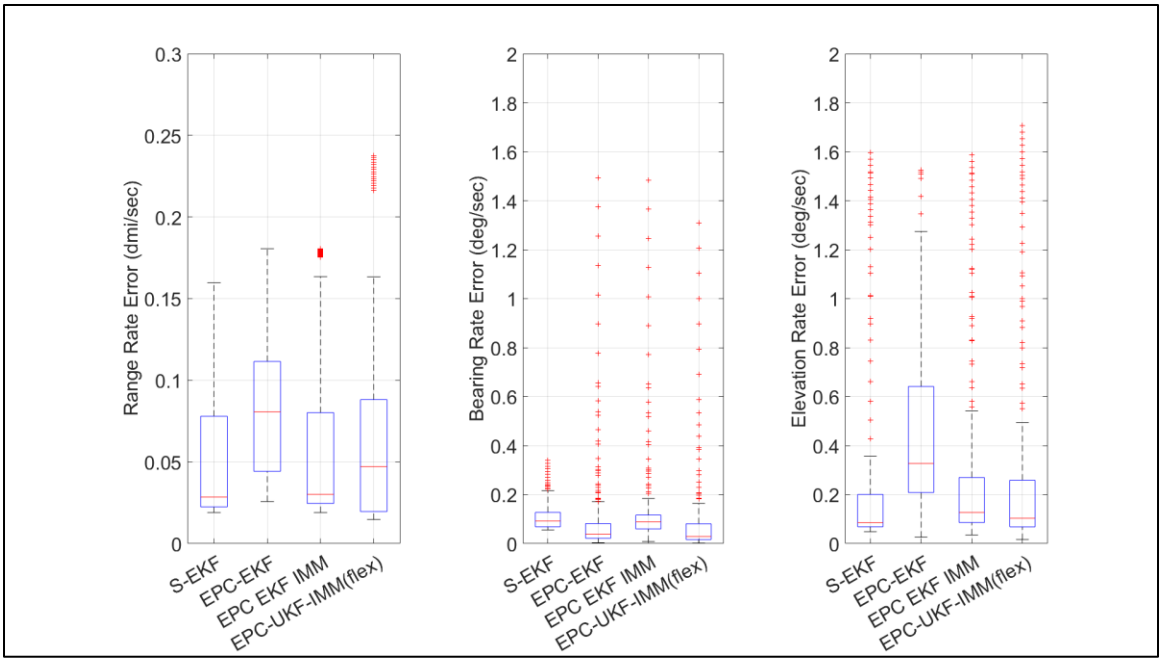

Figure 79: EPC IMM performance study – velocity error box plots for high-diver scenario

4.3.5. Dogleg Target

For the dogleg scenario, the target was NRI in vertical crossrange relative to the sensor, but not in the horizontal crossrange. However, the selection of the optimal model does not occur as quickly as for a high-diver target. While it is true that the dogleg trajectory is NRI in the vertical plane, it travels at a nonzero altitude, which would result in it overshooting the sensor origin. The flexible EPC UKF IMM can independently modify its model probability weights to reach a vertical plane EPC (k_E =10, others set to zero), but it takes longer than for the high-diver. In addition, when the dogleg turn maneuver is executed, there is not much time left to change the models given the proximity of the target to the sensor at the time of the turn. However, Figure 80 shows that the filter tends to favor the full EPC model after the turn, which is the desired result.

The flexible EPC UKF IMM demonstrates superb bearing and elevation RMSE performance in family with the EPC EKF despite the target being non-NRI. Bearing

RMSE performance (Figure 81) is similar for both IMMs, but the flexible EPC UKF IMM performance can achieve a similar elevation RMSE performance (Figure 82) to the EPC EKF, whereas the EPC EKF IMM converges to the S-EKF during the non-NRI portion of the trajectory. During the non-NRI portion of the trajectory, it is clearly preferable to the EPC EKF IMM which is only capable of using the full EPC condition or resorting to using no EPC at all. The box RMSE plots in Figure 83 and Figure 84 emphasize the elevation and elevation rate performance gains for the flexible EPC UKF IMM compared to the EPC EKF IMM.

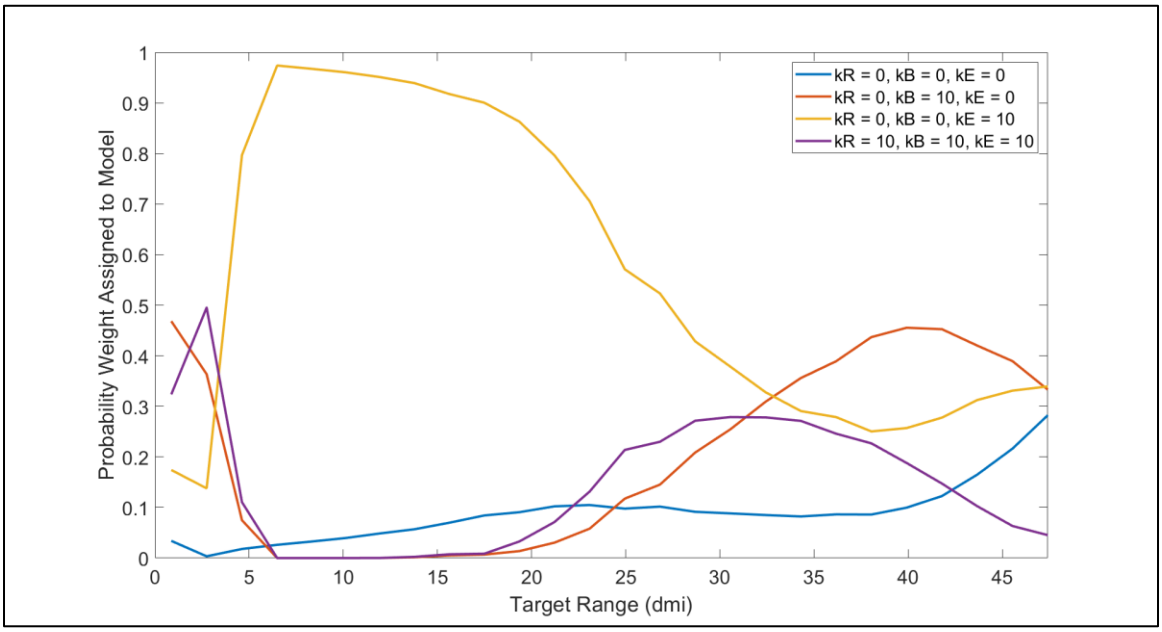

Figure 80: EPC IMM performance study – flexible EPC UKF IMM probability weights for dogleg scenario

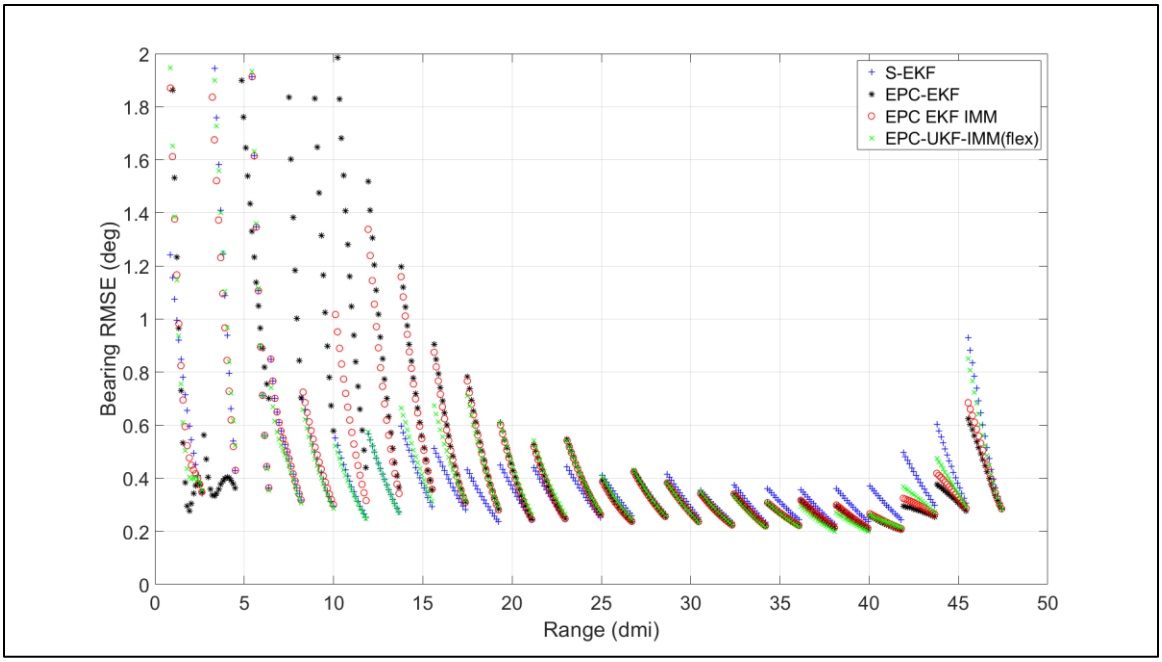

Figure 81: EPC IMM performance study – bearing RMSE for dogleg scenario

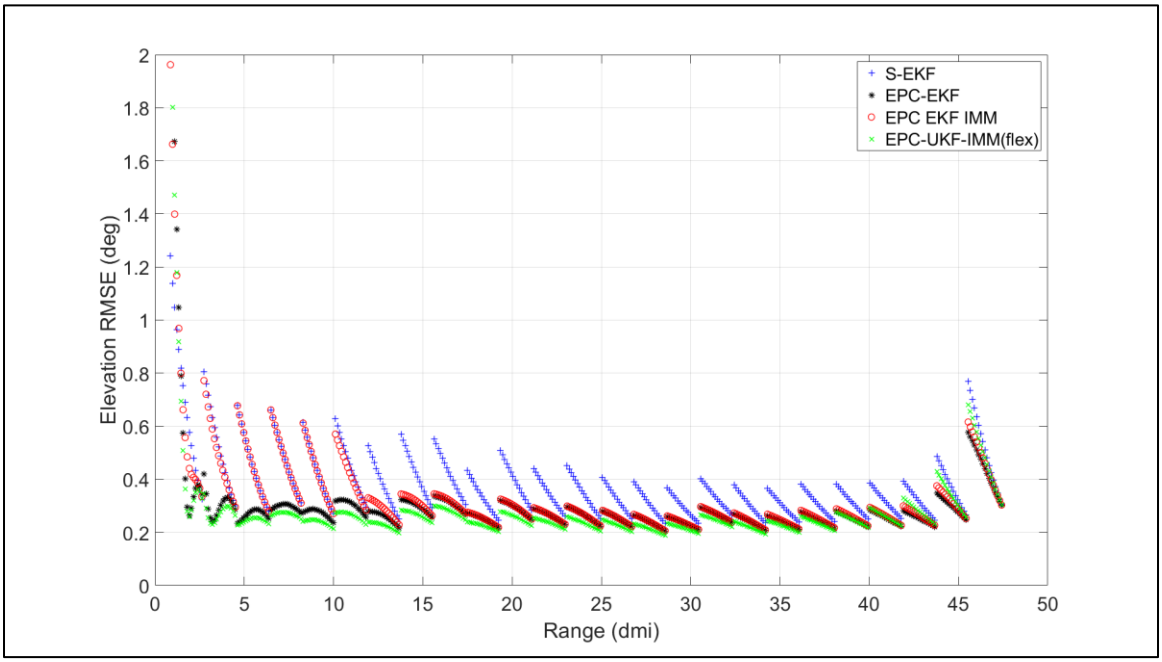

Figure 82: EPC IMM performance study – elevation RMSE for dogleg scenario

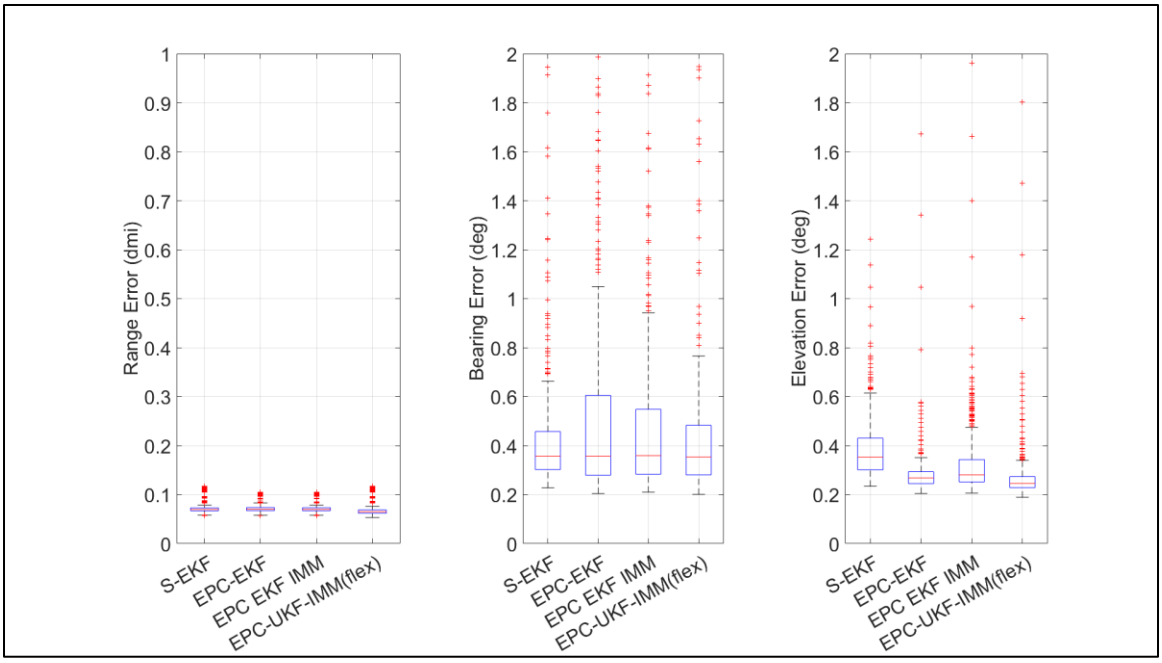

Figure 83: EPC IMM performance study – position error box plots for dogleg scenario

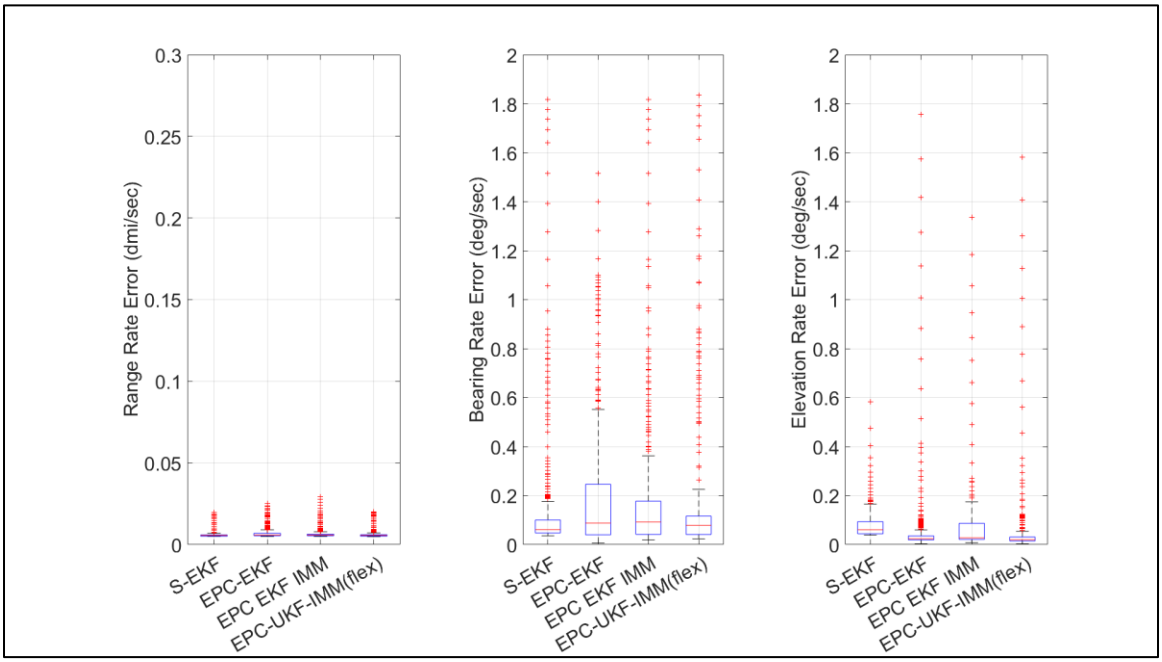

Figure 84: EPC IMM performance study – velocity error box plots for dogleg scenario

4.3.6. EPC IMM Overall Performance Assessment

Choosing a robust filter that leverages the EPC model to achieve performance boosts when compared to legacy filters requires a thorough consideration of the RMSE

performance across all four trajectory types. The flexible EPC UKF IMM was found to offer the best of both worlds, tackling NRI targets with RMSE values nearly as low as the EPC EKF while overcoming the hurdle presented by non-NRI targets. The flexible EPC UKF IMM also removes the performance trade-offs that the EPC EKF IMM allowed. This was made possible by modifying the EPC highlighted in Equation (4.18).

For each trajectory, IMM results are highlighted in green if they were found to be the preferable filter (see Table 15). The EPC EKF IMM was found to offer slightly better performance for NRI targets, while the EPC UKF IMM was found to offer significantly better results than non-NRI targets. Because the target trajectories can never be known a priori, the most prudent filter selection based on this analysis is the flexible EPC UKF IMM. The flexible EPC UKF IMM offers improved performance compared with the legacy S-EKF in all cases, comparable performance against NRI targets, and superior performance against non-NRI targets. Both IMM algorithms were found to be real-time, as shown in Table 16; therefore, the flexible EPC UKF IMM has been shown to be a robust filter worth considering as a replacement for legacy filters.

Table 15: Overall RMSE Performance for EPC IMM Comparison Study

Scenario	Filter Name	RMS Error Overall					
		Range (dmi)	Range Rate (dmi/s)	Bearing (deg)	Bearing Rate (deg/s)	Elevation (deg)	Elevation Rate (deg/s)
Radial Inbound	S-EKF	0.09352	0.07369	0.38450	0.07444	0.37585	0.07383
	EPC-EKF	0.09343	0.07513	0.28745	0.05160	0.28287	0.05184
	EPC-EKF IMM	0.09336	0.07527	0.29247	0.05278	0.28762	0.05302
	EPC UKF IMM	0.09332	0.07525	0.29232	0.05123	0.28828	0.05192
Dutch Roll	S-EKF	0.09797	0.09611	0.51683	0.21300	0.51542	0.38513
	EPC EKF	0.09689	0.09744	0.39311	0.20701	0.37408	0.37431
	EPC EKF IMM	0.09683	0.09754	0.39899	0.20837	0.37990	0.37609
	EPC UKF IMM	0.10086	0.09636	0.37012	0.19646	0.35341	0.36522
High-diver	S-EKF	0.32242	0.24592	0.45894	0.10989	0.66735	0.23737
	EPC EKF	0.35472	0.29001	0.34609	0.10642	1.11607	0.43779
	EPC EKF IMM	0.31738	0.25081	0.42052	0.13544	0.68706	0.29548
	EPC UKF IMM	0.37745	0.25961	0.33835	0.08910	0.67777	0.28033
Dogleg	S-EKF	0.07199	0.07758	0.50161	0.15345	0.38808	0.07982
	EPC EKF	0.07196	0.08432	0.59988	0.20765	0.29580	0.06454
	EPC EKF IMM	0.07158	0.08266	0.52671	0.18539	0.33353	0.06983
	EPC UKF IMM	0.06807	0.07812	0.49902	0.16882	0.27733	0.05682

Table 16: IMM algorithm comparison study: runtime performance table

Scenario	Filter Name	Scenario Duration (sec)	Average Runtime (sec)	Runtime Ratio	Real Time?
Radial Inbound	S-EKF	117	0.00162	0.0014%	Yes
	EPC-EKF	117	0.00155	0.0013%	Yes
	EPC EKF IMM	117	0.00657	0.0056%	Yes
	Flexible EPC UKF IMM	117	0.03905	0.0334%	Yes
Dutch Roll	S-EKF	69	0.00092	0.0013%	Yes
	EPC-EKF	69	0.00091	0.0013%	Yes
	EPC EKF IMM	69	0.00368	0.0053%	Yes
	Flexible EPC UKF IMM	69	0.01967	0.0285%	Yes
High-diver	S-EKF	73	0.00089	0.0012%	Yes
	EPC-EKF	73	0.00087	0.0012%	Yes
	EPC EKF IMM	73	0.00358	0.0049%	Yes
	Flexible EPC UKF IMM	73	0.02022	0.0277%	Yes
Dogleg	S-EKF	108	0.00141	0.0013%	Yes
	EPC-EKF	108	0.00140	0.0013%	Yes
	EPC-EKF IMM	108	0.00591	0.0055%	Yes
	Flexible EPC UKF IMM	108	0.03351	0.0310%	Yes

5. Continuous-Discrete Endpoint-Constrained Filters

5.1. Background

All of the D-D filtering algorithms examined in Chapters 2 and 3 are specific cases of the generalized nonlinear filtering problem and can be tied back to the Zakai equation. Elliot and Haykin [32] provided a concise derivation of the D-D Kalman filter from the Zakai equation. These filtering algorithms also rely on assumptions regarding the underlying probability distribution of the target state and assume that the expected value and covariance of the state are sufficient to fully describe the system at a given time.

Algorithms such as the EKF take this a step further by linearizing the dynamic and observation models about the estimate of the target state and covariance [15][43]. These assumptions imply that the family of Kalman filters is not universally optimal for estimating non-Gaussian stochastic systems [9]. This may have already been apparent after the discussion in Section 2.2.3 where it was shown that the nonlinearity of the observation model renders the Gaussian assumption for the observation predictions invalid under some conditions. The algorithms presented in this chapter do not rely on the same discretization, probability distribution, or linearization assumptions.

5.1.1. The Zakai Equation

For the generalized filtering problem, the stochastic differential equation governing the evolution of the target state is given by:

$$d\boldsymbol{x} = \mathbf{f}\left(\boldsymbol{x},t\right)dt + \mathbf{G}\left(\boldsymbol{x},t\right)d\boldsymbol{v}. \tag{5.1}$$

The function $\mathbf{f}(\boldsymbol{x},t)$ is the state model drift and represents the modeled motion behavior of state $\boldsymbol{x}$ at time t, and the function $\mathbf{G}(\boldsymbol{x},t)$ is a state-dependent modifier of the variance for the dynamic noise vector $\boldsymbol{v}$, which has zero mean and covariance $\mathbf{Q}(t)$. In this context, dynamic noise is represented by Brownian motion process.

The continuous-time observation model is given by the following stochastic differential equation

$$d\boldsymbol{z} = \mathbf{h}\left(\boldsymbol{x},t\right)dt + d\boldsymbol{w}. \tag{5.2}$$

The function $\mathbf{h}(\boldsymbol{x},t)$ is the observation model drift and represents the relation between the measurements and state ($\boldsymbol{x}$), just as its discrete-time counterpart when introduced in Chapter 2. The noise vector $\boldsymbol{w}$ is also treated as a Brownian motion process, but is independent of $\boldsymbol{v}$.

The Zakai equation [124] is the stochastic partial differential equation for the unnormalized probability density of the target state, $\sigma(\boldsymbol{x},t)$, given by

$$d\sigma\left(\boldsymbol{x},t\right) = \mathrm{L}\left[\sigma\left(\boldsymbol{x},t\right)\right]dt + \sigma(\boldsymbol{x},t)\mathbf{h}(\boldsymbol{x},t)\boldsymbol{S}^{-1}d\boldsymbol{z}. \tag{5.3}$$

The operator L is defined as

$$L\left(p\right) = \sum_{i,j}^{N_x} \frac{\partial^2\left[\left(\boldsymbol{G}\mathbf{Q}\left(t\right)\boldsymbol{G}^T\right)_{ij}\, p\right]}{\partial x_i \partial x_j} - \sum_{i=1}^{N_x} \frac{\partial^2\left[\boldsymbol{f}_i p\right]}{\partial x_i}, \tag{5.4}$$

where N_x is the number of elements in the state vector $\boldsymbol{x}$ and $\boldsymbol{G}\mathbf{Q}(\mathrm{t})\boldsymbol{G}^\mathrm{T}$ is the effective diffusion matrix for the stochastic process describing the target dynamics.

Defining the stochastic differential equation (5.3) can be relatively simple if the dynamic and observation models are well understood, but solving it in a practical and real-time manner presents a meaningful challenge [75]. Hence, many filter designers have turned to discretization, probability distribution assumptions, and linearization to

simplify the problem. This is necessary to support real-time solutions to practical problems such as target tracking. For many highly nonlinear filtering problems these approaches are satisfactory. For highly nonlinear and non-Gaussian problems, methods such as the family of particle filters can be used. Alternatively, authors such as Yau [116], Yueh [123], and Balaji [11] recently published direct solution methods to estimate the probability density of the state for some classes of filtering problems. Their work was built on significant advancements in the field from the 1960s to the 1980s by Baras and Blankenship [13], Duncan [31], and Mortensen [87].

5.1.2. The Robust Duncan-Mortensen-Zakai Equation

The Zakai equation is also referred to as the Duncan-Mortensen-Zakai (DMZ) equation, as Duncan [31], Mortensen [87], and Zakai [124] independently reached the same conclusion regarding Equation (5.3) for the nonlinear filtering problem in the 1960s. In the 1980s, Baras et al. published multiple works including [13][14] which leveraged the transformation described by Davis and Marcus [28], allowing for the stochastic PDE in (5.3) to be treated as an ordinary parabolic PDE. This transformation is executed by making the following substitution for the unnormalized density:

$$\rho(\boldsymbol{x},t) = \exp\left[-\mathbf{h}(\boldsymbol{x})\boldsymbol{z}(t)\right]\sigma(\boldsymbol{x},t). \tag{5.5}$$

For a unique path given by a set of observations $\boldsymbol{z}(t)$, Baras et al. [12] demonstrated the existence and uniqueness of solutions to the so-called “robust” DMZ equation. Luo and Yau [74] succinctly described a robust DMZ PDE as:

$$\frac{\partial}{\partial t}\rho(\boldsymbol{x},t) = -\frac{\partial}{\partial t}\left(h^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{S}^{-1}\right)^{\mathrm{T}}\mathbf{z}(t)\rho(\boldsymbol{x},t) + \exp\left(-h^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{S}^{-1}\mathbf{z}(t)\right)\left[L - h^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{S}^{-1}h(\boldsymbol{x})\right]\times \exp\left(h^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{S}^{-1}\mathbf{z}(t)\right). \quad (5.6)$$

This definition of the PDE includes a correction factor found by Wong and Zakai [114] which ensures that the integral of the ordinary PDE converges with the integral of the stochastic PDE it was derived from.

The robust DMZ equation is the focus of a host of modern nonlinear filtering algorithm developments. The PDE governing the evolution of the unnormalized conditional probability $\rho(\boldsymbol{x},t)$ can be solved using various methods. Among these algorithms, one of the most popular is the family of Yau filters for solving continuous nonlinear filtering problems. In the next section, the C-C Yau filter is introduced as a precursor to the continuous-discrete (C-D) approximately optimal Bayesian filters examined in this dissertation.

5.1.3. The Yau Filter

The main goal of the nonlinear filtering problem is to solve the DMZ equation in real time in a memory-less manner. Kalman and Bucy [58][59] developed a filter that still sees myriad applications and modifications, including most of the filters presented in this dissertation. The EKF and UKF algorithms, among many others, were developed to mitigate the Kalman filter's weaknesses in the presence of nonlinear systems. While these algorithms can handle weak nonlinearities, they can still fail when applied to highly nonlinear problems, such as 1D cubic sensors [73].

Particle filters (see Section 3.2) have been popular in recent publications because of their ability to handle non-Gaussian probability distributions subject to nonlinear dynamic and observation models [7]. However, while it is true that the PF becomes asymptotically optimal as more particles are used, it can also result in large computational costs, as seen in our own application [34], as well as in Luo's survey [75]. The increased computational burden of the PF can result in an algorithm that is not real-time, depending on the filter's implementation and available hardware. It is also difficult to determine the number of particles to be used. The Yau filter was developed as an alternative to particle filters and other prior nonlinear filtering algorithms, directly solving the robust DMZ PDE rather than using linearization or Monte Carlo sampling techniques.

According to Balaji [11], the Yau filter is defined as one in which the state model drift $\mathbf{f}(\boldsymbol{x},t)$ is linear plus a gradient term satisfying the Benes condition and with a time- and state-independent diffusion matrix [117]. In continuous-continuous (C-C) filtering theory, the Yau filtering system is defined as one with an additional restriction that the measurement process is a continuous-time stochastic process with linear drift. Yau and his collaborators have shown that a finite-dimensional filter of maximal rank is necessarily a Yau filter [117][118].

In light of its importance, some real-time implementable solutions for Yau filtering systems have been investigated. In [10][119], the solution of the Yau equation was shown to be equivalent to the solution of a system of linear ODEs. A variety of authors have attempted to demonstrate the increased speed and efficiency of executing the Yau filter. Yueh examined and compared a series of time discretization methods

[123] and found that further speed improvements are possible by implementing fast Fourier transformations as part of the algorithm. In addition, it was shown in [121] that some nonlinear observation models can also be solved via ODEs, and in [120], the brothers Yau documented a solution in terms of a power series.

While developing a family of EPC nonlinear target tracking filters, it was found that the direct application of the C-C Yau filter to the EPC problem is not feasible. This was due to several limitations of the Yau filter, as well as limitations of the combat system. The CV spherical target tracking model is necessarily nonlinear with state-dependent diffusion, which disqualifies the Yau filter, and the diffusion matrix is state-dependent. In addition, the C-C filter architecture was found to be unstable when observations were delivered to the system with large update periods. The combat system also does not have access to the underlying continuous-time data which is used to generate the observation data. Balaji [9][10] showed that it is possible to use the principles of the Yau filter for a continuous state and discrete observation model. This continuous-discrete (C-D) filter architecture is the focus of the remaining members of the family of EPC filters explored in this dissertation.

5.2. Continuous-Discrete Filtering

5.2.1. Optimal Bayesian Filtering

Continuous time filters, such as those described in the previous sections, are best suited to problems in which both the state and observation models can be treated as continuous functions of time. For the radar target tracking problem, this approach is impractical, especially for slow rotating radars such as AN/SPS-48 [113], because its

measurements are reported once every four seconds. Such a system lends itself to a discrete-time observational model. However, the equations of motion used to model the motion of the tracked target lend themselves to treatment as continuous.

The continuous-time state model for the general C-D filter is the Langevin equation (5.1) introduced in Section 5.1.1. For the target tracking problem, the state vector **x**(t) represents the current position and velocity information for the object being tracked, whereas the state model drift **f**(***x***,t) describes the physics assumed to govern the motion of the object. The noise **v**(t) is a 3-by-3 matrix vector representing the additive acceleration noise. The noise process ***v***(t) is assumed to be Brownian with covariance **Q**(t). The diffusion modifier **G**(***x***,t) is a 6-by-3 matrix which modifies covariance **Q**(t) based on the target state. The effective diffusion matrix for the system is the composition of two matrices, $\boldsymbol{GQG}^{\mathrm{T}}$. The diffusion controls the rate at which the conditional probability decays or spreads in the absence of new observation data.

The measurement model for the radar is described by the discrete time process

$$\boldsymbol{z}(t_n) = \boldsymbol{h}\left(\boldsymbol{x}(t_n), t_n\right) + \boldsymbol{w}(t_n). \tag{5.7}$$

This is the same measurement model used in all previous D-D EPC filters discussed in this dissertation. See Section 2.2.1 for more details regarding the observation model **h**(***x***) and the covariance of the observation noise ***R***. The PDF of interest, $\mathrm{p}\left(\mathbf{x}(t) \mid \boldsymbol{z}_{t_0:t_n}\right)$, represents the distribution of the target state conditioned on all measurement data accumulated from time t_0 up to t_n.

The universal continuous-discrete filtering problem can be solved by propagating the previous PDF update $\hat{\mathrm{p}}\left(\mathbf{x}(t_{k-1}) \mid \boldsymbol{z}_{k-1}\right)$ forward in time to the most recent recorded

observation to obtain $p_{\text{pred}}\left(\mathbf{x}(t_k)\,|\,\boldsymbol{z}_{k-1}\right)$. These predictions for the PDF are facilitated by solving the Fokker-Planck-Kolmogorov forward equation (FPKfe) [94], given by

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{6}\left[f_i \frac{\partial p}{\partial x_i} + p\frac{\partial f_i}{\partial x_i}\right] + \sum_{i,j}^{6}\frac{\partial^2\left[\left(\boldsymbol{GQG}'\right)_{ij}\, p\right]}{\partial x_i \partial x_j}. \tag{5.8}$$

The prior PDF estimate for the state given all past observations,

$$p_0(\boldsymbol{x}) = \hat{p}\left(\boldsymbol{x}(t_{n-1})\,|\,\boldsymbol{z}_{n-1}\right), \tag{5.9}$$

is considered as the initial condition of (5.8) for each filter update.

The propagated (or predicted) PDF can then be updated using the latest measurement data [11]. The measurement data are represented by the conditional probability of observation, given the current state. In most radar target tracking applications, including those used in our studies, the conditional observation density is assumed to be

$$\begin{aligned} p_{\text{obs}}\left(\boldsymbol{z}_n\,|\,\boldsymbol{x}\right) \sim{}& \frac{1}{\left[\left(2\pi\right)^3 \det\left(\boldsymbol{R}\right)\right]^{1/2}} \times \\ & \exp\left\{-\frac{1}{2}\left(\left[\boldsymbol{z}_n - \boldsymbol{h}\left(\boldsymbol{x}(t_n)\right)\right]^T \boldsymbol{R}^{-1}\left[\boldsymbol{z}_n - \boldsymbol{h}\left(\boldsymbol{x}(t_n)\right)\right]\right)\right\}. \end{aligned} \tag{5.10}$$

After the prediction is performed using FPKfe, the filter update step delivers the optimal estimate of the PDF of the state:

$$\hat{p}\left(\boldsymbol{x}(t_n)\,|\,\boldsymbol{z}_n\right) = \frac{p_{\text{obs}}\left(\boldsymbol{z}_n\,|\,\boldsymbol{x}\right) p_{\text{pred}}\left(\boldsymbol{x}(t_n)\,|\,\boldsymbol{z}_{n-1}\right)}{\int p_{\text{obs}}\left(\boldsymbol{z}_n\,|\,\boldsymbol{\xi}\right) p_{\text{pred}}\left(\boldsymbol{\xi}(t_n)\,|\,\boldsymbol{z}_{n-1}\right) d\boldsymbol{\xi}}. \tag{5.11}$$

A litany of numerical algorithms have been developed to solve the nonlinear filtering problem using this recursive optimal Bayesian filter (OBF) structure, including the methods described by Balaji [9][10][11], Fox et al. [36], Sarkka [95], and Lototsky

[72]. While all these authors made strides in solving their respective filtering problems by leveraging the optimal Bayesian approach, none of them sought to solve a target tracking problem of the complexity or dimensionality found in the spherical CV or EPC models.

5.2.2. Spherical Fokker-Planck-Kolmogorov Forward Equation

In much of the prior work concerning C-D nonlinear filtering, all drift and diffusion functions are assumed to lie in the Cartesian R^n space [9][121]. This will not suffice for the family of EPC target tracking filters because the EPC models live in spherical coordinates. FPKfe must be solved on the manifold described by the spherical coordinates and their respective velocities.

The FPKfe has been used to solve a variety of nonlinear filtering problems in various fields, including chemical reactions [85] and cosmic ray transport [110]. Balaji provides several examples of target tracking in Cartesian coordinates [11] under various constraints, such as constant velocity or constant turn rate. However, the available literature does not feature the FPKfe for tracking a CV target in spherical coordinates; therefore, it is derived in this dissertation for the first time to our knowledge. In this section, we transform FPKfe for the Cartesian constant velocity model with white acceleration noise into its spherical counterpart. From there, the EPC spherical filter equations follow naturally based on our prior derivations in Section 2.4.1.

The continuous-time CV state model has been well described by multiple authors, including Shimkin [102], whose lecture notes clearly lay out the CV model as well as methods for discretizing the plant noise covariance $\boldsymbol{Q}$. For this derivation, we make the

generally acceptable assumption [53] that the time derivative of the state Brownian motion **v**(t) is a zero-mean Gaussian with covariance $\boldsymbol{Q}$

$$\dot{\boldsymbol{v}}(t) \sim \mathrm{N}\left(\boldsymbol{0}, \boldsymbol{Q}(t)\right) \tag{5.12}$$

The state model in continuous time can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), t\right) + \mathbf{G}(\boldsymbol{x})\dot{\boldsymbol{v}}(t) \tag{5.13}$$

We are currently working with a constant velocity target, where the noise contributions are due to unknown accelerations that are treated as random. In ENU Cartesian coordinates, the dynamic equations are linear, and the result is

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{F} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{G} \dot{\boldsymbol{v}}(t). \tag{5.14}$$

Equation (5.14) can be simplified as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{x} \\ 0 \\ \dot{y} \\ 0 \\ \dot{z} \\ 0 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{G} \dot{\boldsymbol{v}}(t). \tag{5.15}$$

We modeled the noise for the CV case to include the greatest acceleration expected for the evaluated target set. These target accelerations are modeled in the matrix **Q** as follows:

$$\boldsymbol{Q} = q^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.16}$$

where the dynamic noise intensity (q) is assumed to be the same across all three dimensions, and can be computed using the methods described in Section 2.2.1. Calculating $\boldsymbol{GQG}^{\mathrm{T}}$ to obtain the effective diffusion yields a matrix with the square of the noise intensity along the main diagonal in the second, fourth, and sixth rows and columns:

$$\boldsymbol{GQG}^{\mathrm{T}} = q^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.17}$$

In Cartesian coordinates, the FPKfe for the Cartesian constant velocity model described in (5.15) is:

$$\begin{aligned} \frac{\partial p}{\partial t} &= -\sum_{i=1}^{6} \left[f_i \frac{\partial p}{\partial x_i} + p \frac{\partial f_i}{\partial x_i} \right] + \sum_{i,j}^{6} \frac{\partial^2 \left[\left(\boldsymbol{GQG}' \right)_{ij} p \right]}{\partial x_i \partial x_j} \\ &= \frac{q^2}{2} \left[\frac{\partial^2 p}{\partial \dot{x}^2} + \frac{\partial^2 p}{\partial \dot{y}^2} + \frac{\partial^2 p}{\partial \dot{z}^2} \right] - \left[\dot{x} \frac{\partial p}{\partial x} + \dot{y} \frac{\partial p}{\partial y} + \dot{z} \frac{\partial p}{\partial z} \right] \end{aligned} \tag{5.18}$$

We must now change the coordinates to a spherical coordinate system. Changing the coordinates makes use of the following spherical-to-Cartesian [43] relations:

$$x = r \sin\theta \cos\varphi, \tag{5.19}$$

$$y = r \cos\theta \cos\varphi, \tag{5.20}$$

$$z = r \sin\varphi, \tag{5.21}$$

$$\dot{x} = \dot{r} \sin\theta \cos\varphi + \dot{\theta} r \cos\theta \cos\varphi - \dot{\varphi} r \sin\theta \sin\varphi, \tag{5.22}$$

$$\dot{y} = \dot{r}\sin\theta\cos\varphi - \dot{\theta}r\sin\theta\cos\varphi - \dot{\varphi}r\cos\theta\sin\varphi, \tag{5.23}$$

$$\dot{z} = \dot{r}\sin\theta + \dot{\varphi}r\cos\theta. \tag{5.24}$$

The Jacobian of the relations in (5.19)–(5.24) and the liberal use of the chain rule for partial derivatives are used to convert Equation (5.18) into spherical coordinates. The Jacobian provides the coefficients of the system of equations as follows:

$$\frac{\partial p}{\partial r} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial p}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial r} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial r} + \frac{\partial p}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial r} + \frac{\partial p}{\partial z}\frac{\partial z}{\partial r} + \frac{\partial p}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial r}, \tag{5.25}$$

$$\frac{\partial p}{\partial \dot{r}} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \dot{r}} + \frac{\partial p}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial \dot{r}} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \dot{r}} + \frac{\partial p}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial \dot{r}} + \frac{\partial p}{\partial z}\frac{\partial z}{\partial \dot{r}} + \frac{\partial p}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial \dot{r}}, \tag{5.26}$$

$$\frac{\partial p}{\partial \theta} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial p}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial \theta} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \theta} + \frac{\partial p}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial \theta} + \frac{\partial p}{\partial z}\frac{\partial z}{\partial \theta} + \frac{\partial p}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial \theta}, \tag{5.27}$$

$$\frac{\partial p}{\partial \dot{\theta}} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \dot{\theta}} + \frac{\partial p}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial \dot{\theta}} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \dot{\theta}} + \frac{\partial p}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial \dot{\theta}} + \frac{\partial p}{\partial z}\frac{\partial z}{\partial \dot{\theta}} + \frac{\partial p}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial \dot{\theta}}, \tag{5.28}$$

$$\frac{\partial p}{\partial \varphi} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \varphi} + \frac{\partial p}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial \varphi} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \varphi} + \frac{\partial p}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial \varphi} + \frac{\partial p}{\partial z}\frac{\partial z}{\partial \varphi} + \frac{\partial p}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial \varphi}, \tag{5.29}$$

$$\frac{\partial p}{\partial \dot{\varphi}} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \dot{\varphi}} + \frac{\partial p}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial \dot{\varphi}} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \dot{\varphi}} + \frac{\partial p}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial \dot{\varphi}} + \frac{\partial p}{\partial z}\frac{\partial z}{\partial \dot{\varphi}} + \frac{\partial p}{\partial \dot{z}}\frac{\partial \dot{z}}{\partial \dot{\varphi}}, \tag{5.30}$$

Using the method described by Chirikjian [27], we can convert all Cartesian spatial derivatives into spherical coordinates:

$$\begin{aligned}\frac{\partial p}{\partial x} = {} & \sin\theta\cos\varphi\frac{\partial p}{\partial r} + \left(\frac{\cos\theta}{r\cos\varphi}\right)\frac{\partial p}{\partial \theta} + \left(-\frac{\sin\theta\sin\varphi}{r}\right)\frac{\partial p}{\partial \varphi} \\ & \left(\dot{\theta}\cos\theta\cos\varphi - \dot{\varphi}\sin\theta\sin\varphi\right)\frac{\partial p}{\partial \dot{r}} + \\ & \left(-\frac{1}{r^2\cos^2\varphi}\right)\left(\dot{r}\cos\theta\cos\varphi + \dot{\theta}r\sin\theta\cos\varphi - \dot{\varphi}r\cos\theta\sin\varphi\right)\frac{\partial p}{\partial \dot{\theta}} + \\ & \left(-\frac{1}{r^2}\right)\left(\dot{\varphi}r\cos\varphi\sin\theta - \dot{r}\sin\theta\sin\varphi + \dot{\theta}r\cos\theta\sin\varphi\right)\frac{\partial p}{\partial \dot{\varphi}}.\end{aligned} \tag{5.31}$$

$$\frac{\partial p}{\partial y} = \cos\theta\cos\varphi\frac{\partial p}{\partial r} - \left(\frac{\sin(\theta)}{r\cos\varphi}\right)\frac{\partial p}{\partial \theta} - \left(\frac{\cos\theta\sin\varphi}{r}\right)\frac{\partial p}{\partial \varphi} - \left(\dot{\theta}\sin\theta\cos\varphi + \dot{\varphi}\cos\theta\sin\varphi\right)\frac{\partial p}{\partial \dot{r}} + \left(-\frac{1}{r^2\cos^2\varphi}\right)\left(\dot{\theta}r\cos\theta\cos\varphi - \dot{r}\cos\varphi\sin\theta + \dot{\varphi}r\sin\theta\sin\varphi\right)\frac{\partial p}{\partial \dot{\theta}} + \frac{1}{r^2}\left(\dot{r}\sin\theta\sin\varphi - \dot{\varphi}r\cos\theta\cos\varphi + \dot{\theta}r\sin\theta\sin\varphi\right)\frac{\partial p}{\partial \dot{\varphi}}; \tag{5.32}$$

$$\frac{\partial p}{\partial z} = \sin\varphi\frac{\partial p}{\partial r} + \dot{\varphi}\cos\varphi\frac{\partial p}{\partial \dot{r}} + \left(\frac{\cos\varphi}{r}\right)\frac{\partial p}{\partial \varphi} + \left(-\frac{1}{r^2}\right)\left(\dot{r}\cos\varphi + \dot{\varphi}r\sin\varphi\right)\frac{\partial p}{\partial \dot{\varphi}}. \tag{5.33}$$

The derivatives with respect to the velocities are

$$\frac{\partial p}{\partial \dot{x}} = \sin\theta\cos\varphi\frac{\partial p}{\partial \dot{r}} + \frac{\cos\theta}{r\cos\varphi}\frac{\partial p}{\partial \dot{\theta}} - \frac{\sin\theta\sin\varphi}{r}\frac{\partial p}{\partial \dot{\varphi}}; \tag{5.34}$$

$$\frac{\partial p}{\partial \dot{y}} = \cos\theta\cos\varphi\frac{\partial p}{\partial \dot{r}} - \frac{\sin\theta}{r\cos\varphi}\frac{\partial p}{\partial \dot{\theta}} - \frac{\cos\theta\sin\varphi}{r}\frac{\partial p}{\partial \dot{\varphi}}; \tag{5.35}$$

$$\frac{\partial p}{\partial \dot{z}} = \sin\varphi\frac{\partial p}{\partial \dot{r}} + \frac{\cos\varphi}{r}\frac{\partial p}{\partial \dot{\varphi}}. \tag{5.36}$$

The Jacobian method must be applied a second time to obtain second-order derivative conversions for the velocities:

$$\frac{\partial^2 p}{\partial \dot{x}^2} = \left(\sin^2\theta\cos^2\varphi\frac{\partial^2 p}{\partial \dot{r}^2} + \frac{\cos\theta\sin\theta\cos\varphi}{r\cos\varphi}\frac{\partial}{\partial \dot{r}}\frac{\partial p}{\partial \dot{\theta}} - \frac{\sin^2\theta\cos\varphi\sin\varphi}{r}\frac{\partial}{\partial \dot{r}}\frac{\partial p}{\partial \dot{\varphi}}\right) + \left(\frac{\cos\theta\sin\theta\cos\varphi}{r\cos\varphi}\frac{\partial}{\partial \dot{\theta}}\frac{\partial p}{\partial \dot{r}} + \frac{\cos^2\theta}{r^2\cos^2\varphi}\frac{\partial^2 p}{\partial \dot{\theta}^2} - \frac{\cos\theta\sin\theta\sin\varphi}{r^2\cos\varphi}\frac{\partial}{\partial \dot{\theta}}\frac{\partial p}{\partial \dot{\varphi}}\right) + \left(-\frac{\sin^2\theta\cos\varphi\sin\varphi}{r}\frac{\partial}{\partial \dot{\varphi}}\frac{\partial p}{\partial \dot{r}} - \frac{\cos\theta\sin\theta\sin\varphi}{r^2\cos\varphi}\frac{\partial}{\partial \dot{\varphi}}\frac{\partial p}{\partial \dot{\theta}} + \frac{\sin^2\theta\sin^2\varphi}{r^2}\frac{\partial^2 p}{\partial \dot{\varphi}^2}\right); \tag{5.37}$$

$$\frac{\partial^2 p}{\partial \dot{y}^2} = \left(\cos^2\theta\cos^2\varphi \frac{\partial^2 p}{\partial \dot{r}^2} - \frac{\sin\theta\cos\theta\cos\varphi}{r\cos\varphi} \frac{\partial}{\partial \dot{r}} \frac{\partial p}{\partial \dot{\theta}} - \frac{\cos^2\theta\cos\varphi\sin\varphi}{r} \frac{\partial}{\partial \dot{r}} \frac{\partial p}{\partial \dot{\varphi}} \right) +$$
$$\left(-\frac{\sin\theta\cos\theta\cos\varphi}{r\cos\varphi} \frac{\partial}{\partial \dot{\theta}} \frac{\partial p}{\partial \dot{r}} + \frac{\sin^2\theta}{r^2\cos^2\varphi} \frac{\partial^2 p}{\partial \dot{\theta}^2} + \frac{\sin\theta\cos\theta\sin\varphi}{r^2\cos\varphi} \frac{\partial}{\partial \dot{\theta}} \frac{\partial p}{\partial \dot{\varphi}} \right) +$$
$$\left(-\frac{\cos^2\theta\cos\varphi\sin\varphi}{r} \frac{\partial}{\partial \dot{\varphi}} \frac{\partial p}{\partial \dot{r}} + \frac{\sin\theta\cos\theta\sin\varphi}{r^2\cos\varphi} \frac{\partial}{\partial \dot{\varphi}} \frac{\partial p}{\partial \dot{\theta}} + \frac{\cos^2\theta\sin^2\varphi}{r^2} \frac{\partial^2 p}{\partial \dot{\varphi}^2} \right);$$

(5.38)

$$\frac{\partial^2 p}{\partial \dot{z}^2} = \left(\sin^2\varphi \frac{\partial p}{\partial \dot{r}^2} + \frac{\sin\varphi\cos\varphi}{r} \frac{\partial}{\partial \dot{r}} \frac{\partial p}{\partial \dot{\varphi}} \right) + \left(\frac{\sin\varphi\cos\varphi}{r} \frac{\partial}{\partial \dot{\varphi}} \frac{\partial p}{\partial \dot{r}} + \frac{\cos^2\varphi}{r^2} \frac{\partial^2 p}{\partial \dot{\varphi}^2} \right).$$

(5.39)

The sum of the second derivatives initially contains many terms, but they can be greatly simplified with basic trigonometric identities. Further simplifications are obtained if we assumed that the noise intensity in each direction was equal (q_x=q_y= q_z=q). Conveniently, the assumption set forth with the D-D filters explored in earlier chapters requires no additional changes to the spherical FPKfe formulation. Collecting the terms, which are the coefficients of each spherical coordinate velocity derivative, we obtain

$$\frac{\partial^2 p}{\partial \dot{x}^2} + \frac{\partial^2 p}{\partial \dot{y}^2} + \frac{\partial^2 p}{\partial \dot{z}^2} = \frac{\partial^2 p}{\partial \dot{r}^2} + \left(\frac{1}{r\cos\varphi} \right)^2 \frac{\partial^2 p}{\partial \dot{\theta}^2} + \left(\frac{1}{r} \right)^2 \frac{\partial^2 p}{\partial \dot{\varphi}^2}, \tag{5.40}$$

for the velocity derivative terms in Equation (5.18). After substituting (5.40) into (5.18), the constant velocity FPKfe for the CV spherical target tracking problem is

$$\begin{aligned}\frac{\partial p}{\partial t} &= \frac{q^2}{2}\left[\frac{\partial^2 p}{\partial \dot{x}^2}+\frac{\partial^2 p}{\partial \dot{y}^2}+\frac{\partial^2 p}{\partial \dot{z}^2}\right]-\left[\dot{x}\frac{\partial p}{\partial x}+\dot{y}\frac{\partial p}{\partial y}+\dot{z}\frac{\partial p}{\partial z}\right] \\ &= \frac{q^2}{2}\left[\frac{\partial^2 p}{\partial \dot{r}^2}+\left(\frac{1}{r\cos\varphi}\right)^2\frac{\partial^2 p}{\partial \dot{\theta}^2}+\left(\frac{1}{r}\right)^2\frac{\partial^2 p}{\partial \dot{\varphi}^2}\right]- \\ &\left(\dot{r}\sin\theta\cos\varphi+\dot{\theta}r\cos\theta\cos\varphi-\dot{\varphi}r\sin\theta\sin\varphi\right)\frac{\partial p}{\partial x}- \\ &\left(\dot{r}\sin\theta\cos\varphi-\dot{\theta}r\sin\theta\cos\varphi-\dot{\varphi}r\cos\theta\sin\varphi\right)\frac{\partial p}{\partial y}- \\ &\left(\dot{r}\sin\theta+\dot{\varphi}r\cos\theta\right)\frac{\partial p}{\partial z}.\end{aligned} \tag{5.41}$$

The final step is to collect the terms for the first-order partial derivatives. Doing so, we obtain:

$$\begin{aligned}&\frac{\partial p}{\partial t} = \frac{\sigma_w^2}{2}\left[\frac{\partial^2 p}{\partial \dot{r}^2}+\left(\frac{1}{r\cos\varphi}\right)^2\frac{\partial^2 p}{\partial \dot{\theta}^2}+\left(\frac{1}{r}\right)^2\frac{\partial^2 p}{\partial \dot{\varphi}^2}\right]- \\ &\left[\dot{r}\frac{\partial p}{\partial r}+r\left(\dot{\varphi}^2+\dot{\theta}^2\cos^2\varphi\right)\frac{\partial p}{\partial \dot{r}}\right]- \\ &\left[\dot{\theta}\frac{\partial p}{\partial \theta}+2\dot{\theta}\left(\dot{\varphi}\tan\varphi-\frac{\dot{r}}{r}\right)\frac{\partial p}{\partial \dot{\theta}}\right]- \\ &\left[\dot{\varphi}\frac{\partial p}{\partial \varphi}-\left(\dot{\theta}^2\cos\varphi\sin\varphi+\frac{2\dot{\varphi}\dot{r}}{r}\right)\frac{\partial p}{\partial \dot{\varphi}}\right].\end{aligned} \tag{5.42}$$

The terms that act as coefficients for the first-order derivatives of the velocity components were the same as those found in the CV spherical dynamic model (2.103). Applying the proportional navigation constant to the appropriate terms in the same way as in Section 2.4.1 yields

$$\frac{\partial p}{\partial t} = \frac{\sigma_w^2}{2}\left[\frac{\partial^2 p}{\partial \dot{r}^2} + \left(\frac{1}{r\cos\varphi}\right)^2 \frac{\partial^2 p}{\partial \dot{\theta}^2} + \left(\frac{1}{r}\right)^2 \frac{\partial^2 p}{\partial \dot{\varphi}^2}\right] -$$
$$\left[\dot{r}\frac{\partial p}{\partial r} + (1-k)r\left(\dot{\varphi}^2 + \dot{\theta}^2\cos^2\varphi\right)\frac{\partial p}{\partial \dot{r}}\right] -$$
$$\left[\dot{\theta}\frac{\partial p}{\partial \theta} + (2-k)\dot{\theta}\left(\dot{\varphi}\tan\varphi - \frac{\dot{r}}{r}\right)\frac{\partial p}{\partial \dot{\theta}}\right] - \tag{5.43}$$
$$\left[\dot{\varphi}\frac{\partial p}{\partial \varphi} - \left((1-k)\dot{\theta}^2\cos\varphi\sin\varphi + \frac{(2-k)\dot{\varphi}\dot{r}}{r}\right)\frac{\partial p}{\partial \dot{\varphi}}\right].$$

The PDE (5.43) represents the drift and diffusion of the state model for the EPC spherical model. This equation can be used to propagate forward in time the PDF governing the target state in the absence of measurement data, as described in Section 5.2.1. Now that the FPKfe for the EPC spherical target tracking problem has been defined, the next step is to examine methods for solving the PDE, which may be used in the formulation of an EPC OBF. This will be facilitated through the examination of a handful of simpler problems with lower dimensionality, ultimately building up to the EPC OBF methods, which will be compared with the D-D algorithms previously presented by Ford and Haug [45][34].

5.3.Preliminary Work with OBFs

5.3.1. OBF on the Circle with Known Velocity

The simplest version of the target tracking problem on the sphere is tracking only the azimuth. Tracking a particle on a circle that moves with a known constant velocity is a simple way to demonstrate how the drift and diffusion terms of FPKfe influence the evolution of the target state PDF. In this study, we demonstrate how FPKfe is used to

propagate an initial PDF for the target state forward in time, in the absence of new measurement data.

In the 1970s, Willsky and Lo [111][112] presented a series of examples for estimating processes on the circle S^1, ultimately leading to a filter architecture that supports phase-locked loops. One example is a particle traveling on a circle with a known constant velocity. Taking their example and applying it to our basic problem (5.43), we can model FPKfe for the state of the particle (θ) with

$$\frac{\partial p}{\partial t} = \frac{q^2}{2}\left[\frac{\partial^2 p}{\partial \theta^2}\right] - \left[\omega_c \frac{\partial p}{\partial \theta}\right], \tag{5.44}$$

where q is the dynamic noise intensity, and ω_c is the target's known constant velocity. In his examples, Willsky used Fourier basis functions to solve the PDE to great effect. For this example, we opted to leverage Euler's method for backward-time, central space (BTCS) [24] because of the utility of discretization methods in solving the higher-dimensional problems examined later in this chapter, including (5.43).

The BTCS finite-difference method discretizes the solution of (5.44) such that it can be repeatedly solved in terms of a system of ODEs. By discretizing time into increments of Δt and angle into increments of $\Delta\theta$, one can use the relations

$$\frac{\partial p}{\partial t} \approx \frac{1}{\Delta t}\left[p\left(\theta_j, t_n\right) - p\left(\theta_j, t_{n-1}\right)\right], \tag{5.45}$$

$$\frac{\partial^2 p}{\partial \theta^2} \approx \frac{1}{\left(\Delta\theta\right)^2}\left[p\left(\theta_{j+1}, t_n\right) + p\left(\theta_{j-1}, t_n\right) - 2p\left(\theta_j, t_n\right)\right], \tag{5.46}$$

and

$$\frac{\partial p}{\partial \theta} \approx \frac{1}{2\Delta\theta}\left[p\left(\theta_{j+1}, t_n\right) - p\left(\theta_{j-1}, t_{n-1}\right)\right] \tag{5.47}$$

to rewrite (5.44) as

$$\begin{aligned}&\frac{1}{\Delta t}\left[p\left(\theta_j,t_n\right)-p\left(\theta_j,t_{n-1}\right)\right]=\ldots\\&\frac{q^2}{2}\frac{1}{\left(\Delta\theta\right)^2}\left[p\left(\theta_{j+1},t_n\right)+p\left(\theta_{j-1},t_n\right)-2p\left(\theta_j,t_n\right)\right]-\ldots\\&\omega_c\frac{1}{2\Delta\theta}\left[p\left(\theta_{j+1},t_n\right)-p\left(\theta_{j-1},t_n\right)\right].\end{aligned}\tag{5.48}$$

The difference equation (5.48) can be rearranged into a system of equations to solve for the discretized function **p**(θ,t), where the probability distribution is transformed into a vector representation to capture its value at each angle node. The system of equations is as follows:

$$\left(\boldsymbol{I}-\Delta t\boldsymbol{B}\right)\boldsymbol{p}(\boldsymbol{\theta},t_n)=\boldsymbol{p}(\boldsymbol{\theta},t_{n-1}),\tag{5.49}$$

where the elements of transition matrix ***B*** can be defined along its main diagonal and the first upper and first lower diagonals.

$$\begin{aligned}B_{jj'}=&\frac{q^2}{\left(\Delta\theta\right)^2}\delta\left(j,j'\right)-\frac{1}{2}\left(\frac{q^2}{\left(\Delta\theta\right)^2}-\frac{\omega_c}{\Delta\theta}\right)\delta\left(j,j'+1\right)-\\&\frac{1}{2}\left(\frac{q^2}{\left(\Delta\theta\right)^2}+\frac{\omega_c}{\Delta\theta}\right)\delta\left(j,j'-1\right).\end{aligned}\tag{5.50}$$

Solving system (5.49) can be performed using a multitude of methods. All algorithms developed and tested in this dissertation were written in MATLAB, which uses the logic process illustrated in Figure 85 to determine which method is optimal for the given system. For Equation (5.49), LU decomposition was used.

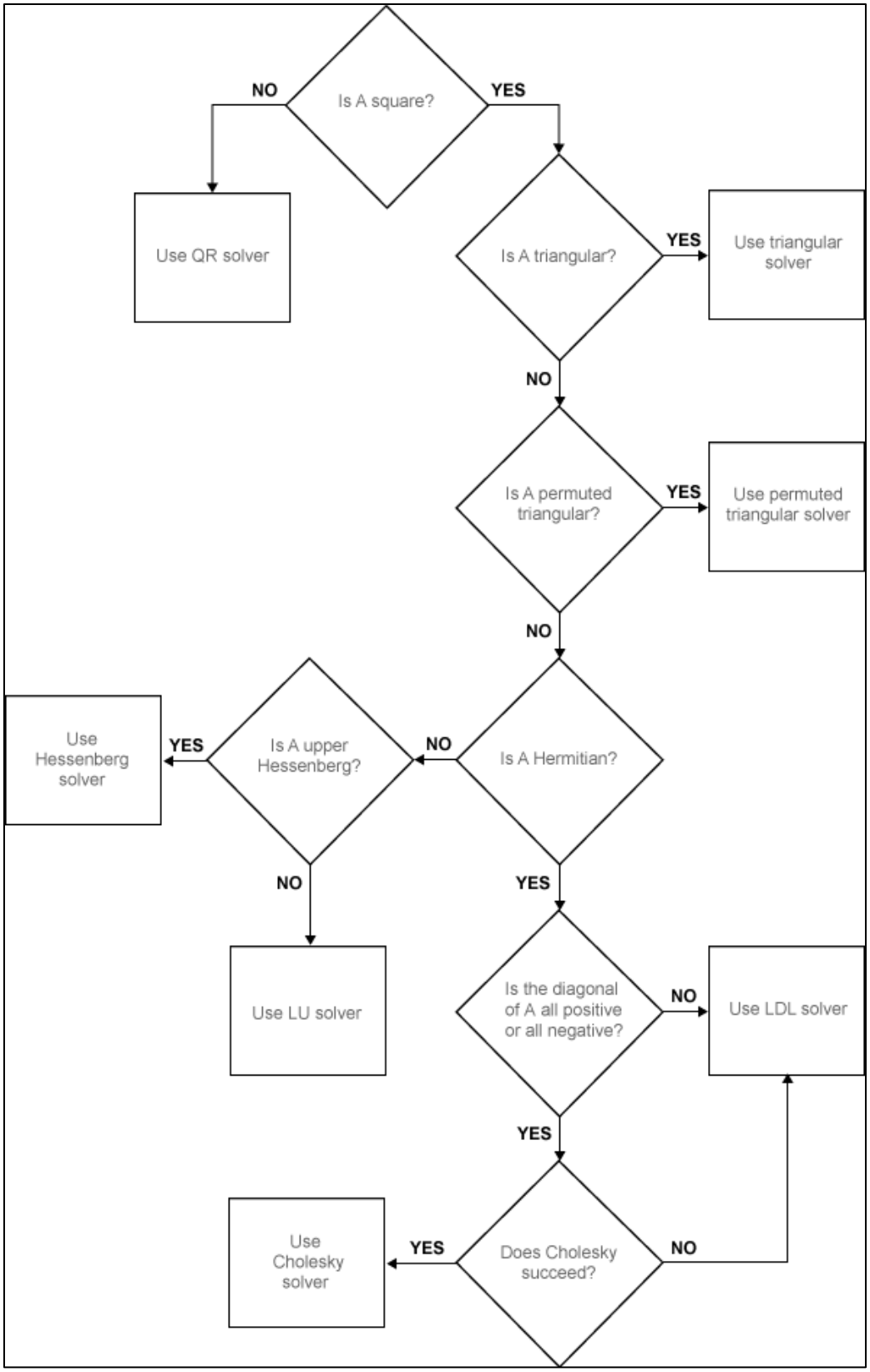

Figure 85: Flow diagram for MATLAB's linear system solver mldivide, given non-sparse input matrix (***A***) as shown on the MATLAB help page [81].

Solving Equation (5.49) given an initial state PDF $p_0(\theta)$, propagates the PDF forward by one time increment Δt. The diffusion of the PDF caused by the terms originating from (5.46) can be tuned by changing the value of q. A large value of q implies that the uncertainty of the target state increases at a faster rate than for a small value of q. This uncertainty translates into a faster spread of the distribution over time. The known constant velocity ω_c translates into a shift in the PDF over time. Both the spread (diffusion) and shift (drift) behaviors can be observed in the results of a simple numerical experiment, as shown in Figure 86.

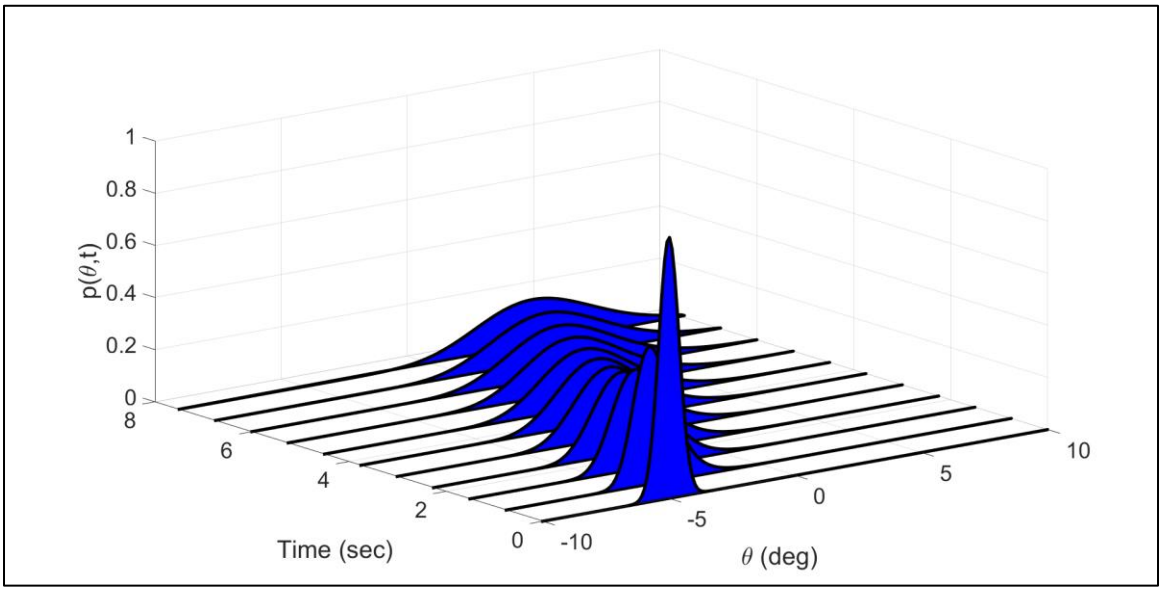

Figure 86: Normalized PDF diffusion and drift for particle moving on the circle with known constant velocity.

In the example above, the particle starts at an angle of -5 degrees and travels clockwise at a rate (ω_c) of 1.25 deg/sec. Dynamic noise intensity is assumed to be q = 0.573 deg/sec, with time increment Δt = 0.25 seconds and angle increment $\Delta\theta$ = 0.1 degrees. The initial PDF was propagated forward by 33 time increments, although only 11 of these were plotted for visualization purposes. If, at any time, a new measurement is delivered to the tracking algorithm, the estimated PDF for the state can be updated using

(5.11), which reduces the uncertainty in the system and prompts the algorithm to propagate the PDF forward again until the next measurement. The example and basic principles laid out in this section can be extended to more complex problems using similar techniques.

Forward time centered space (FTCS) [24] will be explored for later implementations of the EPC OBF owing to its memory savings and computation speed improvements. FTCS is formulated similarly to the BTCS but removes the need to solve (5.49) in exchange for smaller increments of time to satisfy the system's stability requirements [24]. For FTCS, the substitution (5.45) becomes

$$\frac{\partial p}{\partial t} \approx \frac{1}{\Delta t}\left[p\left(\theta_j, t_{k+1}\right) - p\left(\theta_j, t_k\right)\right], \tag{5.51}$$

while (5.46)–(5.47) remain constant. After shifting all time indices back by one step, the FTCS difference equation becomes

$$\boldsymbol{p}(\boldsymbol{\theta}, t_n) = \left(\boldsymbol{I} + \Delta t \boldsymbol{B}\right) \boldsymbol{p}(\boldsymbol{\theta}, t_{n-1}). \tag{5.52}$$

5.3.2. OBF on the Circle with Unknown Velocity

Most available journal articles that discuss using OBFs or Yau-type filters demonstrate their algorithms by exercising them against one-dimensional problems [9][10][11][13][14][19][73][74][85][111][120][121] [123]. However, to track an object moving on S^1 or S^2 with an unknown velocity, problems of higher dimensionality must be considered. When a target tracking problem is approached using a CV assumption, a multitude of simple methods, including the linear Kalman filter, can be used. However, for the sake of better-illustrating the procedures which will be used for the EPC spherical

OBF, we shall use the discretization methods discussed in Section 5.3.1 to solve the following PDE

$$\frac{\partial p}{\partial t} = \frac{q^2}{2}\left[\frac{\partial^2 p}{\partial \dot{\theta}^2}\right] - \left[\dot{\theta}\frac{\partial p}{\partial \theta}\right]. \tag{5.53}$$

Now that two unknown variables are at play, bearing, and bearing rate, it is necessary to define a new indexing structure to keep track of the PDF on the discretized grid. For the algorithms examined in this dissertation, the FTCS and BTCS methods for solving FPKfe were used. This means that PDF p($\boldsymbol{x}$,t) is maintained as a vector of all unique combinations of the bearing and bearing rate points in the discretized grid. This choice allows the transition matrix $\boldsymbol{B}$ to remain a matrix, even for higher-dimensional problems. While this approach suffers from the so-called "curse of dimensionality" [36], it is still possible to construct functional nonlinear target tracking filters if the grid sizes are chosen carefully.

To maintain the PDF as a vector, define the indexing matrix

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{j}_1 & \boldsymbol{j}_2 \end{bmatrix}, \tag{5.54}$$

where the columns represent the indices in the discretized grid for the bearing ($\boldsymbol{j}_1$) and bearing rate ($\boldsymbol{j}_2$). Each row of J represents a unique combination of the bearing and bearing rate indices, thus defining the entire grid space on which FPKfe is being solved. Individual elements $\boldsymbol{j}_1$ (j_1) and $\boldsymbol{j}_2$ (j_2) can be used alongside time index i to obtain the FTCS difference equation:

$$\left[p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2)}\right) - p\left(t^{(i)},\theta^{(j_1)},\dot{\theta}^{(j_2)}\right)\right] = \frac{q^2\Delta t}{2\left(\Delta\dot{\theta}\right)^2} \times$$
$$\left[p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2+1)}\right) - 2p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2)}\right) + p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2-1)}\right)\right] - \quad (5.55)$$
$$\frac{\dot{\theta}^{(j_2)}\Delta t}{2\left(\Delta\theta\right)}\left[p\left(t^{(i+1)},\theta^{(j_1+1)},\dot{\theta}^{(j_2)}\right) - p\left(t^{(i+1)},\theta^{(j_1-1)},\dot{\theta}^{(j_2)}\right)\right].$$

Rearranging the difference equation to obtain the predicted PDF on the left-hand side yields:

$$p\left(t^{(i)},\theta^{(j_1)},\dot{\theta}^{(j_2)}\right) = \left[1 + \frac{q^2\Delta t}{\left(\Delta\dot{\theta}\right)^2}\right] p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2)}\right) +$$
$$\left[-\frac{q^2\Delta t}{2\left(\Delta\dot{\theta}\right)^2}\right] p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2+1)}\right) + \left[-\frac{q^2\Delta t}{2\left(\Delta\dot{\theta}\right)^2}\right] p\left(t^{(i+1)},\theta^{(j_1)},\dot{\theta}^{(j_2-1)}\right) + \quad (5.56)$$
$$\left[\frac{\dot{\theta}^{(j_2)}\Delta t}{2\left(\Delta\theta\right)}\right] p\left(t^{(i+1)},\theta^{(j_1+1)},\dot{\theta}^{(j_2)}\right) + \left[-\frac{\dot{\theta}^{(j_2)}\Delta t}{2\left(\Delta\theta\right)}\right] p\left(t^{(i+1)},\theta^{(j_1-1)},\dot{\theta}^{(j_2)}\right)$$

The elements of the transition matrix are

$$\boldsymbol{B}_{jj'} = \left[\frac{q^2\Delta t}{\left(\Delta\dot{\theta}\right)^2}\right]\delta\left(j_1, j_1'\right)\delta\left(j_2, j_2'\right) +$$
$$\left[-\frac{q^2\Delta t}{2\left(\Delta\dot{\theta}\right)^2}\right]\delta\left(j_1, j_1'\right)\delta\left(j_2, j_2'+1\right) +$$
$$\left[-\frac{q^2\Delta t}{2\left(\Delta\dot{\theta}\right)^2}\right]\delta\left(j_1, j_1'\right)\delta\left(j_2, j_2'-1\right) \quad (5.57)$$
$$\left[\frac{\dot{\theta}^{(j_2)}\Delta t}{2\left(\Delta\theta\right)}\right]\delta\left(j_1, j_1'+1\right)\delta\left(j_2, j_2'\right) +$$
$$\left[-\frac{\dot{\theta}^{(j_2)}\Delta t}{2\left(\Delta\theta\right)}\right]\delta\left(j_1, j_1'-1\right)\delta\left(j_2, j_2'\right).$$

A series of simple experiments were performed to validate the functionality of this filter alongside the linear D-D Kalman filter. The results are found in Figure 87 and Figure 88. The angle and velocity errors for the OBF tend to be the same if not slightly better than those for the KF despite using the same input parameters regarding measurement data, dynamic model noise intensity (q), and update periods. These promising results motivate further exploration to develop an EPC OBF that can be compared to the family of EPC target tracking filters already established.

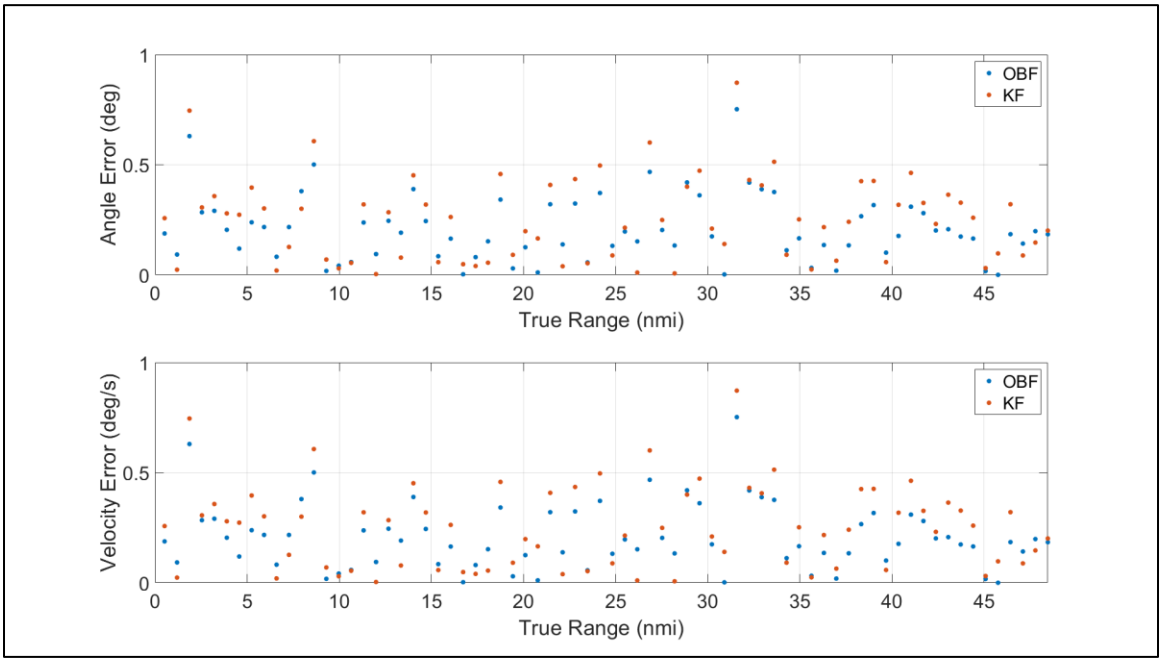

Figure 87: Angular and velocity error comparison for the linear KF and OBF for the radially inbound trajectory.

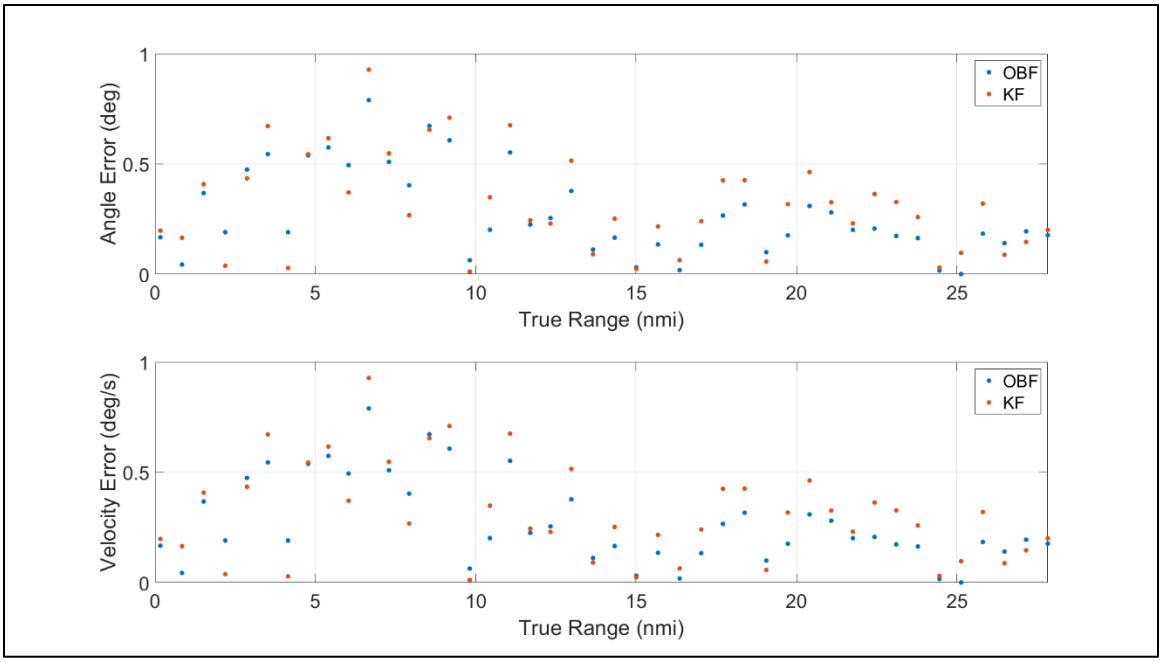

Figure 88: Angular and velocity error comparison for the linear KF and OBF for the Dutch roll trajectory.

5.3.3. FTCS Finite Difference OBF Algorithm

The initial state and covariance data delivered by the combat system to the D-D filters can also be used to initialize the OBFs. This can be performed by generating the initial discretized probability vector $\mathbf{p}_0(\mathbf{x})$ assuming a Gaussian distribution because only the initial mean ($\boldsymbol{\mu}_0$) and covariance ($\boldsymbol{\Sigma}_0$) are provided:

$$\mathbf{p}_0(\boldsymbol{x}) = \frac{1}{(2\pi)^3 |\boldsymbol{\Sigma}_0|^{1/2}} \exp\left[-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_0) \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_0)^{\mathrm{T}} \right]. \tag{5.58}$$

The algorithm is capable of handling other types of initial distributions, given that prior knowledge regarding incoming state estimate data is available.

The initial distribution must be propagated to the system update time until the measurement update arrives. Recall that the combat system under examination updates every 0.25 seconds, while measurement updates are performed once every 4 seconds.

Owing to its simplicity and speed, the FTCS method will be used to propagate the much larger finite-difference equations found in the higher-dimensional EPC OBF algorithms. The time increment must be sufficiently small to maintain the stability in the solution [24]. The algorithm multiplies the propagation steps together N_{prop} times, saving off the probability data every 0.25 seconds, until 4 seconds has been reached. Thus, the propagation steps for the initial probability density are:

$$\boldsymbol{p}_{pred}\left(\boldsymbol{x},t_1\right)=\left[\prod_{i=1}^{N_{prop}}\left(\boldsymbol{I}+\Delta t\boldsymbol{B}\right)\right]\boldsymbol{p}_0\left(\boldsymbol{x}\right). \tag{5.59}$$

For all updates after the first measurement update, the propagation steps are

$$\boldsymbol{p}_{pred}\left(\boldsymbol{x},t_n\right)=\left[\prod_{i=1}^{N_{prop}}\left(\boldsymbol{I}+\Delta t\boldsymbol{B}\right)\right]\hat{\boldsymbol{p}}(\boldsymbol{\theta},t_{n-1}). \tag{5.60}$$

As described in Section 5.2.1, observations for radar target tracking are typically treated as Gaussian. This was also done to maintain consistency with previously presented algorithms. When the latest observation data are delivered, the measurement probability vector

$$\mathbf{p}_{obs}\left(\boldsymbol{x},t_n\right)=\frac{1}{\left(2\pi\right)^3\left|\boldsymbol{\Sigma}_0\right|^{1/2}}\exp\left[-\frac{1}{2}\left(\boldsymbol{z}-\boldsymbol{z}_{meas}\right)\boldsymbol{R}^{-1}\left(\boldsymbol{z}-\boldsymbol{z}_{meas}\right)^{\mathrm{T}}\right], \tag{5.61}$$

where $\boldsymbol{R}$ and the components of the measurement vector $\boldsymbol{z}$ are defined in Section 2.2.1.

To perform the filter update step from Equation (5.11), the observation and predicted probabilities are multiplied element-wise and normalized using the trapezoidal numerical integration method [24]

$$\hat{\boldsymbol{p}}(\boldsymbol{x},t_n)=\frac{\boldsymbol{p}_{obs}\left(\boldsymbol{x},t_n\right)\odot\boldsymbol{p}_{pred}\left(\boldsymbol{x},t_n\right)}{\int\boldsymbol{p}_{obs}\left(\boldsymbol{x},t_n\right)\odot\boldsymbol{p}_{pred}\left(\boldsymbol{x},t_n\right)d\boldsymbol{x}}. \tag{5.62}$$

The updated state and covariance for the filter can be computed using the updated probability density:

$$\hat{\boldsymbol{x}}_{n|n} = \int \boldsymbol{x}^{\mathrm{T}} \hat{\boldsymbol{p}} d\boldsymbol{x} \tag{5.63}$$

and

$$\boldsymbol{P}_{n|n}^{xx} = \int \left(\boldsymbol{x} - \hat{\boldsymbol{x}}_{n|n}\right)\left(\boldsymbol{x} - \hat{\boldsymbol{x}}_{n|n}\right)^{\mathrm{T}} \hat{\boldsymbol{p}} d\boldsymbol{x}. \tag{5.64}$$

All discretization for the FTCS OBF algorithm were performed based on evidence gathered from experiments, similar to the one described in Section 5.3.2. All spatial components and their velocities were tested using separate linear OBFs. Discretization increments were reduced until each filter's accuracy for the two NRI trajectories (radial inbound and Dutch roll) was in the family or superior to previously documented D-D filter algorithms. The resulting discretization of the six dimensional space is presented in Table 17.

Table 17: FTCS OBF discretization table

Parameter	Units	Measurement Accuracy	Grid Min	Grid Max	Grid Increment
Range	dmi	0.15	1	50	0.075
Range Rate	knots	20	-1775	-710	10
Bearing	deg	0.3	-2	2	0.15
Bearing Rate	deg/s	N/A	-1.5	1.5	0.15
Elevation	deg	0.3	-2	2	0.15
Elevation Rate	deg/s	N/A	-1.5	1.5	0.15

5.4.Development of Practical EPC OBFs

5.4.1. EPC Spherical OBF Prediction

In the course of developing the finite difference EPC spherical OBF, it was found that the curse of dimensionality impacted our ability to utilize [36]. Transition matrix ***B***,

which could be conveniently generated offline to save runtime for lower-dimensional problems, grew too large to be held in memory for the computers executing the algorithm. This occurred despite leveraging MATLAB's sparse matrix storage methods [83]. In addition, the execution time predictably suffered to the extent that even if sufficient memory was available, the algorithm could never be executed in real time. Naval vessels that may benefit most from using an EPC target tracking algorithm are unlikely to have a computer resources with near-limitless memory; therefore, it is necessary to acknowledge these constraints. Modifications were made to the algorithm to support a more practical solution. To make those modifications, however, it is helpful to reference the finite difference equation given for the full EPC spherical FPKfe, so it will be developed in this section.

Recall that the FPKfe for the EPC spherical model is

$$\begin{aligned}\frac{\partial p}{\partial t} = \frac{\sigma_w^2}{2}&\left[\frac{\partial^2 p}{\partial \dot{r}^2} + \left(\frac{1}{r\cos\varphi}\right)^2 \frac{\partial^2 p}{\partial \dot{\theta}^2} + \left(\frac{1}{r}\right)^2 \frac{\partial^2 p}{\partial \dot{\varphi}^2}\right] - \\ &\left[\dot{r}\frac{\partial p}{\partial r} + (1-k)r\left(\dot{\varphi}^2 + \dot{\theta}^2\cos^2\varphi\right)\frac{\partial p}{\partial \dot{r}}\right] - \\ &\left[\dot{\theta}\frac{\partial p}{\partial \theta} + (2-k)\dot{\theta}\left(\dot{\varphi}\tan\varphi - \frac{\dot{r}}{r}\right)\frac{\partial p}{\partial \dot{\theta}}\right] - \\ &\left[\dot{\varphi}\frac{\partial p}{\partial \varphi} - \left((1-k)\dot{\theta}^2\cos\varphi\sin\varphi + \frac{(2-k)\dot{\varphi}\dot{r}}{r}\right)\frac{\partial p}{\partial \dot{\varphi}}\right].\end{aligned} \tag{5.65}$$

Excluding time, this PDE had six dimensions: range, range rate, bearing, bearing rate, elevation, and elevation rate. All six dimensions were divided into appropriate grid sizes, which are defined in Table 17. The grids are indexed using the same matrix $\boldsymbol{J}$ from (5.54) ; however, in this problem, the index matrix has six columns (one for each dimension):

$$\boldsymbol{J} = [\underset{r}{\boldsymbol{j}_1} \quad \underset{\dot{r}}{\boldsymbol{j}_2} \quad \underset{\theta}{\boldsymbol{j}_3} \quad \underset{\dot{\theta}}{\boldsymbol{j}_4} \quad \underset{\varphi}{\boldsymbol{j}_5} \quad \underset{\dot{\varphi}}{\boldsymbol{j}_6}]. \tag{5.66}$$

For each node in the finite grid, these six dimensions have a "forward" and a "backward" neighbor node, as well as a central node. The terms of $\boldsymbol{B}$ are defined in terms of these nodes, and are listed below.

1. Central node (main diagonal of $\boldsymbol{B}$, all indices in a given row of $\boldsymbol{J}$ match).

$$\tilde{\boldsymbol{B}}_1(\boldsymbol{J}) = -\Delta t q^2 \left(\frac{1}{\left(\Delta \dot{r}\right)^2} + \frac{1}{\left(\Delta \dot{\theta}\right)^2 \left(r^{(j_1)} \cos \varphi^{(j_5)}\right)^2} + \frac{1}{\left(\Delta \dot{\varphi}\right)^2 \left(r^{(j_1)}\right)^2} \right) \tag{5.67}$$

2. Forward range (range index $j_1 = j_1$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_2(\boldsymbol{J}) = -\frac{\Delta t \left(\dot{r}^{(j_2)}\right)}{2\Delta r} \tag{5.68}$$

3. Backward range (range index $j_1 = j_1$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_3(\boldsymbol{J}) = \frac{\Delta t \left(\dot{r}^{(j_2)}\right)}{2\Delta r} \tag{5.69}$$

4. Forward range rate (range rate index $j_2 = j_2$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_4(\boldsymbol{J}) = \frac{\Delta t q^2}{2} \left(\frac{1}{\left(\Delta \dot{r}\right)^2} - \frac{(1-k) r \left(\left(\dot{\varphi}^{(j_6)}\right)^2 + \left(\dot{\theta}^{(j_4)}\right)^2 \cos^2 \varphi^{(j_5)} \right)}{\Delta \dot{r}} \right) \tag{5.70}$$

5. Backward range rate (range rate index $j_2 = j_2$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_5(\boldsymbol{J}) = \frac{\Delta t q^2}{2} \left(\frac{1}{\left(\Delta \dot{r}\right)^2} + \frac{(1-k) r \left(\left(\dot{\varphi}^{(j_6)}\right)^2 + \left(\dot{\theta}^{(j_4)}\right)^2 \cos^2 \varphi^{(j_5)} \right)}{\Delta \dot{r}} \right) \tag{5.71}$$

6. Forward bearing (bearing index $j_3 = j_3$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_6(\boldsymbol{J}) = -\frac{\Delta t \left(\dot{\theta}^{(j_4)}\right)}{2\Delta \theta} \tag{5.72}$$

7. Backward bearing (bearing index $j_3 = j_3$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_7(\boldsymbol{J}) = \frac{\Delta t\left(\dot{\theta}^{(j_4)}\right)}{2\Delta\theta} \tag{5.73}$$

8. Forward bearing rate (bearing rate index $j_4 = j_4$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_8(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\theta}\right)^2\left(r^{(j_1)}\cos\varphi^{(j_5)}\right)^2} - \frac{(2-k)\dot{\theta}\left(\dot{\varphi}^{(j_6)}\tan\varphi^{(j_5)} - \dfrac{\dot{r}^{(j_2)}}{r^{(j_1)}}\right)}{\Delta\dot{\theta}}\right). \tag{5.74}$$

9. Backward bearing rate (bearing rate index $j_4 = j_4$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_9(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\theta}\right)^2\left(r^{(j_1)}\cos\varphi^{(j_5)}\right)^2} + \frac{(2-k)\dot{\theta}\left(\dot{\varphi}^{(j_6)}\tan\varphi^{(j_5)} - \dfrac{\dot{r}^{(j_2)}}{r^{(j_1)}}\right)}{\Delta\dot{\theta}}\right). \tag{5.75}$$

10. Forward elevation (elevation index $j_5 = j_5$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_{10}(\boldsymbol{J}) = -\frac{\Delta t\left(\dot{\varphi}^{(j_6)}\right)}{2\Delta\varphi}. \tag{5.76}$$

11. Backward elevation (elevation index $j_5 = j_5$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_{11}(\boldsymbol{J}) = \frac{\Delta t\left(\dot{\varphi}^{(j_6)}\right)}{2\Delta\varphi}. \tag{5.77}$$

12. Forward elevation rate (elevation rate index $j_6 = j_6$'+1, all other indices match):

$$\begin{aligned}\tilde{\boldsymbol{B}}_{12}(\boldsymbol{J}) = \frac{\Delta t}{2}\Bigg(&\frac{q^2}{\left(\Delta\dot{\varphi}\right)^2\left(r^{(j_1)}\right)^2} + \\ &\frac{1}{\Delta\dot{\varphi}}\left(\frac{(1-k)\left(\dot{\theta}^{(j_4)}\right)^2\sin 2\varphi^{(j_5)}}{2} + \frac{(2-k)\dot{\varphi}^{(j_6)}\dot{r}^{(j_2)}}{r^{(j_1)}}\right)\Bigg).\end{aligned} \tag{5.78}$$

13. Backward elevation rate (elevation rate index $j_6 = j_6$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_{13}(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\varphi}\right)^2 \left(r^{(j_1)}\right)^2} - \frac{1}{\Delta\dot{\varphi}}\left(\frac{\left(1-k\right)\left(\dot{\theta}^{(j_4)}\right)^2 \sin 2\varphi^{(j_5)}}{2} + \frac{\left(2-k\right)\dot{\varphi}^{(j_6)}\dot{r}^{(j_2)}}{r^{(j_1)}} \right) \right). \tag{5.79}$$

The sum of the matrices

$$\boldsymbol{B} = \sum_{i=1}^{13} \tilde{\boldsymbol{B}}_{\boldsymbol{i}} \tag{5.80}$$

delivers the transition matrix $\boldsymbol{B}$ required to execute the FTCS finite-difference method (5.52). In the next two sections, the storage size of $\boldsymbol{B}$ and the computation time of (5.52) are reduced by treating the discretized probability vector as sparse, leveraging GPU capabilities, and modifying the finite-difference equations by adding faster "companion" filters in some dimensions.

5.4.2. Planar EPC OBF with Separate Elevation Filter

The six-dimensional FTCS EPC spherical OBF may not be tractable using the currently available computing capabilities, but it is still possible to execute the OBF algorithm laid out in Section 5.3.3 if we simplify the problem to include range, range rate, bearing, and bearing rate. This filtering approach can be used for incoming targets, whose maneuvers tend to be isolated in the horizontal plane relative to the sensor. In addition, to prevent loss of elevation information, a separate elevation filter can be maintained. By removing the elevation and elevation rate components from the transition matrix, we successfully reduced the size of $\boldsymbol{B}$ to the extent that simulations could be completed for comparison with other target tracking algorithms. The greatest gains in the algorithm speed were achieved by leveraging the computer's GPU using MATLAB's

gpuArray functions [80]. This modification resulted in an approximately one order of magnitude speed improvement for the planar EPC OBF as well as the spherical EPC OBF covered in Section 5.4.3. Transition matrix B was stored using MATLAB's sparse matrix tools [83].

The planar EPC OBF was composed of two separate OBFs using two separate FPKfes. The first OBF concerns the range, range rate, bearing, and bearing rate, with the index matrix:

$$\boldsymbol{J} = [\underset{r}{\boldsymbol{j}_1} \quad \underset{\dot{r}}{\boldsymbol{j}_2} \quad \underset{\theta}{\boldsymbol{j}_3} \quad \underset{\dot{\theta}}{\boldsymbol{j}_4}]. \tag{5.81}$$

The propagation of the planar OBF state PDF is represented by the PDE

$$\begin{aligned} \frac{\partial p_{pl}\left(r,\dot{r},\theta,\dot{\theta},t\right)}{\partial t} = \frac{q^2}{2}\left[\frac{\partial^2 p_{pl}}{\partial \dot{r}^2} + \left(\frac{1}{r}\right)^2 \frac{\partial^2 p_{pl}}{\partial \dot{\theta}^2}\right] - \\ \left[\dot{r}\frac{\partial p_{pl}}{\partial r} + \left(1-k\right)r\dot{\theta}^2\frac{\partial p_{pl}}{\partial \dot{r}}\right] - \\ \left[\dot{\theta}\frac{\partial p_{pl}}{\partial \theta} - \frac{\left(2-k\right)\dot{\theta}\dot{r}}{r}\frac{\partial p_{pl}}{\partial \dot{\theta}}\right] \end{aligned} \tag{5.82}$$

Because a separate elevation filter runs parallel to the planar filter, the previously updated estimates of elevation $\hat{\varphi}$ and $\hat{\dot{\varphi}}$ can be leveraged to improve the estimation of the remaining components. Therefore, these terms are included in the finite difference calculations where relevant.

1. Central node (main diagonal of $\boldsymbol{B}$, all indices in a given row of $\boldsymbol{J}$ match):

$$\tilde{\boldsymbol{B}}_1\left(\boldsymbol{J}\right) = -\Delta t q^2\left(\frac{1}{\left(\Delta\dot{r}\right)^2} + \frac{1}{\left(\Delta\dot{\theta}\right)^2\left(r^{(j_1)}\cos\varphi^{(j_5)}\right)^2}\right) \tag{5.83}$$

2. Forward range (range index $j_1 = j_1' + 1$, all other indices match):

$$\tilde{\boldsymbol{B}}_2(\boldsymbol{J}) = -\frac{\Delta t\left(\dot{r}^{(j_2)}\right)}{2\Delta r} \tag{5.84}$$

3. Backward range (range index $j_1 = j_1$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_3(\boldsymbol{J}) = \frac{\Delta t\left(\dot{r}^{(j_2)}\right)}{2\Delta r} \tag{5.85}$$

4. Forward range rate (range rate index $j_2 = j_2$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_4(\boldsymbol{J}) = \frac{\Delta t q^2}{2}\left(\frac{1}{\left(\Delta\dot{r}\right)^2} - \frac{(1-k)r\left(\left(\hat{\dot{\varphi}}\right)^2 + \left(\dot{\theta}^{(j_4)}\right)^2\cos^2\hat{\varphi}\right)}{\Delta\dot{r}}\right) \tag{5.86}$$

5. Backward range rate (range rate index $j_2 = j_2$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_5(\boldsymbol{J}) = \frac{\Delta t q^2}{2}\left(\frac{1}{\left(\Delta\dot{r}\right)^2} + \frac{(1-k)r\left(\left(\hat{\dot{\varphi}}\right)^2 + \left(\dot{\theta}^{(j_4)}\right)^2\cos^2\hat{\varphi}\right)}{\Delta\dot{r}}\right) \tag{5.87}$$

6. Forward bearing (bearing index $j_3 = j_3$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_6(\boldsymbol{J}) = -\frac{\Delta t\left(\dot{\theta}^{(j_4)}\right)}{2\Delta\theta} \tag{5.88}$$

7. Backward bearing (bearing index $j_3 = j_3$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_7(\boldsymbol{J}) = \frac{\Delta t\left(\dot{\theta}^{(j_4)}\right)}{2\Delta\theta} \tag{5.89}$$

8. Forward bearing rate (bearing rate index $j_4 = j_4$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_8(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\theta}\right)^2\left(r^{(j_1)}\cos\hat{\varphi}\right)^2} - \frac{(2-k)\dot{\theta}\left(\hat{\dot{\varphi}}\tan\hat{\varphi} - \dfrac{\dot{r}^{(j_2)}}{r^{(j_1)}}\right)}{\Delta\dot{\theta}}\right). \tag{5.90}$$

9. Backward bearing rate (bearing rate index $j_4 = j_4$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_9(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\theta}\right)^2\left(r^{(j_1)}\cos\hat{\varphi}\right)^2} + \frac{(2-k)\dot{\theta}\left(\hat{\dot{\varphi}}\tan\hat{\varphi} - \dfrac{\dot{r}^{(j_2)}}{r^{(j_1)}}\right)}{\Delta\dot{\theta}} \right). \tag{5.91}$$

The transition matrix is obtained by summing the above matrices.

$$\boldsymbol{B} = \sum_{i=1}^{9} \tilde{\boldsymbol{B}}_i \tag{5.92}$$

The second filter is a simple OBF for elevation and elevation rate, similar to the filter examined in Section 5.3.2. Elevation OBF propagation can be performed using the PDE, which includes the previous estimate of the target range $\hat{r}$ to properly scale the diffusion intensity q:

$$\frac{\partial p}{\partial t} = \frac{q^2}{2\left(\hat{r}\right)^2}\left[\frac{\partial^2 p}{\partial\dot{\varphi}^2}\right] - \left[\dot{\varphi}\frac{\partial p}{\partial\varphi}\right]. \tag{5.93}$$

The transition matrix for the elevation filter is

$$\begin{aligned}
\boldsymbol{B}_{jj'} = &\left[\frac{q^2\Delta t}{\left(\hat{r}\Delta\dot{\varphi}\right)^2}\right]\delta\left(j_1, j_1'\right)\delta\left(j_2, j_2'\right) + \\
&\left[-\frac{q^2\Delta t}{2\left(\hat{r}\Delta\dot{\varphi}\right)^2}\right]\delta\left(j_1, j_1'\right)\delta\left(j_2, j_2'+1\right) + \\
&\left[-\frac{q^2\Delta t}{2\left(\hat{r}\Delta\dot{\varphi}\right)^2}\right]\delta\left(j_1, j_1'\right)\delta\left(j_2, j_2'-1\right) \\
&\left[\frac{\dot{\varphi}^{(j_2)}\Delta t}{2\left(\Delta\varphi\right)}\right]\delta\left(j_1, j_1'+1\right)\delta\left(j_2, j_2'\right) + \\
&\left[-\frac{\dot{\varphi}^{(j_2)}\Delta t}{2\left(\Delta\varphi\right)}\right]\delta\left(j_1, j_1'-1\right)\delta\left(j_2, j_2'\right),
\end{aligned} \tag{5.94}$$

with indexing matrix

$$\boldsymbol{J} = \underset{\varphi}{[\boldsymbol{j}_1} \quad \underset{\dot{\varphi}}{\boldsymbol{j}_2}]. \tag{5.95}$$

The planar FTCS EPC OBF algorithm runs using trajectories examined in previous performance studies. The main challenge concerning the computation speed for this algorithm arose because of the large extent of the range grid. Targets approaching from 50 nautical miles to approximately zero nautical miles with a measurement accuracy on the order of tenths of miles result in a very large grid (see Table 17). In the next section, another filter obtained by splitting the dimensions into range and angle components is presented. This algorithm eliminates the large range grid problem by solving a PDE with only the angle and angular velocity components.

5.4.3. EPC OBF on the Sphere with Separate Range Filter

Some memory storage and computation speed gains were made by separating the range, range rate, bearing, and bearing rate from the elevation and elevation rate in Section 5.4.2. However, it was found that even greater speed performance gains could be achieved by separating the range and the range rate from the angular components and their velocities. The range and range rate filters provided the necessary estimates to be used with the angular filter. The estimated range $\hat{r}$ and range rate $\hat{\dot{r}}$ from the previous filter update step are used to calculate the transition matrix elements. The spherical (S2) OBF PDE used to propagate the conditional state distribution is

$$\begin{aligned}\frac{\partial p_{S2}\left(\theta,\dot{\theta},\varphi,\dot{\varphi},t\right)}{\partial t} = \frac{q^2}{2}&\left[\left(\frac{1}{\hat{r}\cos\varphi}\right)^2\frac{\partial^2 p_{S2}}{\partial\dot{\theta}^2}+\left(\frac{1}{\hat{r}}\right)^2\frac{\partial^2 p_{S2}}{\partial\dot{\varphi}^2}\right]- \\ &\left[\dot{\theta}\frac{\partial p_{S2}}{\partial\theta}+\left(2-k\right)\dot{\theta}\left(\dot{\varphi}\tan\varphi-\frac{\hat{\dot{r}}}{\hat{r}}\right)\frac{\partial p_{S2}}{\partial\dot{\theta}}\right]- \\ &\left[\dot{\varphi}\frac{\partial p_{S2}}{\partial\varphi}-\left(\left(1-k\right)\dot{\theta}^2\cos\varphi\sin\varphi+\frac{\left(2-k\right)\dot{\varphi}\hat{\dot{r}}}{\hat{r}}\right)\frac{\partial p_{S2}}{\partial\dot{\varphi}}\right].\end{aligned} \tag{5.96}$$

The indexing matrix used for the angular filter is

$$\boldsymbol{J} = [\underset{\theta}{\boldsymbol{j}_3} \quad \underset{\dot{\theta}}{\boldsymbol{j}_4} \quad \underset{\varphi}{\boldsymbol{j}_5} \quad \underset{\varphi}{\boldsymbol{j}_6}]. \tag{5.97}$$

The filter transition matrix components are as follows. Once again, the transition matrix can be computed using Equation (5.92).

1. Central node (main diagonal of $\boldsymbol{B}$, all indices in a given row of $\boldsymbol{J}$ match):

$$\tilde{\boldsymbol{B}}_1(\boldsymbol{J}) = -\Delta t q^2 \left(\frac{1}{\left(\Delta\dot{\theta}\right)^2 \left(\hat{r}\cos\varphi^{(j_5)}\right)^2} + \frac{1}{\left(\Delta\dot{\varphi}\right)^2 \left(\hat{r}\right)^2} \right). \tag{5.98}$$

2. Forward in bearing (bearing index $j_3 = j_3$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_6(\boldsymbol{J}) = -\frac{\Delta t\left(\dot{\theta}^{(j_4)}\right)}{2\Delta\theta} \tag{5.99}$$

3. Backward in bearing (bearing index $j_3 = j_3$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_7(\boldsymbol{J}) = \frac{\Delta t\left(\dot{\theta}^{(j_4)}\right)}{2\Delta\theta} \tag{5.100}$$

4. Forward in bearing rate (bearing rate index $j_4 = j_4$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_8(\boldsymbol{J}) = \frac{\Delta t}{2} \left(\frac{q^2}{\left(\Delta\dot{\theta}\right)^2 \left(\hat{r}\cos\varphi^{(j_5)}\right)^2} - \frac{(2-k)\dot{\theta}\left(\dot{\varphi}^{(j_6)}\tan\varphi^{(j_5)} - \dfrac{\dot{\hat{r}}}{\hat{r}}\right)}{\Delta\dot{\theta}} \right). \tag{5.101}$$

5. Backward bearing rate (bearing rate index $j_4 = j_4$'-1, all other indices match):

$$\tilde{\boldsymbol{B}}_9(\boldsymbol{J}) = \frac{\Delta t}{2} \left(\frac{q^2}{\left(\Delta\dot{\theta}\right)^2 \left(\hat{r}\cos\varphi^{(j_5)}\right)^2} + \frac{(2-k)\dot{\theta}\left(\dot{\varphi}^{(j_6)}\tan\varphi^{(j_5)} - \dfrac{\dot{\hat{r}}}{\hat{r}}\right)}{\Delta\dot{\theta}} \right). \tag{5.102}$$

6. Forward elevation (elevation index $j_5 = j_5$'+1, all other indices match):

$$\tilde{\boldsymbol{B}}_{10}(\boldsymbol{J}) = -\frac{\Delta t\left(\dot{\varphi}^{(j_6)}\right)}{2\Delta\varphi}. \tag{5.103}$$

7. Backward elevation (elevation index $j_5 = j_5$'-1, all other indices match):

$$\boldsymbol{B}_{11}(\boldsymbol{J}) = \frac{\Delta t\left(\dot{\varphi}^{(j_6)}\right)}{2\Delta\varphi}. \tag{5.104}$$

8. Forward elevation rate (elevation rate index $j_6 = j_6$'+1, all other indices match):

$$\boldsymbol{B}_{12}(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\varphi}\right)^2\left(\hat{r}\ \right)^2} + \frac{1}{\Delta\dot{\varphi}}\left(\frac{\left(1-k\right)\left(\dot{\theta}^{(j_4)}\right)^2 \sin 2\varphi^{(j_5)}}{2} + \frac{\left(2-k\right)\dot{\varphi}^{(j_6)}\hat{\dot{r}}}{\hat{r}}\right)\right). \tag{5.105}$$

9. Backward elevation rate (elevation rate index $j_6 = j_6$'-1, all other indices match):

$$\boldsymbol{B}_{13}(\boldsymbol{J}) = \frac{\Delta t}{2}\left(\frac{q^2}{\left(\Delta\dot{\varphi}\right)^2\left(\hat{r}\ \right)^2} - \frac{1}{\Delta\dot{\varphi}}\left(\frac{\left(1-k\right)\left(\dot{\theta}^{(j_4)}\right)^2 \sin 2\varphi^{(j_5)}}{2} + \frac{\left(2-k\right)\dot{\varphi}^{(j_6)}\hat{\dot{r}}}{\hat{r}}\right)\right). \tag{5.106}$$

5.5. Performance Study of C-D and D-D Filters

5.5.1. EPC OBF Study Goals

The OBFs developed in this dissertation are a proof-of-concept to demonstrate that C-D filtering algorithms that track the evolution of a target state over time are feasible for a target tracking system. Given the findings of previous studies (see Section 2.6.6 and Section 3.3.6), the non-NRI targets shall not be examined, as it is already known that filters using only the EPC model will not perform well against these targets.

In addition, they would require simulation over substantially larger finite difference grids due to their large variation in range and angle. The initial thrust of our study of a family of EPC filters [45] focused only on NRI targets, so the OBF prototype filters were also compared with their D-D predecessors against the set of NRI targets.

The goals of this study were to establish the performance capabilities of the C-D OBFs developed in Sections 5.4.2 and 5.4.3. The planar EPC OBF (EPC POBF) and spherical EPC OBF (EPC S2OBF) have their error performance compared with the family of D-D EPC filters presented in [45] and [34]. It is expected that the computational burden set by the large finite-difference grids required for the propagation of the state probability density will result in increased simulation runtimes. However, if the error performance is excellent compared to the other EPC filters, then the EPC OBFs merit further development to reduce execution times.

5.5.2. Radially Inbound Target

For the radially inbound target, both EPC OBFs performed quite well, particularly in bearing RMSE, as shown in Figure 89. In fact, both filters outperform the EPC BPF in bearing RMSE and do not feature any significant divergence behavior near the end of the trajectory, as many other filters do. For elevation (see Figure 90), the EPC S2OBF on the sphere performs better than the planar EPC OBF as expected, because the planar EPC OBF's separate elevation filter is not designed to retain the endpoint constraint feature. The EPC BPF was the second-best filter for elevating RMSE performance.

The box plots for the errors in all dimensions are shown in Figure 91 and Figure 92. These plots continue to emphasize that both OBFs offer benefits over their D-D

counterparts in terms of bearing and bearing rate, whereas only the EPC S2OBF retains this advantage for elevation-error performance. It is also notable that both OBFs feature much more compact error data, with few or no outliers, compared to the other filters.

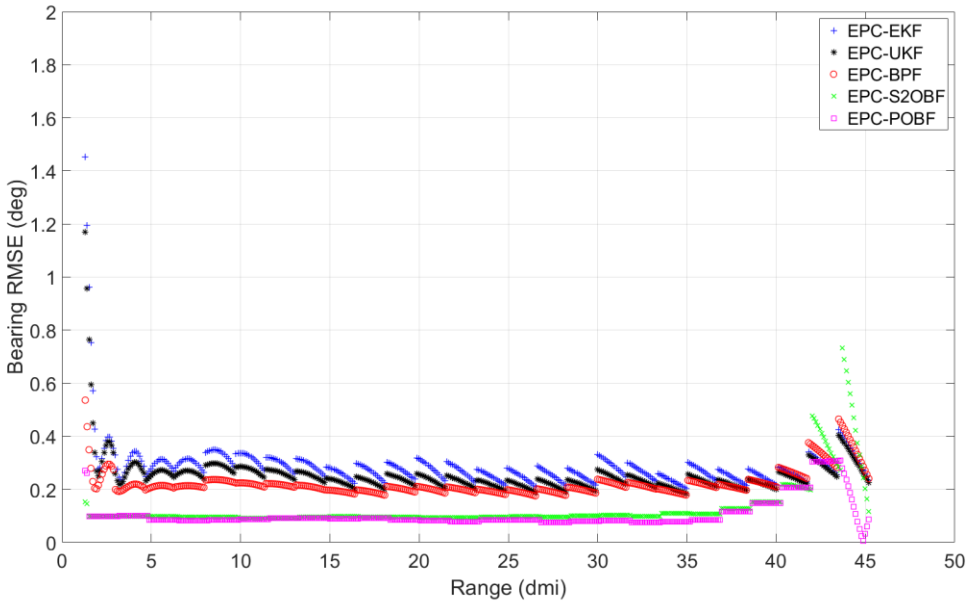

Figure 89: OBF performance study – bearing RMSE radially inbound trajectory.

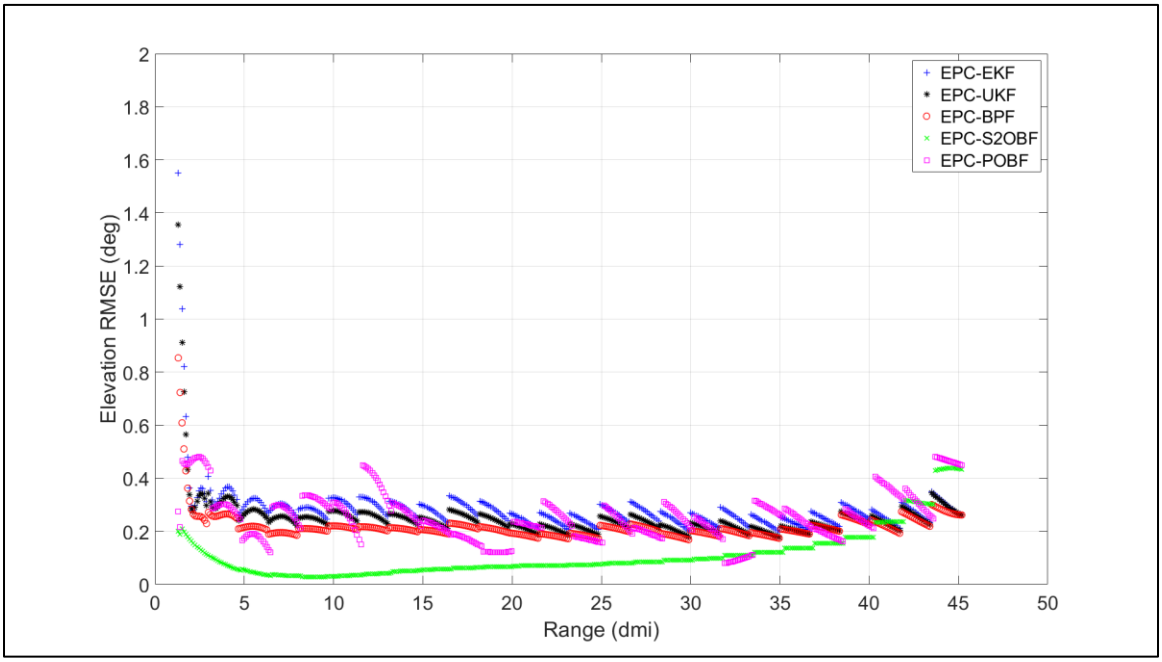

Figure 90: OBF performance study – elevation RMSE for radially inbound scenario

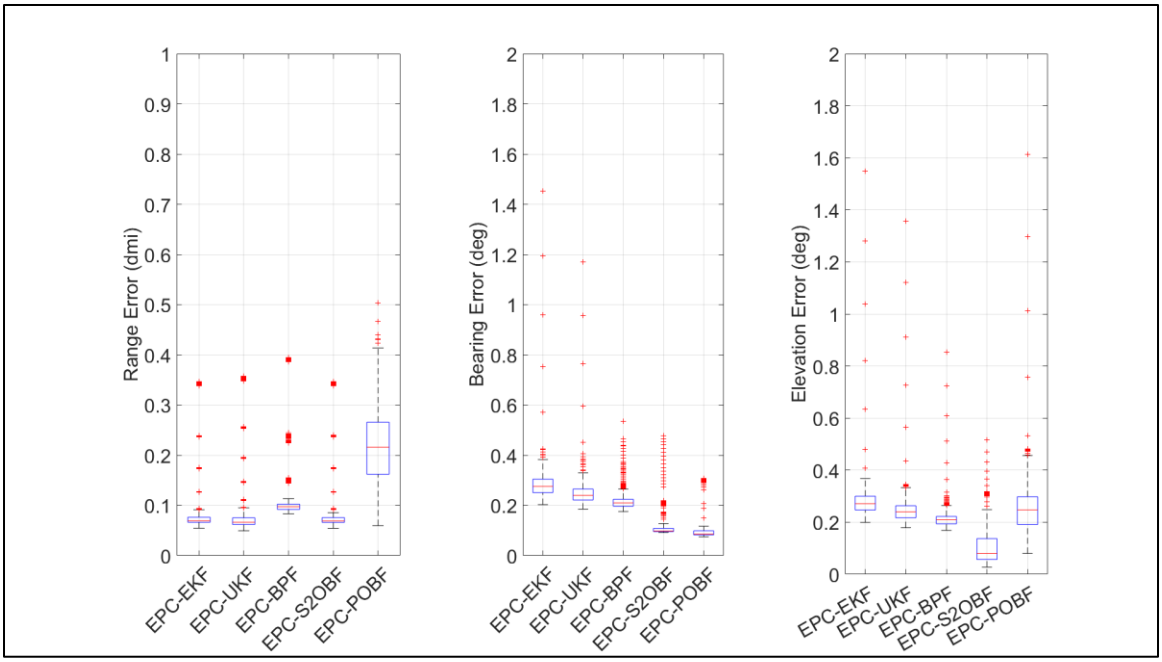

Figure 91: OBF performance study – position error box for radially inbound scenario

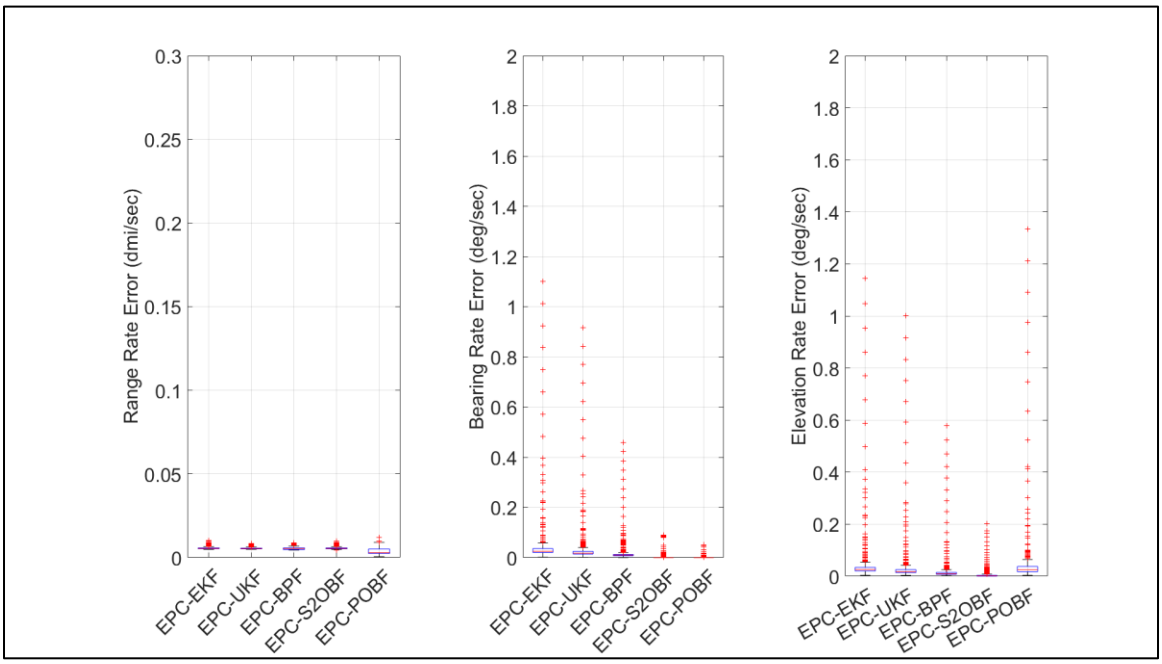

Figure 92: OBF performance study – velocity error box plots for radially inbound scenario

5.5.3. Dutch Roll Target

The EPC OBFs continued to demonstrate excellent error performance against the challenging Dutch roll target. Both filters once again hold the advantage in terms of bearing RMSE, while the planar filter suffers some losses for elevation RMSE the

spherical OBF exhibits superior bearing and elevation RMSE performance during both the full NRI inbound period of the trajectory and the weaving period, as seen in Figure 93 and Figure 94.

The box plots show a story similar to the radially inbound trajectory. The EPC BPF and both OBFs were highly competitive for the Dutch roll target in all dimensions, as shown in Figure 95 and Figure 96. The spherical EPC OBF has superior elevation RMSE results, although not by as wide a margin as for the radially inbound trajectory.

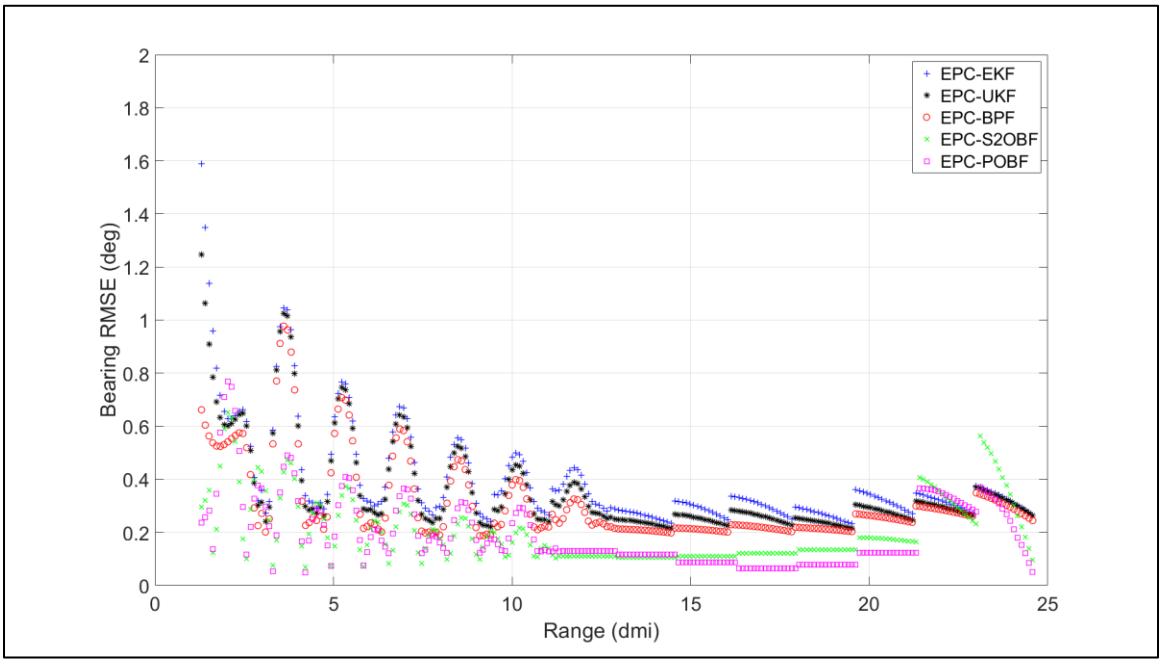

Figure 93: OBF performance study – bearing RMSE for Dutch roll trajectory

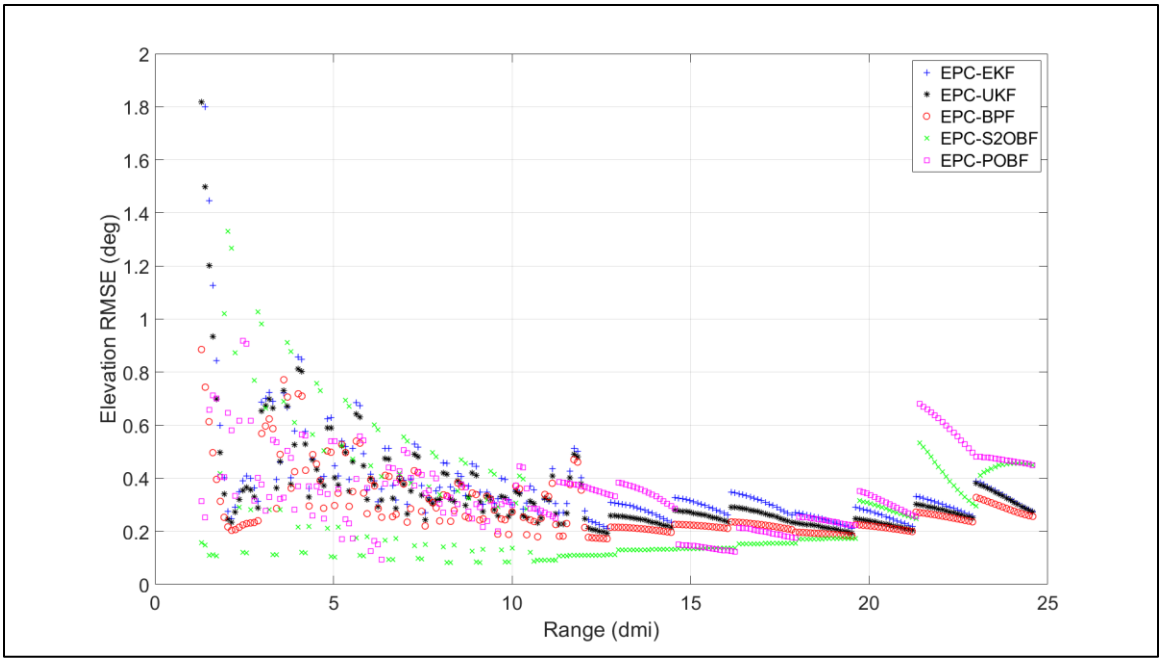

Figure 94: OBF performance study – elevation RMSE for Dutch roll trajectory

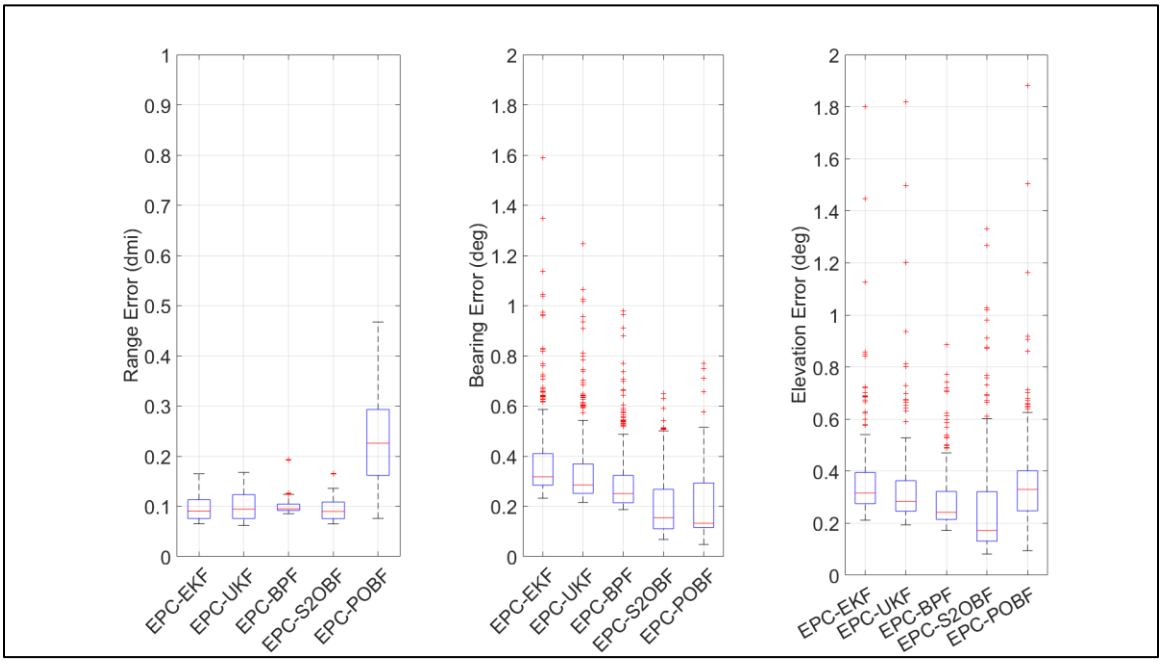

Figure 95: OBF performance study – position error box plots for Dutch roll trajectory

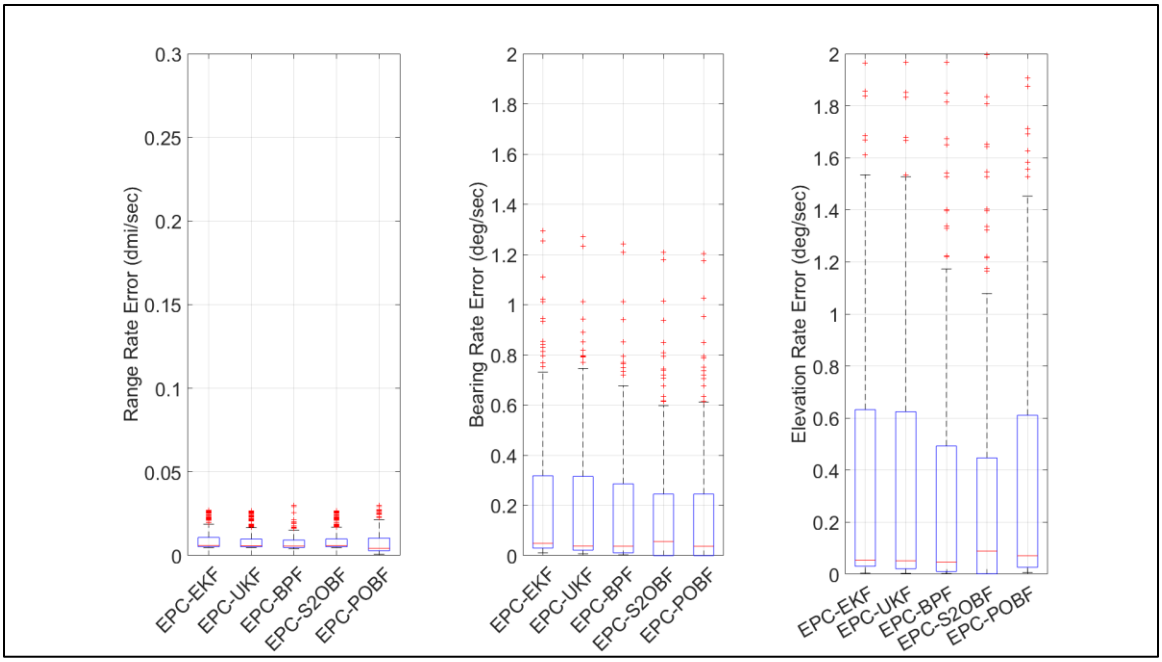

Figure 96: OBF performance study – velocity error box plots for Dutch roll trajectory

5.5.4. EPC OBF Study Overall Performance Assessment

Both OBFs perform admirably in terms of error performance, beating out their D-D peers in most cases for NRI targets, as seen in Table 18. With further development, both filters could potentially add significant value to a self-defense combat system. Current limitations include coarse grids that are required to prevent excessive memory burden and increase computation speed. With a smaller or dynamically updated grid, the OBFs can potentially achieve even better performance than those demonstrated in this study. However, any modifications made to the OBFs will also have to address the slow computation times observed in these experiments, perhaps by leveraging a larger set of CPUs to compute solutions in different zones around the sensor origin. The planar EPC OBF, with its large range and range rate grids, requires considerably more modifications to form a functional filter than the spherical EPC OBF, whose run times are large but not nearly as large as its fellow OBF, as shown in Table 19.

Further development of the OBFs appears worthwhile based on these results, as RMSE performance gains of up to 10% are achievable with these algorithms. If upgrades to these algorithms can sufficiently address the runtime and memory challenges, then perhaps the full EPC OBF presented in Section 5.4.1 can replace both filters. The OBF algorithm proof of concept was successful for a set of spherical coordinates and nonlinear dynamics featured in the EPC model.

Table 18: Overall RMSE performance for the family of EPC filters

Scenario	Filter Name	RMS Error Overall					
		Range (dmi)	Range Rate (dmi/s)	Bearing (deg)	Bearing Rate (deg/s)	Elevation (deg)	Elevation Rate (deg/s)
Radial Inbound	EPC EKF	0.0934	0.0751	0.2875	0.0516	0.2829	0.0518
	EPC UKF	0.0944	0.0740	0.2527	0.0391	0.2507	0.0404
	EPC BPF	0.1158	0.0755	0.2230	0.0211	0.2180	0.0232
	EPC POBF	0.2359	0.0626	0.1111	0.0050	0.2668	0.0519
	EPC S2OBF	0.1080	0.0745	0.1296	0.0115	0.1184	0.0075
Dutch Roll	EPC EKF	0.0969	0.0974	0.3931	0.2070	0.3741	0.3743
	EPC UKF	0.1007	0.0964	0.3566	0.1934	0.3392	0.3599
	EPC BPF	0.1053	0.0903	0.3092	0.1729	0.2879	0.3301
	EPC POBF	0.2437	0.0872	0.2108	0.1546	0.3679	0.3483
	EPC S2OBF	0.1070	0.0942	0.2188	0.1599	0.2761	0.2996

Table 19: Runtime performance for the family of EPC filters

Scenario	Filter Name	Scenario Duration (sec)	Average Runtime (sec)	Runtime Ratio	Real Time?
Radial Inbound	EPC-EKF	117	0.00149	0.0016%	Yes
	EPC-UKF	117	0.00709	0.0082%	Yes
	EPC-BPF	117	11.39978	10.0062%	No
	EPC-POBF	117	6296.43	5381.57%	No
	EPC-S2OBF	117	275.22	235.23%	No
Dutch Roll	EPC-EKF	69	0.00082	0.0013%	Yes
	EPC-UKF	69	0.00383	0.0057%	Yes
	EPC-BPF	69	5.87340	9.2457%	No
	EPC-POBF	69	3527.41	5112.2%	No
	EPC-S2OBF	69	154.15	223.4%	No

6. Conclusion

6.1.Summary of Findings

The initial thrust of the EPC filtering research documented in this dissertation focused on using an endpoint-constrained EKF software upgrade to replace or supplement the legacy C-EKF algorithm commonly used in combat system target trackers. The flaws of the C-EKF model and the challenge of tracking an NRI target with a low update rate sensor were discussed in detail. By significantly reducing angular targeting errors, the EPC EKF provides a cost-effective alternative to potentially expensive hardware updates for legacy vessels. After studying the application of the EPC model to a wider set of D-D filters, C-D filters, and multiple model algorithms, it is important to revisit the original goals of this research.

The goals of the EPC studies were threefold: the first goal was to determine whether EPC filters offer improved tracking performance against maneuvering NRI targets when compared to legacy S-EKF and C-EKF. This goal was met, as the studies at the end of Chapters 2 through 5 demonstrated significant tracking error performance improvements for all EPC filters against the set of NRI targets. Ultimately, it was found that EPC OBF algorithms offered the best RMSE performance, followed closely by the EPC BPF.

The second goal was to address the performance degradation of EPC filters when the target is not NRI for its full duration. The case where the C-EKF and S-EKF both outperform the EPC filters is when the target is non-NRI. Replacing the legacy filters wholesale with any EPC algorithm would not be advisable because the non-NRI tracking performance is sacrificed to obtain superior NRI tracking performance. The flexible EPC

UKF IMM has been established as the best current candidate for implementation since it offers superior performance to the EPC EKF for NRI targets and does not have the computational burden found in the EPC BPF, or the EPC OBFs for that matter.

The third goal was to ensure that the algorithms could be executed in real time, as the operators of the combat system and the downstream algorithms using ASM track data are typically operating on a tight schedule. From the empirical algorithm timing results captured in each study, it was found that the EPC EKF and EPC UKF are firmly placed in the real-time algorithm category. With proper parallelization, compiled code, and fewer particles, the EPC BPF can conceivably become a real-time algorithm, although it may still not be suitable as a filter block within an IMM. The EPC OBFs run quite slowly, and despite their excellent error performance for NRI targets, they are not suitable for implementation in a combat system or as part of an IMM algorithm in its current form. Ultimately, the flexible EPC UKF IMM is the preferred algorithm for addressing all three goals based on this research.

6.2.Next Steps and Future Development Opportunities

Of the D-D filter algorithms examined, the EPC BPF was the strongest performer against NRI targets. Reducing the number of particles used results in a significant reduction in the run time, in exchange for a modest error performance loss. The next steps in EPC BPF development will include optimizing the EPC BPF for parallelization and rewriting the code in a compiled language such as C++ or FORTRAN to evaluate whether the algorithm becomes real time under these conditions. If successful, it would

be worth researching the optimal method to include particle filters in an IMM structure, as Lei et al. [66] did in their conference paper.

The EPC OBFs offer superior target tracking performance to all filters studied, and this was accomplished with fairly coarse dimensional grids to support a high-dimensional solution. The data storage requirements and run time are too long for a fixed grid implementation in a combat system. The primary focus of further studies will involve leveraging sparse tensor methods, such as the one explored by Fox et al. [36] to reduce the memory burden and increase the algorithm speed. Additional research will also revisit the adaptive grid approach studied by Holley [50] to enable a closer real-time implementation of the EPC OBF. With so much room for further performance gains in terms of error, run time, and data storage, OBFs will be the main focus of the author's continuing EPC filter research.

References

[1] V. Aidala and S. Hammel. “Utilization of modified polar coordinates for bearings-only tracking.” *IEEE Transactions on Automatic Control*, Vol. 28 (1983): pp. 283-294.

[2] G. Alcalay, C. Seren, G. Hardier, M. Delporte, and P. Goupil, “An Adaptive Extended Kalman Filter for Monitoring and Estimating Key Aircraft Flight Parameters,” *IFAC-PapersOnline*, Vol. 51, Art. No. 24, pp. 620–627, 2018.

[3] G. T. Aldrich, W. B. Krabill, “An Application of Kalman Technique to Aircraft and Missile Radar Tracking,” *AIAA Journal*, Vol. 11, Art. No. 7, pp. 932-938, 1973

[4] A. T. Alouani, and W. D. Blair, "Use of a Kinematic Constraint in Tracking Constant Speed, Maneuvering Targets”, *IEEE Transactions on Automatic Control.*, Jul. 1993, Vol. 38, Iss. 7, pp. 1107-1111, DOI: 10.1109/9.231465.

[5] F. Ahmed and R. Mandal, “Survey on the Kalman Filter and Related Algorithms,” *International Research Journal of Engineering and Technology*, Vol. 5, Iss. 5, 2018.

[6] “AN/SPS-48,” NARA & DVIDS Public Domain Archive, December 1990, Available: https://nara.getarchive.net/media/a-view-of-the-sps-48b-3-d-search-radar-antenna-on-the-nuclear-powered-aircraft-3cf89f.

[7] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian tracking,", *IEEE Transactions on Signal Processing*, Vol. 50, Art. No. 2, pp. 174-188, 2002, DOI: 10.1109/78.978374.

[8] V. Awasthi, and K. Raj, “A Survey of Kalman Filter Algorithms and Variants in State Estimation,” *Current Approaches in Science and Technology Research*, Vol. 15, pp. 1-14, 2021.

[9] B. Balaji, “Discrete Path Integral Filtering,” *Entropy*, 2009, DOI: 10.3390/e110300402.

[10] B. Balaji, “Optimal Solution of Finite Dimensional Filtering Problems via Solution of Linear ODEs,” *IEEE Radar Conference*, pp. 8, 2006.

[11] B. Balaji, “Universal, Continuous-Discrete Nonlinear Yau Filtering I: Affine, Linear State Model with State-Independent Diffusion Matrix,” *arXiv: Data Analysis, Statistics and Probability*, 2008.

[12] J. S. Baras, G. L. Blankenship, and W. E. Hopkins, “Existence, Uniqueness, and Asymptotic Behavior of Solutions to a Class of Zakai Equations with Unbounded Coefficients,” *IEEE Transactions on Automatic Control*, Vol 28, Art. No. 2, 1983.

[13] J. S. Baras, G. L. Blankenship, and S. K. Mitter, “Nonlinear Filtering of Diffusion Processes,” *Proceedings of 8th IFAC World Congress*, Vol 5, Art. No. 2, pp. 57-62, 1981.

[14] J. S. Baras, “Nonlinear Filtering of Diffusion Processes: Implementable Approximations to the Duncan-Mortensen-Zakai Equation,” *Proceedings of the 1982 IEEE International Symposium of Information Theory*, 1982.

[15] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons Inc., New York, USA, 2001.

[16] Y. Bar-Shalom, “Target Tracking and Data Fusion: How to Get the Most Out of Your Sensors and Make a Living Out of It, An Overview of Tracking Algorithms for Cluttered and Multitarget-Multisensor Environments,” Presentation, University of Connecticut, CT, USA, 2017.

[17] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, Storrs, CT, USA: YBS Publishing, 2011.
[18] W. D. Blair, G. A. Watson and T. R. Rice, "Interacting Multiple Model Filter for Tracking Maneuvering Targets in Spherical Coordinates," *IEEE Proceedings of the. SOUTHEASTCON '91*, 1991, Vol. 2, pp. 1055-1059, DOI: 10.1109/SECON.1991.14792.
[19] G. L. Blankenship and J. S. Baras, "Nonlinear Filtering of Diffusion Processes: A Generic Example," *18th IEEE Conference on Decision and Control*, 1979.
[20] H. P. Bloom, "An Efficient Filter for Abruptly Changing Systems," in *Proceedings of the 23rd IEEE Conference on Decision and Control*, Las Vegas, NV, 1984, 656-658
[21] C. Caginalp and G. Caginalp. "The Quotient of Normal Random Variables and Application to Asset Price Fat Tails." *Physica A: Statistical Mechanics and its Applications*, Vol. 499, pp. 457–471, 2018.
[22] G. Casella and R. L. Berger, *Statistical Inference*, Duxbury, 2002
[23] C. B. Chang, J. A. Tabaczynski, "Application of state estimation to target tracking", *IEEE Transactions on Automatic Control*, 1984, AC-29(2), pp. 98-109.
[24] W. Cheney and D. Kincaid, *Numerical Mathematics and Computing (6th ed.)*, Brooks/Cole Publishing Co., CA, USA, 2007.
[25] M. L. Ching, "Kalman Filter Tracking of a Ballistic Missile Using Forward Looking Infrared Measurements and Doppler Returns," M.S. Thesis, Air Force Institute of Technology, Ohio School of Engineering, Ohio, USA, 1992.
[26] D. Creal, "A Survey of Sequential Monte Carlo Methods for Economics and Finance," *Econometric Reviews*, Vol. 31, Art. No. 2, pp. 245-296, 2011
[27] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods*, Birkhäuser Boston, MA, USA, 2009.
[28] M. A. Davis and S. I. Marcus, *An Introduction to Nonlinear Filtering*, In: M. Hazewinkel, J. C. Willems (eds) *Stochastic Systems: The Mathematics of Filtering and Identification and Applications*, *NATO Advanced Study Institute Series*, Vol. 78, Springer, Dordrecht, 1981, DOI: 10.1007/978-94-009-8546-9_4.
[29] R. E. Day, "Coupling Dynamics in Aircraft: a Historical Perspective," *NASA Special Publication 532*, 1997.
[30] F. Daum, J. Huang, "Curse of Dimensionality and Particle Filters," *IEEE Aerospace Conference*, Vol.4, 2003.
[31] T. E. Duncan, "Probability Densities for Diffusion Processes with Applications to Nonlinear Filtering Theory", PhD dissertation, Stanford University, 1967.
[32] R. J. Elliott, S. Haykin, "A Zakai Equation Derivation of the Extended Kalman Filter," *Automatica*, Vol. 46, pp. 620-624, 2010, DOI: 10.1016/j.automatica.2010.01.006.
[33] M. E. Farmer, R.-L. Hsu, A. K. Jain, "Interacting Multiple Model (IMM) Kalman Filters for Robust High Speed Human Motion Tracking," *2002 International Conference on Pattern Recognition*, Vol. 2, pp. 20-23, 2002, DOI: 10.1109/ICPR.2002.1048226.
[34] K. R. Ford and A. J. Haug, "A Study of Endpoint-Constrained Nonlinear Tracking Filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, Art. No. 6, pp. 3952-3961, 2021, DOI: 10.1109/TAES.2021.3090919.

[35] K. R. Ford and A. J. Haug, "The Probability Density Function of Bearing Obtained from a Cartesian-to-Polar Transformation," *IEEE Access Regular Papers*, Vol. 10, pp. 32803-32809, 2022, DOI: 10.1109/ACCESS.2022.3161974.
[36] C. Fox, S. Dolgov, M. E. Morrison, and T. C. Molteno, "Grid Methods for Bayes-Optimal Continuous-Discrete Filtering and Utilizing a Functional Tensor Train Representation," *Inverse Problems in Science and Engineering*, Vol. 29, Art. No. 8, pp. 1199-1217, 2020, DOI: 10.1080/17415977.2020.1862109.
[37] A. F. Genovese, "The Interacting Multiple Model Algorithm for Accurate State Estimation of Maneuvering Targets," *Johns Hopkins APL Technical Digest*, Vol. 22, Art. No. 4, 2010, pp. 614-623.
[38] J. B. B. Gomes, "An Overview on Target Tracking Using Multiple Model Methods", M.S. Thesis, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal, 2008.
[39] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proceedings of Radar and Signal Processing*, Vol. 140, pp. 107-113. DOI: 10.1049/ip-f-2.1993.0015.
[40] I. S. Gradsheteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 7th Edition, Academic Press, 2007.
[41] E. Hajiramezanali, "Scalable Optimal Bayesian Classification of Single-Cell Trajectories Under Regulatory Model Uncertainty." *BMC genomics*, Vol. 20(6), pp. 435, 2019, DOI:10.1186/s12864-019-5720-3
[42] P. D. Hanlon and P. S. Maybeck, "Characterization of Kalman Filter Residuals in the Presence of Mismodeling," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, Art. No. 1, pp. 114-131, 2000.
[43] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*, John Wiley & Sons, Hoboken, NJ, USA, 2012.
[44] A. J. Haug, "A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes," *MITRE Technical Report*, Vol. MTR 05W0000004, 2005.
[45] A. J. Haug and K. R. Ford, "An end-point-constrained extended Kalman filter for tracking maneuvering near-radially inbound targets," *2016 19th International Conference on Information Fusion (FUSION)*, 2016, pp. 312-318.
[46] A. J. Haug, and L. M. Williams, "A Spherical Constant Velocity Model for Target Tracking in Three Dimensions," *IEEE Aerospace Conference*, pp. 1-15, 2012.
[47] N. J. Higham, "Computing Real Square Roots of a Real Matrix,"*Linear Algebra and its Applications*, Vol. 88-89, pp. 405-430, 1987.
[48] D. V. Hinkley, "On the Ratio of Two Correlated Normal Random Variables". *Biometrika*, Vol. 56 (1969): pp. 635-639.
[49] J. D. Hol, T. B. Schon, and F. Gustafsson, "On Resampling Algorithms for Particle Filters," *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, pp. 79-82, 2006.
[50] K. Holley, "Applications of the Multigrid Algorithm to Solving the Zakai Equation of Nonlinear Filtering with VLSI Implementation," PhD dissertation, Dept. of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA, 1986.
[51] J. Humpherys, P. Redd, and J. West. "A Fresh Look at the Kalman Filter," *SIAM Review*, Vol. 54, Iss. 4, pp. 801-823, 2012.

[52] R. J. Hyndman, A. B. Koehler, "Another look at measures of forecast accuracy". *International Journal of Forecasting*. Vol. 22, Iss. 4, pp. 679–688, 2006, DOI: 10.1016/j.ijforecast.2006.03.001.
[53] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academy Press, NY, 1970.
[54] R. Jinan and T. Raveendran, "Particle Filters for Multiple Target Tracking," *Procedia Technology*, Vol. 24, pp. 980-987, 2016.
[55] S. J. Julier, J. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems", *International Symposium for Aerospace/Defense Sensing, Simulation and Controls*, Vol. 3, 1997.
[56] S. J. Julier, and J. J. LaViola, "On Kalman Filtering with Nonlinear Equality Constraints", *IEEE Transactions on Signal Processing*, Vol. 55(6), pp. 2774-2784, 2007, DOI: 10.1109/TSP.2007.893949
[57] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, Vol. 92, Art. No. 3, pp. 401-422, 2004.
[58] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *ASME Transactions, Journal of Basic Engineering*, Mar 1960.
[59] R. E. Kalman and R. S. Bucy, "New Results in Linear Prediction and Filtering Theory," *ASME Transactions, Journal of Basic Engineering*, Vol. 83, pp. 95-108, 1961.
[60] R. Kandepu, L. Imsland, and B.A. Foss, "Constrained State Estimation Using the Unscented Kalman Filter", *16th Mediterranean Conference on Control and Automation*, pp. 1453-1458, 2008.
[61] T. Karvonen, "Stability of Linear and Non-Linear Kalman Filters" M.S. Thesis, University of Helsinki, December 2014.
[62] H. Khazraj, F. Faria da Silva and C. L. Bak, "A Performance Comparison Between Extended Kalman Filter and Unscented Kalman Filter in Power System Dynamic State Estimation," *2016 51st International Universities Power Engineering Conference (UPEC)*, pp. 1-6, 2016, DOI: 10.1109/UPEC.2016.8114125.
[63] P. Kotaru and K. Sreenath, "Variation Based Extended Kalman Filter on S2" *18th European Contr. Conf. (ECC)*, Naples, Italy, 2019, pp. 875-882.
[64] M. Krishnan, "The Noncentral Bivariate Chi Distribution", *SIAM Review*, Vol. 9, pp. 708-714, 1967.
[65] J. Lake, "Anti-Ship Missile Evolution," Asian Military Review, January 2020, Available: https://www.asianmilitaryreview.com/2020/01/anti-ship-missile-evolution.
[66] Z. Lei, J. Liu, E. Liang, and Y. Wang, "Application of the Particle Filter in IMM in Target Tracking Algorithm," *Proceedings of the International Conference on Bioinformatics and Computational Intelligence 2017*, pp. 62-65, 2017.
[67] J. M. Lemos, J. Girao, A. S. Silva, J. S. Marques, "The Fokker-Planck Equation in Estimation and Control," *IFAC*, Vol. 52, Art. No. 7, pp. 91-95, 2019.
[68] D. Lerro and Y. Bar-Shalom, "Tracking with Consistent Converted Measurements vs the EKF", *IFAC Proc.*, Vol. 26, Issue 2, Part 5, pp. 365-368, 1993.
[69] X. R. Li, "Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking," *IEEE Transactions on Control Systems Technology*, Vol. 1, Art. No. 3, 1993.

[70] T. Li, J. M. Corchado, J. Bajo, S. Sun, and J. F. De Paz, "Effectiveness of Bayesian filters: An information fusion perspective," *Information Sciences*, Vol. 329, pp. 670-689, 2016.
[71] X. R. Li, V. Jilkov, "A Survey of Maneuvering Target Tracking: Approximation Techniques for Nonlinear Filtering," *SPIE Defense + Commercial Sensing*, 2004.
[72] S. V. Lototsky and B. L. Rozovskii, "Recursive Nonlinear Filter for a Continuous Discrete-Time Model: Separation of Parameters and Observations," *IEEE Transactions on Automatic Control*, Vol. 43, Art. No. 8, August 1998.
[73] X. Luo, S.-T. Yau, "Complete Real Time Solution of the General Nonlinear Filtering Problem Without Memory," *IEEE Transactions on Automatic Control*, Vol. 58, Art. No. 10, pp. 2563-2578, 2013.
[74] X. Luo, S.-T. Yau, "A Novel Algorithm to Solve the Robust DMZ Equation in Real Time," *51st IEEE Conference on Decision and Control*, December 2012.
[75] X. Luo, "On Recent Advance of Nonlinear Filtering Theory: Emphases on Global Approaches," *Pure and Applied Mathematics Quarterly*, Vol. 10, Art. No. 4, pp. 685-721, 2014.
[76] D. Magill, "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE transactions on Automatic Control*, Vol. 10, Art. No. 4, pp. 434-439, 1965.
[77] M. Mallick, "A Note on Bearing Measurement Model," *ResearchGate*, 2018, DOI: 10.13140/RG.2.2.13441.35681.
[78] K. V. Mardia and P. E. Jupp. *Directional Statistics*. Chichester New York: J. Wiley, 2000.
[79] G. Marsaglia, "Ratios of Normal Variables and Ratios of Sums of Uniform Variables." *J. Amer. Stat. Assoc.*, Vol. 60, Art. No. 309 (1965): pp. 193–204.
[80] MathWorks™, (2021), *GPU Computing Toolbox™: User's Guide (R2021b)*, Retrieved 15 Dec. 2021 from www.mathworks.com/help/matlab/ref/gpuArray.
[81] MathWorks™, (2021), *Linear Algebra Toolbox™: User's Guide (R2021b)*, Retrieved 12 Dec. 2021 from www.mathworks.com/help/matlab/ref/mldivide.
[82] MathWorks™, (2021), *Linear Algebra Toolbox™: User's Guide (R2021b)*, Retrieved 12 Dec. 2021 from www.mathworks.com/help/matlab/ref/sqrtm.
[83] MathWorks™, (2021), *Sparse Matrices Toolbox™: User's Guide (R2021b)*, Retrieved 15 Dec. 2021 from www.mathworks.com/help/matlab/ref/sparse.
[84] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting Multiple Model Methods in Target Tracking: A Survey," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, Art. No. 1, 1998, pp. 103-123.
[85] M. Mider, M. Schauer, and F. van der Meulen, "Continuous-discrete smoothing of diffusions," *Electronic Journal of Statistics*, Vol. 15, pp. 4295-4352, 2021.
[86] M. Mir, "Cruise Missile Target Trajectory Movement Prediction Based on Optimal 3D Kalman Filter with Firefly Algorithm," *ArXiv*, abs/1807.07006, 2018.
[87] R. E. Mortensen, "Mathematical Problems of Modeling Stochastic Nonlinear Dynamic Systems," *Journal of Statistical Physics*, Vol 1, pp. 271-296, 1969. DOI: 10.1007/BF01007481.
[88] N. F. Palumbo, R. A Blauwkamp, and J. M. Lloyd, "Basic principles of homing guidance", *Johns Hopkins APL Technical Digest*, Vol. 29, Iss. 1, pp. 25-41 2010.
[89] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1965.

[90] S. Pasandideh, “Building a Navy Fit for War: A Survey of Anti-Ship Cruise Missiles,” *NATO Association of Canada*, March 2014.
[91] R. R. Pitre, V. P. Jilkov, X. R. Li, “A Comparative Study of Multiple-Model Algorithms for Maneuvering Target Tracking,” *Proceedings of the SPIE, Signal Processing, Sensor Fusion, and Target Recognition XIV*, Vol. 5809, pp. 549-560, 2005.
[92] A. Pollastri and F. Tornaghi, “Some Properties of the Arctangent Distribution.” *Statistica & Applicazioni*, Vol. 2, Art. No. 1, pp. 3-18, 2004.
[93] S. O. Rice, “Mathematical Analysis of Random Noise.” *The Bell System Technical Journal*, Vol. 24 pp. 46–156, 1945.
[94] H. Risken, *The Fokker-Planck Equation: Methods of Solution and Applications*, 2nd ed., H. Haken, Ed. Springer-Verlag, 1999.
[95] S. Sarkka, “Recursive Bayesian Inference on Stochastic Differential Equations,” PhD dissertation, Laboratory of Computational Engineering, Helsinki University of Technology, Espoo, Finland, 2006.
[96] M. S. Schlosser and K. Kroschel, "Limits in tracking with extended Kalman filters," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, Art. No. 4, pp. 1351-1359, 2004.
[97] J. C. Schulte, “An analysis of the historical effectiveness of anti-ship cruise missiles in littoral warfare,” M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA, September 1994.
[98] D. W. Scott, “Multivariate density estimation and visualization.”, *Handbook of Statistics*, Vol. 23: Data Mining and Computational Statistics, Springer; 2004
[99] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall, 1986.
[100] D. J. Simon and T. L. Chia, "Kalman filtering with state equality constraints", *IEEE Transactions on Aerospace and Electronic Systems*, Vol 38, Art. No. 1, pp. 128-136, 2002, DOI: 10.1109/7.993234
[101] D. Simon, "Kalman filtering with state constraints: A survey of linear and nonlinear algorithms", *IET Control Theory Appl.*, Vol. 4, Iss. 8, pp. 1303–1318, 2010, DOI: 10.1049/ietcta. 2009.0032.
[102] N. Shimkin, “Estimation and Identification in Dynamical Systems,” Lecture Notes, Department of Electrical Engineering, Israel Institute of Technology, Israel, 2009.
[103] S. Song and J. Wu, "Motion State Estimation of Target Vehicle under Unknown Time-Varying Noises Based on Improved Square-Root Cubature Kalman Filter" *Sensors*, Vol. 20, Iss. 9, Art. No. 2620, 2020, DOI: 10.3390/s20092620.
[104] D. Svensson, "Derivation of the Discrete-Time Constant Turn Rate and Acceleration Motion Model," *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1-5, 2019, DOI: 10.1109/SDF.2019.8916654.
[105] K. R. Symon, *Mechanics, Second Edition*, Addison-Wesley, 1960
[106] M. Tahk and J. L. Speyer, "Target tracking problems subject to kinematic constraints", *IEEE Transactions on Automatic Control*, Vol. 35, Iss. 3, pp. 324-326, 1990, DOI:10.1109/9.50348
[107] J. W. Thomas, R. J. Bailey, W. D. Stuckey, et al., “SSDS Mk 2 Combat System Integration,” *Johns Hopkins APL Technical Digest*, Vol. 22, Art No. 4, pp. 547-563, 2001.

[108] S. Ungarala, E. Dolence, and K. Li, "Constrained extended Kalman filter for nonlinear state estimation", *8th IFAC Symposium on Dynamics and Control Process Systems*, Vol. 2, pp. 63-68, 2007.
[109] E. A. Wan and R. V. Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153-158, 2000.
[110] A. Wawrzynczak, R. Modzelewska, and G. Agnieszka, "Algorithms for Forward and Backward Solution of the Fokker-Planck Equation in the Heliospheric Transport of Cosmic Rays," *PPAM*, pp. 14-23, 2017.
[111] A. S. Willsky, "Estimation for Rotational Processes with One Degree of Freedom – Part I: Analysis," *IEEE Transactions on Information Theory*, Vol. 20, pp. 577-583, 1974.
[112] A. S. Willsky and J. T. Lo, "Estimation for Rotational Processes with One Degree of Freedom – Part II: Discrete-Time Processes," *IEEE Transactions on Information Theory*, Vol. 20, pp. 584-590, 1974.
[113] C. Wolf, "AN/SPS-48E," *RadarTutorial.eu*, Accessed on: 17 Dec. 2017.
[114] E. Wong, M. Zakai, "On the Convergence of Ordinary Integrals to Stochastic Integrals," *The Annals of Mathematical Statistics*, Vol. 36, Art. No. 5, pp.1560-1564 1965.
[115] C. Yang and E. Blasch, "Kalman Filtering with Nonlinear Constraints", *International Conference on Information Fusion*, 2006.
[116] C. D. Yang and C. C. Yang, "Analytical Solution of Generalized 3D Proportional Navigation", *Proc. 34th Conference on Decision & Control*, pp. 3974-3979, 1991.
[117] S. S.-T. Yau, "Complete Classification of Finite-Dimensional Estimation Algebras of Maximal Rank," *International Journal of Control*, Vol. 76, Art. No. 7, pp. 657-677, 2003.
[118] S.-T. Yau and S. Yau, "Existence and Uniqueness of Solutions for Duncan-Mortensen-Zakai Equations," *44th IEEE Conference on Decision and Control*, 2005.
[119] S.-T. Yau and Y.-T. Lai, "Explicit Solution of DMZ Equation in Nonlinear Filtering Via Solution of ODEs," *IEEE Transactions on Automatic Control*, Vol. 48, pp. 505-508, 2003.
[120] S.-T. Yau and S. S.-T. Yau, "Nonlinear Filtering and Time Varying Schrodinger Equation I," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, Art. No. 1, pp. 284-292, 2004
[121] S. S.-T. Yau and S.-T. Yau, "Solution of filtering problem with nonlinear observations," *SIAM Journal of Control and Optimization*, Vol. 44, Art. No. 3, pp. 1019-1039, 2005.
[122] J. Yepes, I. Hwang, and M. Rotea, "An Intent-Based Trajectory Prediction Algorithm for Air Traffic Control." *AIAA Guidance, Navigation, and Control Conf.*, San Francisco, USA, 2005.
[123] M. Yueh, W. Lin, and S.-T. Yau, "An Efficient Algorithm of Yau-Yau Method for Solving Nonlinear Filtering Problems," *Communications in Information and Systems*, Vol. 14, Art. No. 2, pp. 111-134, 2014.
[124] M. Zakai, "On the Optimal Filtering of Diffusion Processes," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, Vol. 11, pp. 230-243, 1969.

[125] G. Zhou, K. Li, X. Chen, L. Wu, and T. Kirubarajan, “State Estimation with Destination Constraint Using Pseudo-measurements.” *Signal Processing*, Vol. 145, pp. 155-166, 2018.