ECE 767

Multitarget Tracking and Multisensor Information Fusion

Report assignment 1
Instructor: Dr. R. Tharmarasa

Tara Rajabi Nezhad Siahpoosh
Student number: 400548784

Winter 2024

Contents

Problem description

The task entails implementing a nearest-neighbor Constant-Memory Kalman Filter (CMKF) tracker within a specified scenario involving a moving target and sensor-generated data. The report aims to address the following components:

Key Components:

- Target Generation: Understanding the motion model and noise characteristics of the target.
- Sensor Simulation: Generating false alarms with a specified distribution.
- Tracker Implementation: Implementing nearest-neighbor data association and CMKF filtering.
- Performance Evaluation: Analyzing RMSE metrics and studying performance variations with different scenarios.

Objectives:

- Develop a functional nearest-neighbor CMKF tracker.
- Evaluate its performance using RMSE metrics.
- Investigate how different target trajectories and sensor parameters impact tracking accuracy.

Implementation details

The tracking system is implemented using a Kalman filter-based approach, comprising several key functions and parameter definitions. The main script orchestrates the simulation, while functions such as moveTarget.m, generateMeasurements.m, dataAssociation.m, and the KalmanFilter.m handle specific aspects of the tracking process. Also, in the parameter.m file all the parameters have been defined.

Main Script

The main script initializes variables for tracking performance evaluation and orchestrates the simulation across multiple Monte Carlo runs. Within each run, the script iterates over time steps, updating the target's true state, generating measurements, performing data association, and applying the Kalman filter for state estimation. It calculates RMSE values for position and speed, plots trajectories, and assesses tracking performance.

move Target

This function updates the target's state based on its previous state and motion parameters. It utilizes the motion model equation, incorporating the state transition matrix and process noise to accurately model the target's motion over time. The moveTarget.m computes the updated state of the target at time step k using the motion model equation:

$$xk = F * xk_1 + G * V$$ *Equation 1*

This equation represents the state transition process, where the new state x_k is obtained by applying the state transition matrix F to the previous state x_{k-1}, and adding the effect of process noise $G * V$.

By incorporating motion parameters and process noise characteristics, it facilitates the accurate modeling of the target's motion over time. Understanding the implementation details of this function is essential for ensuring the robustness and effectiveness of the tracking system.

Generate Measurements

The generateMeasurements.m function simulates sensor measurements within the tracking system. It considers sensor parameters and target state information to generate realistic measurements, including both target measurements and false alarms. The function performs the following key steps:

1. Extract essential sensor parameters from the parameter structure.
2. Calculate coverage volume based on specified range and azimuth.
3. Generate measurements from the target with a probability determined by the detection probability parameter.
4. Add measurement noise and check if the measurements fall within the coverage volume.
5. Generate false alarms based on specified parameters and add them to the measurement array.

This function's implementation ensures realistic sensor behavior simulation, essential for evaluating tracking system performance.

Data association

The dataAssociation.m function is pivotal in tracking systems, ensuring accurate association between measurements and predicted target states. By implementing the Mahalanobis distance metric within a specified gating region, this function enhances the reliability and robustness of the tracking process. A thorough understanding of its implementation details is essential for optimizing tracking performance.

The dataAssociation function begins by initializing variables to store the associated measurement ID, associated measurement, minimum distance, and corresponding measurement covariance matrix. It then iterates through each measurement obtained from the sensors. For each measurement, it converts the measurement covariance matrix from polar to Cartesian coordinates and predicts the measurement using the measurement matrix H and the predicted target state estimate $\hat{x}_k$. The function computes the innovation or measurement residual and calculates the Mahalanobis distance using the innovation, measurement covariance matrix S, and its inverse. It checks if the measurement lies within the specified gating region and if its distance is smaller than the previous minimum distance. If the conditions are met, it updates the associated measurement ID, associated measurement, minimum Mahalanobis distance, and corresponding measurement covariance matrix. Finally, the function returns the associated measurement ID, associated measurement, minimum Mahalanobis distance, and corresponding measurement covariance matrix.

Kalman Filter

The KalmanFilter.m implements the Kalman filter algorithm for estimating the target state based on noisy measurements. It consists of prediction and update steps. In the prediction step, it predicts the state estimate and covariance matrix at the current time step using the state transition matrix, previous state estimate, and process noise covariance matrix. In the update step, if measurements are available, it calculates the predicted measurement, innovation, innovation covariance, Kalman gain, and updates the state estimate and covariance matrix using these values. This function provides an optimal estimate of the target state by considering system dynamics and measurement noise.

Plots of the truth and estimated trajectories

Note that all the following plots are implemented with 100 monte runs and 50-time steps. the fixed parameter values are in Table 1.

Parameters	Value
Monte Carlo Runs	100
Number of Time Steps	50
Target Start Time	0.80
Target Start State	[100, 30, 3000, 20]
Target Process Noise	0.5
Target Velocity	[10, 10]
Sensor Sampling Time	2
Sensor Position	[1000, 500]
Sensor Velocity	[0, 0]
Sensor Measurement Error Range	10 meters
Sensor Measurement Error Azimuth	0.01 radians
Sensor Detection Probability (Pd)	0.9
Sensor False Alarm Density	10^{-4}
Sensor Coverage Range Bound	0 to 10000
Sensor Coverage Azimuth Bound	-π to π
Tracker Gate Size	chi2inv (0.99, 2)
Tracker Initial Covariance Matrix	Diagonal $[100^2, 10^2 2, 100^2, 10^2]$
Performance Evaluation Gate Size	150

Table 1

Considering the Table 1 condition the truth and estimated trajectories and the RMSE of this scenario is plotted in Figure 1 and Figure 2. Note that in all the following results the **red lines are truth, and the blue lines are the estimated trajectories**.

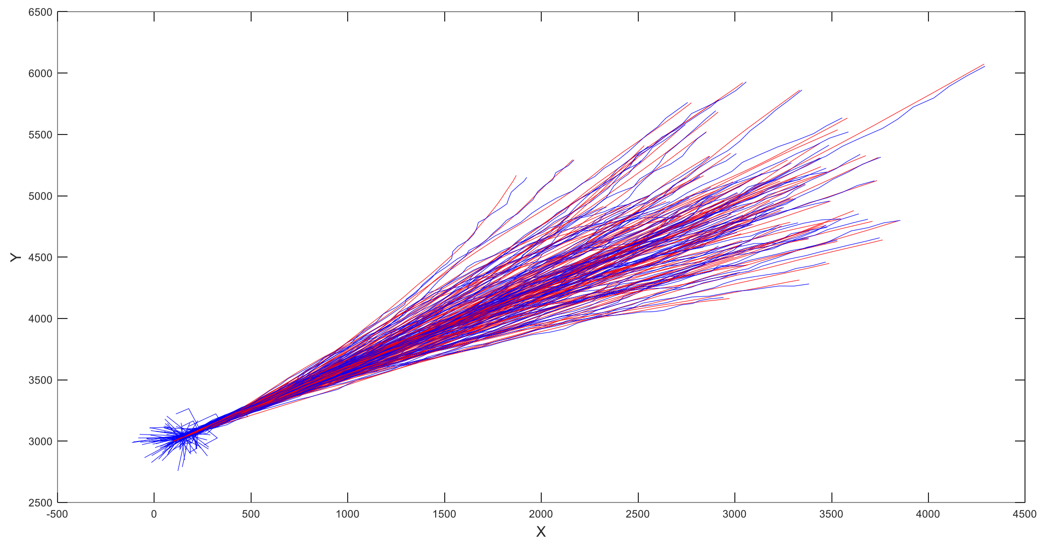

Figure 1: main scenario trajectory

In Figure 1 we can see that most of trajectories are true.

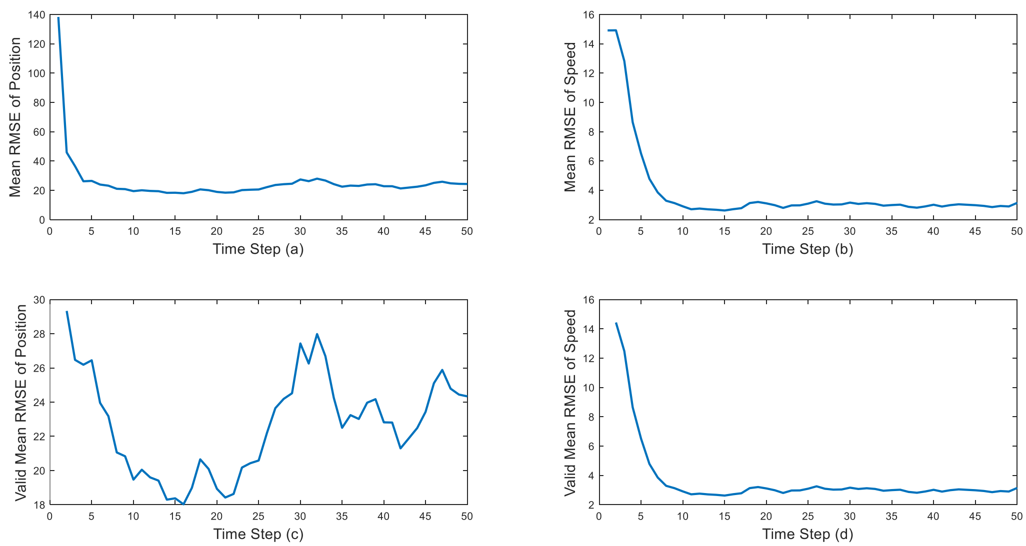

Figure 2: main scenario RMSE

In Figure 2, we present an analysis of the root mean square error (RMSE) metrics for both position and speed tracking over multiple time steps and Monte Carlo runs.

(a) The plot represents the mean position RMSE for each time step averaged across all Monte Carlo runs. In this specific case, the overall mean position RMSE is calculated to be 25.4449.

(b) Moving to the speed tracking, the plot illustrates the RMSE of speed over time steps.

(c) This diagram displays the exact same RMSE values as in (a), but with invalid detections removed. In this context, the term "invalid" refers to detections that are deemed unreliable or inaccurate which is our case are those which are above 150 meters. In our scenario, only the first

detection is considered invalid, which is expected since it's the initial estimate the system makes. Subsequent detections are considered valid as the system refines its estimate based on additional information.
(d) Similar to (c), this plot focuses on the RMSE of valid speed detections, i.e., speed detections corresponding to valid position detections. This filtering process ensures that only reliable speed measurements are considered for analysis, enhancing the accuracy of our assessment.
By presenting these plots, we gain insights into the performance of our tracking system, understanding both the overall tracking errors and the impact of filtering out unreliable detections on our speed estimation.

The fluctuation in the RMSE values observed in Figure 2(c) can indeed be attributed to the changing distance between the target and the sensor over time. Initially, as the target moves closer to the sensor, the measurements tend to be more accurate, resulting in lower RMSE values. This is because the sensor has a better ability to detect and track the target when it is in close proximity.
However, as the target moves away from the sensor, the measurements become less accurate due to factors such as reduced signal strength and increased susceptibility to noise. This leads to an increase in RMSE values as the tracking accuracy decreases.
This pattern of decreasing and then increasing RMSE values reflects the dynamic nature of the tracking process, where the reliability of measurements is influenced by the distance between the target and the sensor. Understanding these fluctuations helps in interpreting the tracking performance and optimizing the tracking system for varying target-sensor distances.

The effect of Probability of detection

In an ideal scenario where there are no faults and the probability of detection is 100%, we expect perfect tracking performance where all trajectories follow the true states precisely. Figure 3 and **Error! Reference source not found.** provide visual and quantitative assessments of the tracking performance under these ideal conditions.

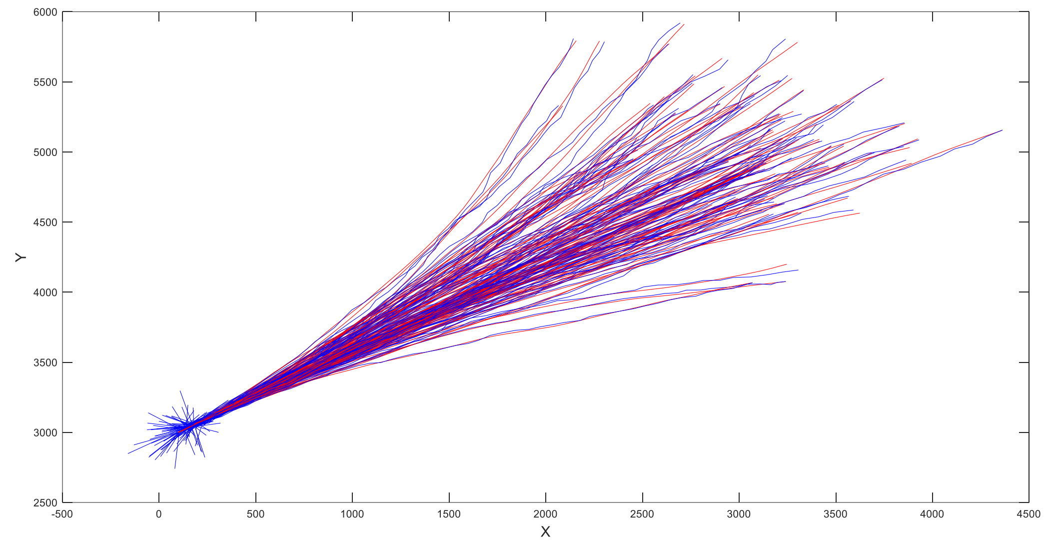

Figure 3: ideal scenario trajectories.

Figure 3 illustrates the trajectories of the tracked objects overlaid on the true trajectories. In this ideal scenario, we expect the tracked trajectories to perfectly align with the true trajectories, demonstrating accurate tracking without any deviations or errors. Any deviation observed in the tracked trajectories from the true trajectories would indicate potential inaccuracies or limitations in the tracking system, even under ideal conditions.

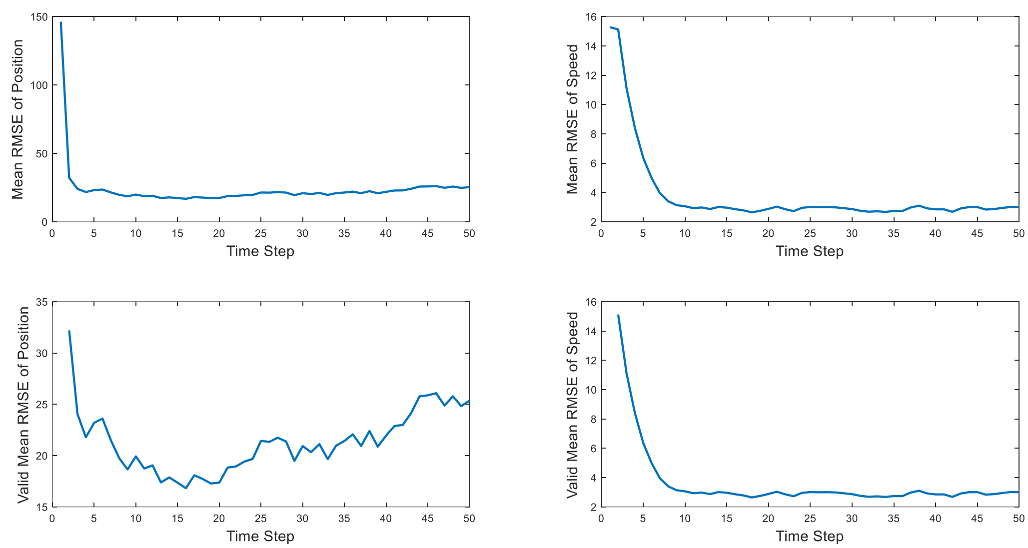

Figure 4: ideal scenario RMSE

The mean RMSE of position in this case is slightly lower at 23.8448 compared to the previous scenario. **Error! Reference source not found.** illustrates the RMSE values between tracked and true trajectories over time. Low and consistent RMSE values indicate minimal deviation between tracked and true trajectories, reflecting accurate tracking. However, any significant fluctuations

or spikes in RMSE may suggest moments of decreased tracking accuracy, possibly due to environmental factors, sensor limitations, or algorithmic issues.
Analyzing both the trajectory plots and RMSE values allows us to evaluate the effectiveness and reliability of the tracking system under ideal conditions. Any observed discrepancies should be investigated to identify areas for potential improvement in the tracking algorithm or system configuration.

Now let’s consider a situation where the probability of detection is zero.

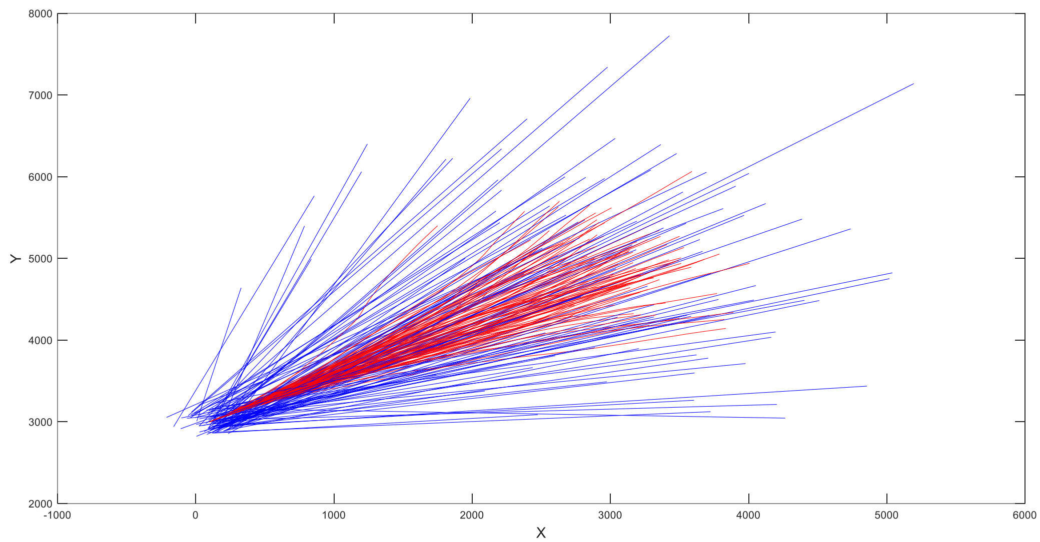

Figure 5: zero probability of detection trajectories.

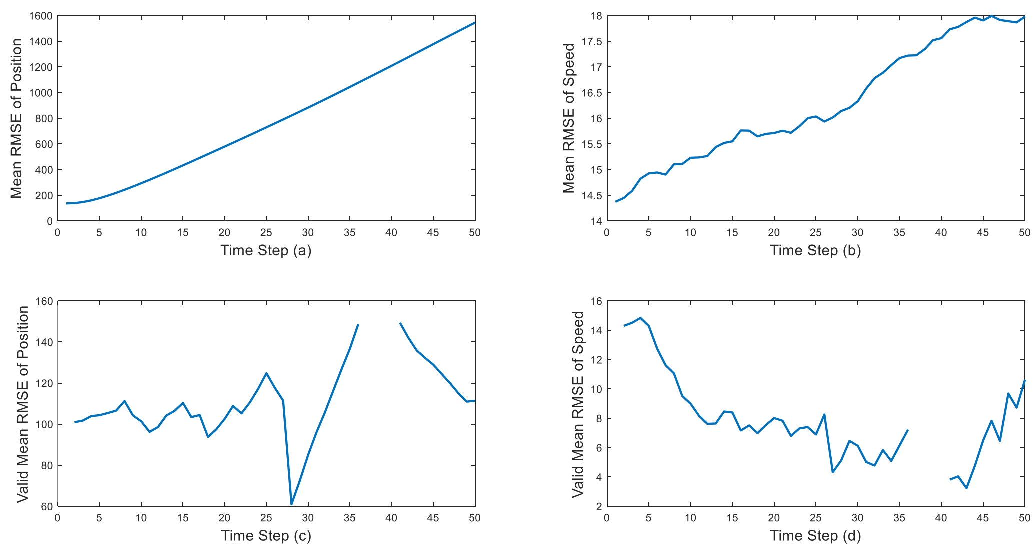

Figure 6: : zero probability of detection RMSE

This situation leads to a breakdown in tracking performance, as the system relies entirely on predicted estimates without any real-time feedback from the environment. Consequently, the system loses track of the target, resulting in unreliable state estimates and increased uncertainty in the target's trajectory. Without the ability to detect the target, the tracking algorithm struggles to maintain situational awareness and respond effectively to changing circumstances. Critical events may go unnoticed, posing risks to operational capabilities, especially in surveillance or defense applications where reliable target tracking is paramount. Thus, a zero probability of detection underscores the importance of robust sensor systems and tracking algorithms capable of operating effectively in challenging environments and maintaining tracking performance under adverse conditions.

To conclude these, as the probability of detection (PD) decreases from 1 to 0, the number of trajectories effectively diminishes due to the tracking system's inability to detect and follow targets reliably. At $PD = 1$, indicating perfect detection, the system can track all targets present within its sensing range, resulting in a one-to-one correspondence between detected targets and tracked trajectories. However, as PD decreases, the system misses more targets, leading to fewer trajectories being generated. Consequently, the tracking system's effectiveness and coverage decline as the PD decreases, impacting its ability to provide accurate situational awareness and make informed decisions. This underscores the critical role of PD in determining the tracking system's performance and highlights the need for robust detection capabilities to maintain comprehensive tracking coverage in dynamic environments.

The effect of sensor false alarm density

In the Figure 1 and Figure 2 the false alarm density was 10^{-4}. Increasing the false alarm density from 10^{-4} to 10^{-3} would significantly affect the tracking system's performance. The higher false alarm rate would result in an increased number of spurious measurements that are not associated with actual targets, leading to a cluttered sensor output and decreased detection reliability. This increase in false alarms complicates the data association process, as the system must differentiate between valid target measurements and false alarms, potentially resulting in more frequent association errors and reduced tracking accuracy. Additionally, handling a higher false alarm density requires additional computational resources, potentially impacting real-time performance. Overall, the higher false alarm density would likely lead to decreased tracking accuracy, reduced situational awareness, and challenges in decision-making, emphasizing the importance of robust data association algorithms and filtering techniques to mitigate these effects and maintain tracking performance. This effect has been shown is Figure 7 and Figure 8.

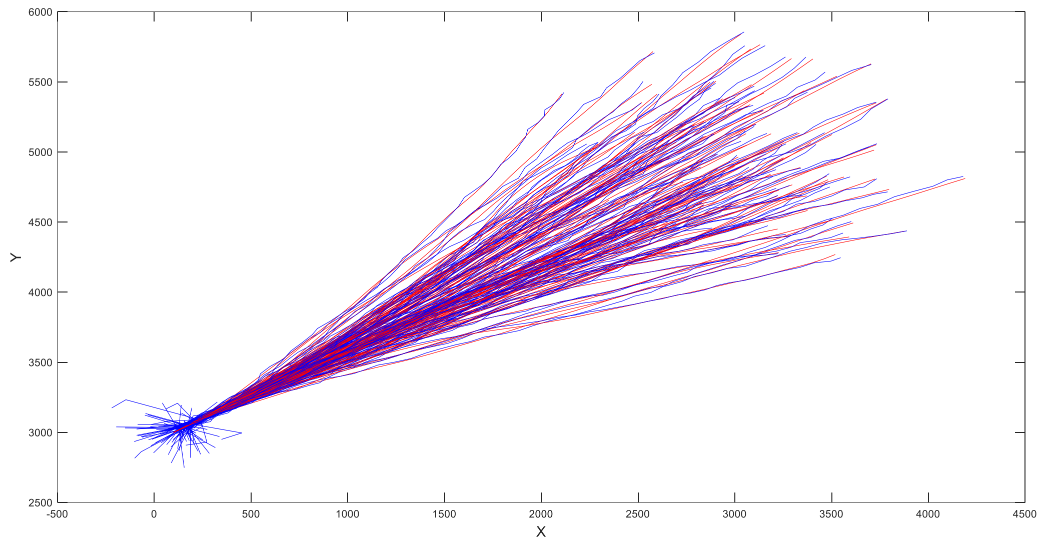

Figure 7: higher sensor false alarm density trajectories.

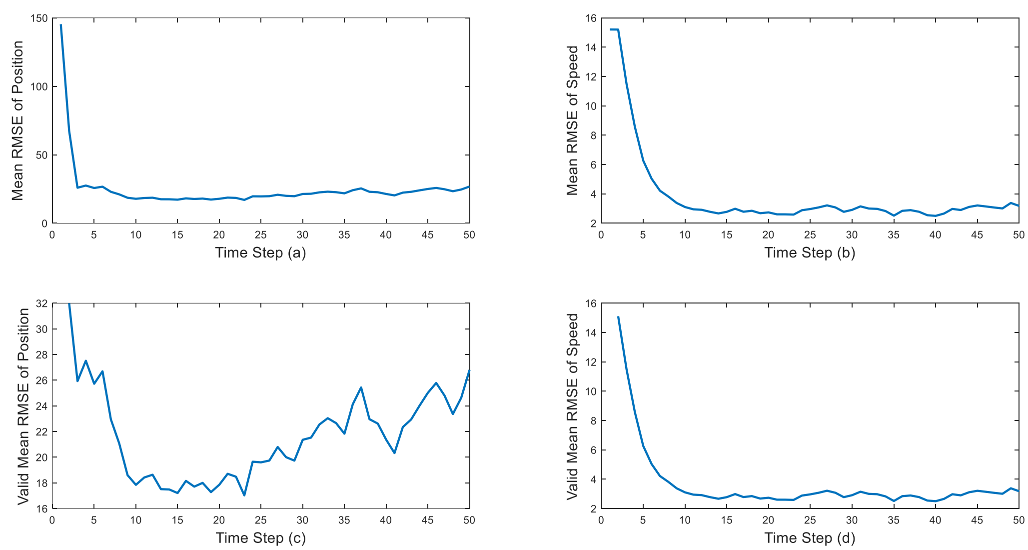

Figure 8: higher sensor false alarm density RMSE.

The effect of Target Start State

As mentioned before in Figure 1 and Figure 2 the target initial state was $[100, 30, 3000, 20]$ representing the initial positions and velocities in both the x and y dimensions. However, for this scenario, we have modified the initial state to $[700, 30, 7007, 20]$. This adjustment brings the target's initial position and velocity closer to the sensor's location at $[100, 500]$. The Root Mean Square Error (RMSE) initially remains low but eventually increases over time. This behavior can be attributed to the initial proximity of the target to the sensor, resulting in more accurate measurements and lower tracking errors in the early stages. However, as time progresses, the target moves away from the sensor, leading to a decrease in measurement accuracy and an increase in tracking errors. Consequently, the RMSE values gradually rise as the distance between the target and the sensor increases, highlighting the impact of target-sensor distance on tracking performance. This observation underscores the importance of considering the initial target state and its relation to the sensor location when evaluating tracking performance, as well as the dynamic nature of tracking errors over time.

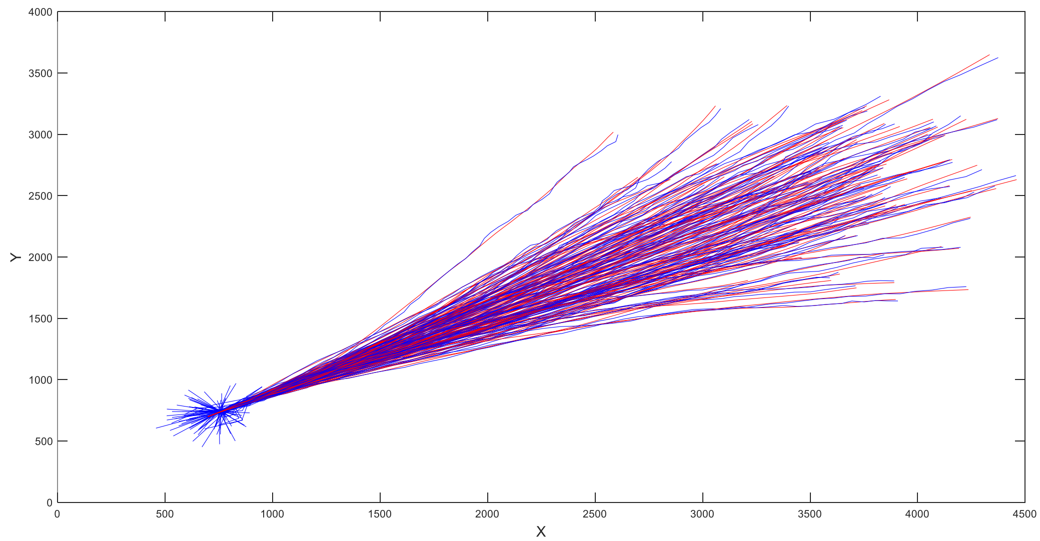

Figure 9: Target Start State (position change) trajectories.

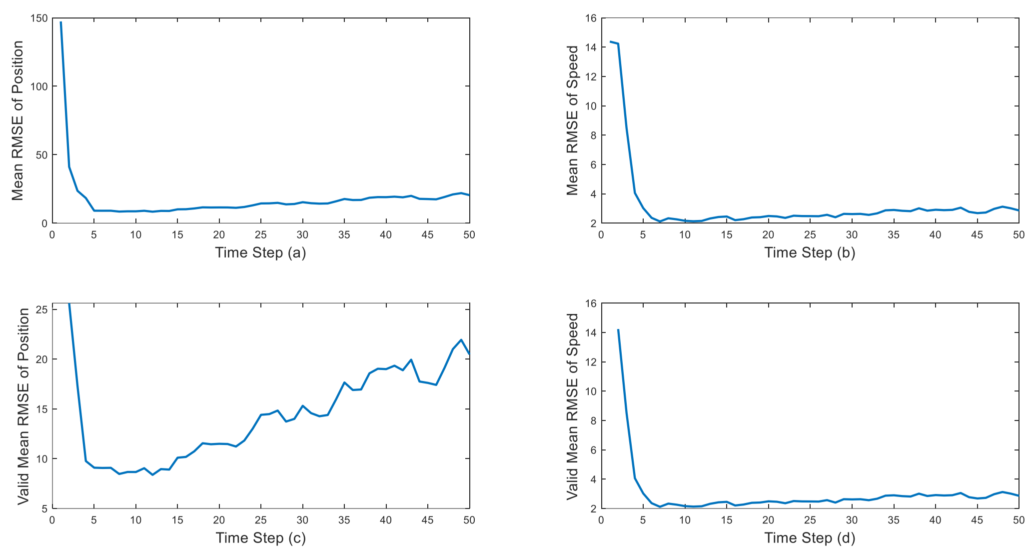

Figure 10: Target Start State (position change) RMSE

In this scenario, the initial target states have been adjusted to $[100, 70, 3000, 70]$, specifically increasing the target's velocity while maintaining its initial positions. This modification introduces a faster-moving target into the tracking scenario, which can have significant implications for the tracking system's performance. The higher target velocity of 70 in both the x and y dimensions accelerate the target's motion, leading to faster traversal of the tracking area and more rapid changes in position between consecutive time steps. As a result, the tracking algorithm faces challenges in accurately predicting the target's future positions based on its current state and motion model. This increased velocity also complicates the data association process, as larger displacements between consecutive measurements require the system to effectively handle these changes and associate measurements with the correct target states. Consequently, tracking accuracy may be impacted, with potential inaccuracies arising if the system fails to update the target's state quickly enough to reflect its actual position and velocity. Evaluating the RMSE values in this scenario would likely reveal changes reflecting the increased target velocity, with potentially higher RMSE values indicating greater tracking errors resulting from the challenges associated with tracking faster-moving targets.

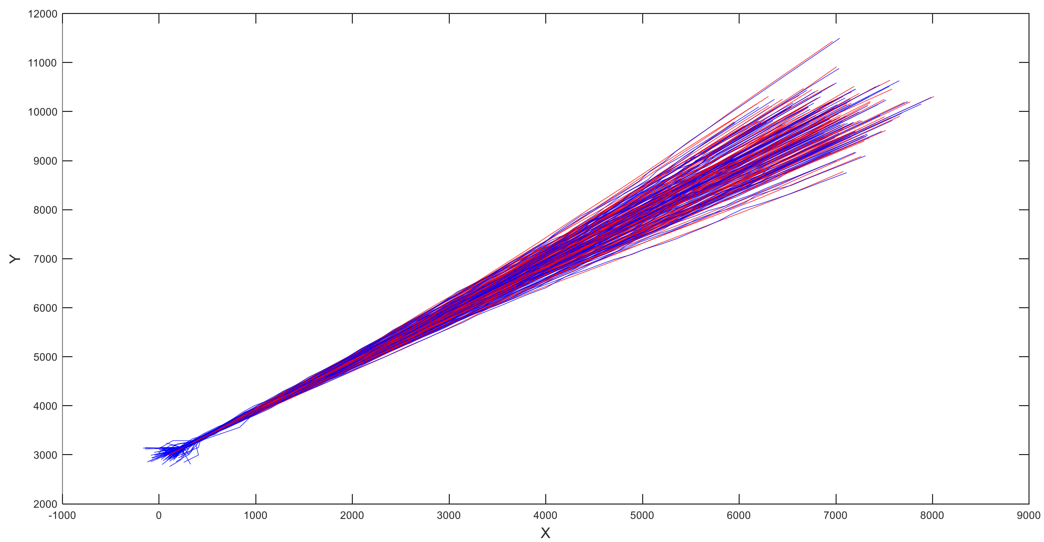

Figure 11: Target Start State (speed change) trajectories.

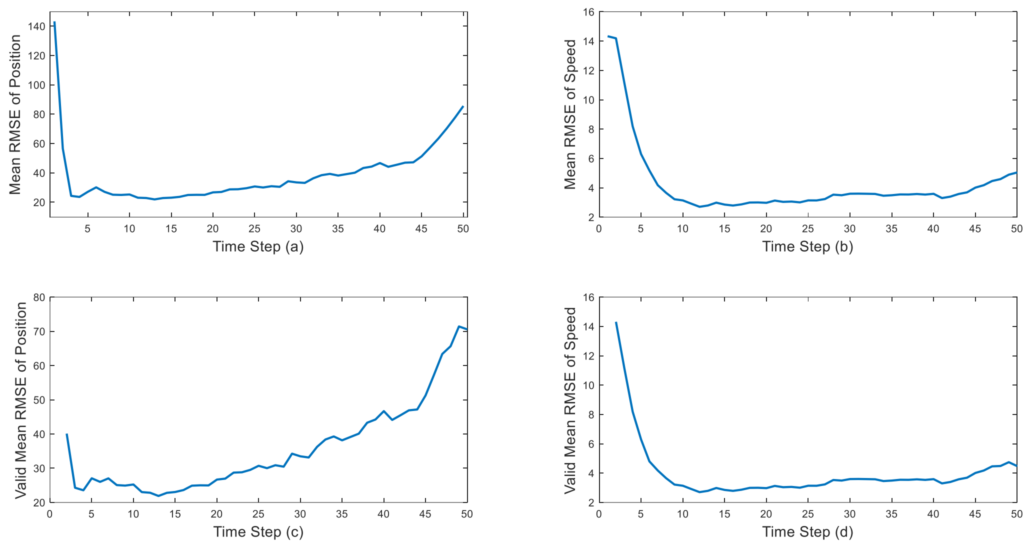

Figure 12: Target Start State (speed change) RMSE

The effect of process noise

In the main scenario the process noise is 0.5 in the next scenario it is increased to be one. Increasing the process noise introduces greater uncertainty into the tracking system, leading to several challenges. Firstly, the system's predictions may deviate more from the actual target trajectory due to this increased uncertainty. Consequently, tracking errors tend to be larger as the system struggles to compensate for the unpredictability in the target's motion. Moreover, the performance of the Kalman filter, a fundamental component in tracking systems for estimating target states, is adversely affected by higher process noise levels. The filter relies on accurate process noise modeling to effectively filter out noise and generate optimal state estimates. However, with increased process noise, the filter's performance is compromised, resulting in less accurate state estimates and potentially poorer overall tracking performance. Additionally, higher process noise levels pose challenges for data association, making it more difficult to correctly associate measurements with predicted target states. The increased uncertainty in target predictions can lead to higher rates of association errors, impacting the overall tracking accuracy. Therefore, while adjusting process noise parameters is essential for adapting to varying tracking conditions, careful consideration is necessary to maintain a balance between accuracy and robustness in the tracking system.

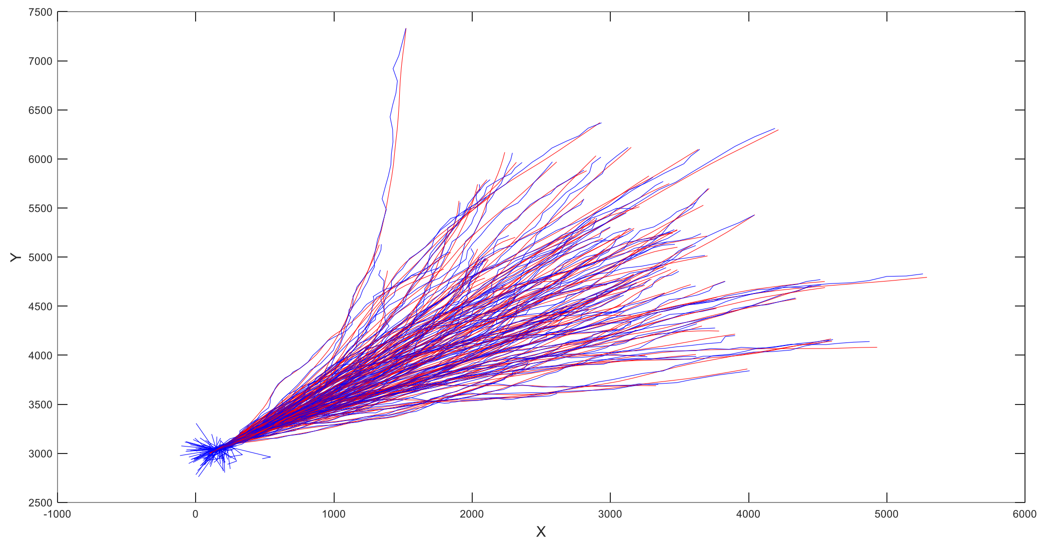

Figure 13: increased process noise trajectories.

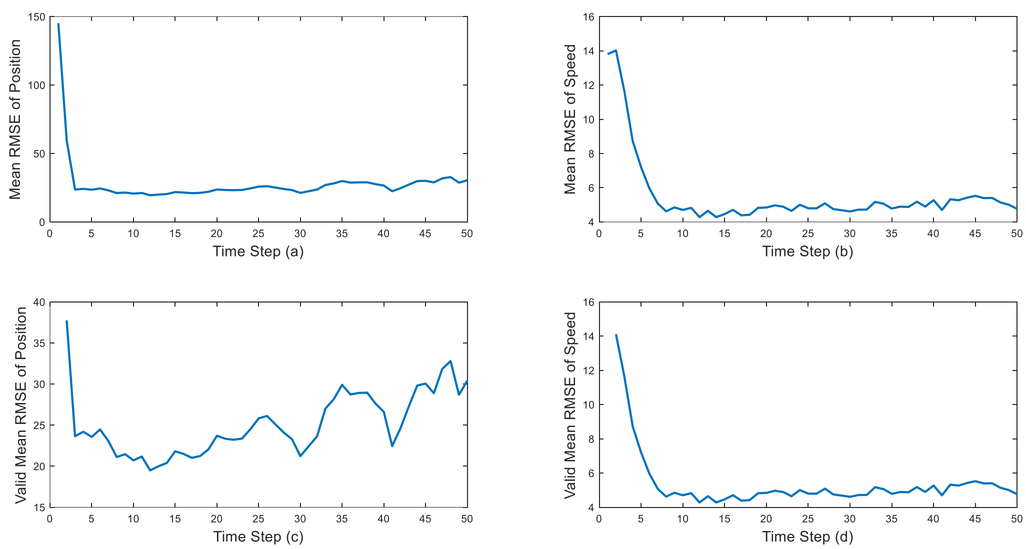

Figure 14: increased process noise trajectories.

The effect of sensor measurement error

Effect of Range Error (Error Range = 50 meter to Error Azimuth = 0.01 radians): Increasing the error range introduces greater uncertainty in distance measurements obtained from the sensor. With less precise distance information, the tracking system may struggle to accurately estimate the target's position, leading to larger deviations between estimated and actual target positions. This decrease in tracking accuracy can result in higher tracking errors and difficulties in data association, as the system must contend with increased variability in distance measurements. Consequently, maintaining accurate tracking becomes challenging, highlighting the importance of precise range measurements for optimal tracking performance.

Effect of Azimuth Error (Error Range = 10 meters to Error Azimuth = 0.05 radians): A larger azimuth error contributes to increased uncertainty in directional measurements obtained from the sensor. This heightened uncertainty in target direction or bearing can lead to decreased tracking accuracy, particularly in scenarios where precise directional information is critical for accurate tracking. With larger deviations between estimated and actual target directions, the tracking system may experience higher tracking errors and encounter difficulties in data association. Managing these challenges becomes essential for maintaining accurate tracking, underscoring the significance of precise azimuth measurements in tracking system design and operation.

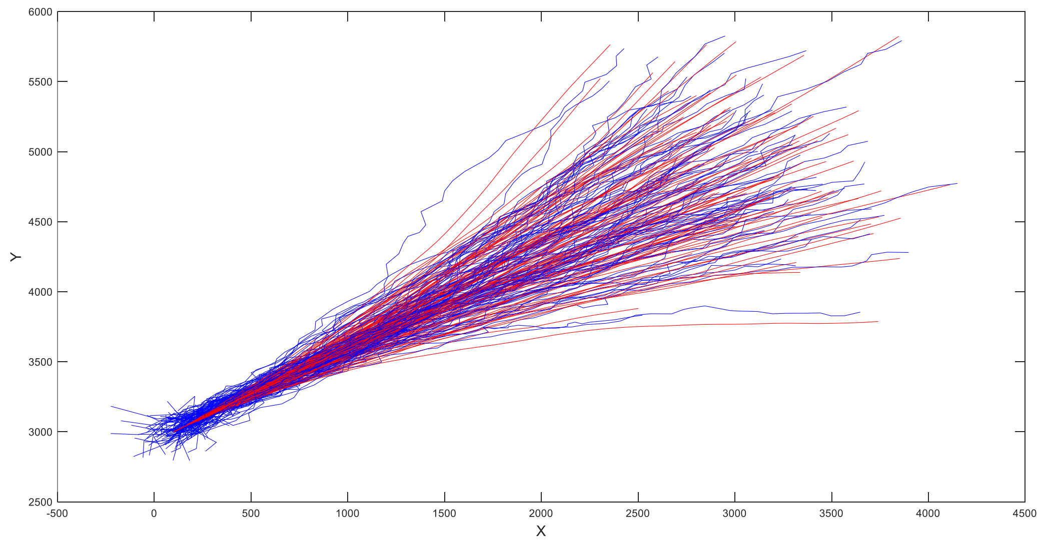

Figure 15: The effect of sensor measurement error trajectories.

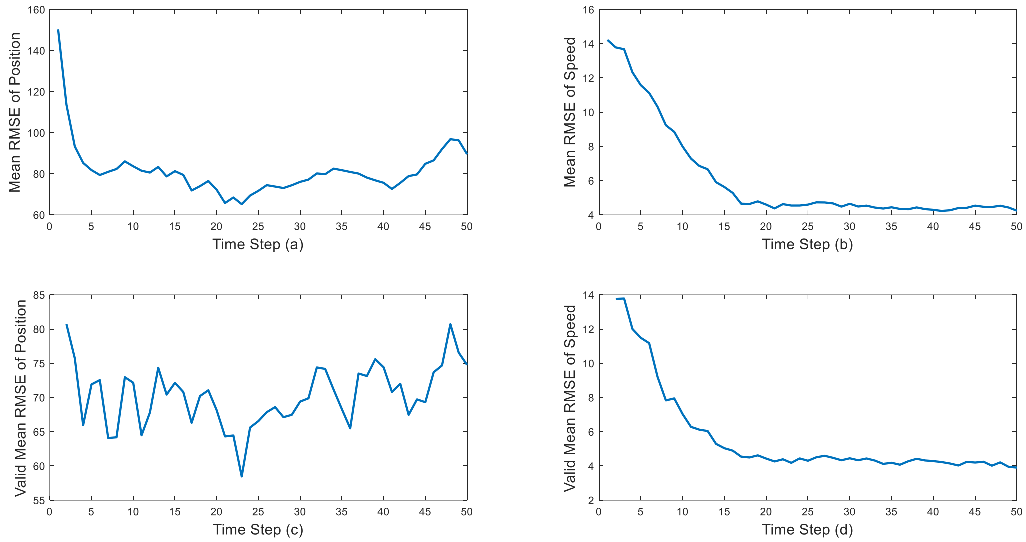

Figure 16: The effect of sensor measurement error RMSE.

In Figure 16, the increased RMSE values vividly illustrate the impact of larger range and azimuth errors on tracking accuracy. The RMSE values are noticeably higher compared to scenarios with lower sensor errors. This increase in RMSE values reflects the degradation in tracking accuracy caused by the larger errors in range and azimuth measurements. Therefore, Figure 16 serves as a visual confirmation of the challenges posed by increased sensor errors and highlights the importance of addressing these issues through proper calibration and adjustment of sensor parameters to maintain effective tracking performance.

The effect of sensor coverage range

Reducing the coverage range from [0, 10000] to [0, 1000] in the tracking scenario would significantly alter the system's performance and coverage capabilities. With a smaller coverage area, the sensor's detection capabilities become constrained, limiting the region where targets can be reliably detected and tracked. This reduction in coverage range would lead to a decreased detection range, meaning that targets located beyond the new coverage boundaries would no longer be within the sensor's reach. Consequently, the tracking system may encounter challenges in accurately tracking targets, particularly those moving outside the sensor's coverage area. This could result in increased tracking errors and difficulties in maintaining continuous tracking of targets as they move within and outside the sensor's limited coverage range. Furthermore, the reduction in coverage range increases the likelihood of missed targets, as objects entering the tracking environment beyond the new coverage boundaries may go undetected by the sensor. As a result of these changes, the RMSE values are expected to increase, reflecting the degraded tracking accuracy and the system's reduced ability to effectively track targets within the constrained coverage area. Therefore, careful consideration and calibration of coverage parameters are crucial to mitigate the impact of reduced coverage range on tracking performance and ensure optimal tracking accuracy in the given environment.

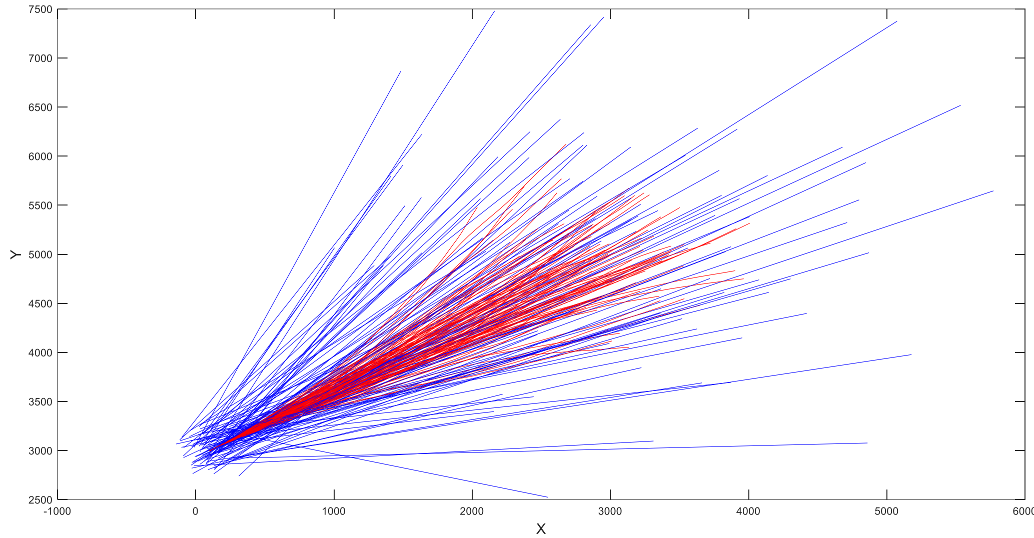

Figure 17: decreasing sensor coverage range trajectories.

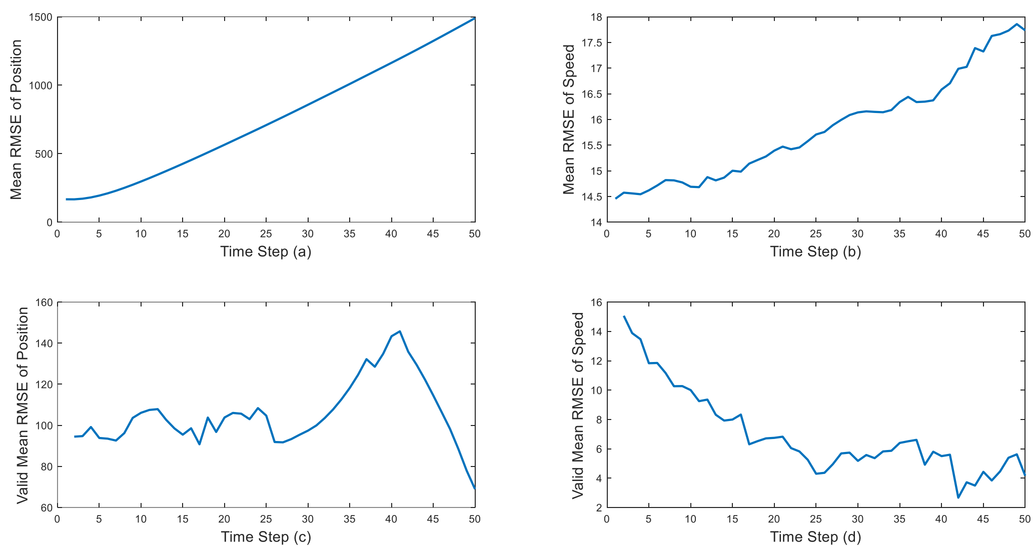

Figure 18: decreasing sensor coverage range RMSE.

In this scenario, where the sensor's coverage range has been reduced to [0, 1000], the sensor indeed struggle to track targets that fall outside its coverage area. With the target positioned beyond the sensor's limited coverage range, the sensor lacks the capability to detect or track the target effectively. As a result, the tracking system may experience difficulties in maintaining continuous tracking of the target's motion and providing accurate state estimates. In this scenario sensor can’t track target since the target is not in the coverage sensor area.

Decreasing the sensor coverage azimuth from [-π, π] to [-π/2, π/2] would narrow the sensor's coverage area in the azimuthal direction, potentially limiting its ability to detect and track targets over a wider range of azimuth angles resulting in increasing the RMSE. Here's how this change might impact the tracking system.

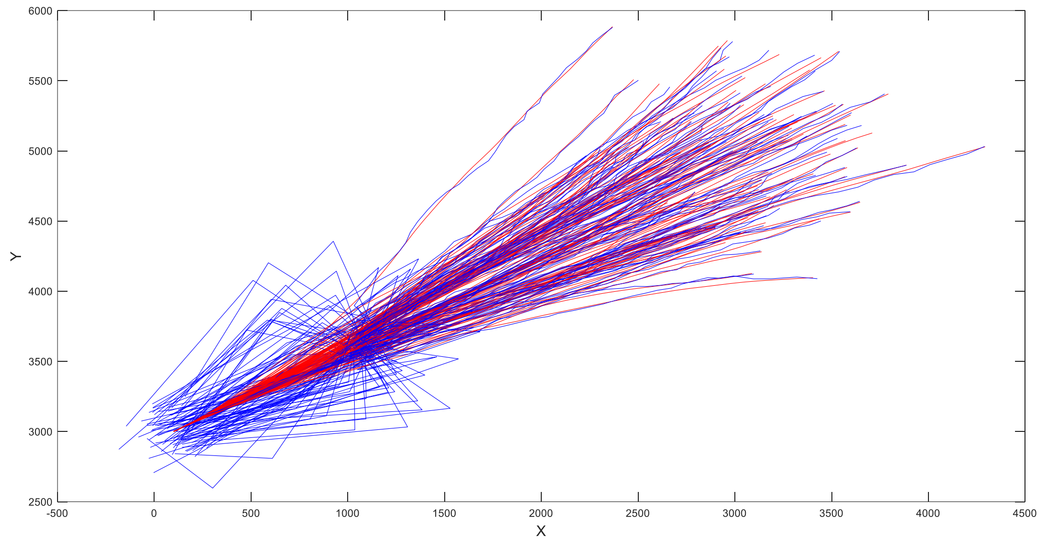

Figure 19: decreasing sensor coverage azimuth trajectories.

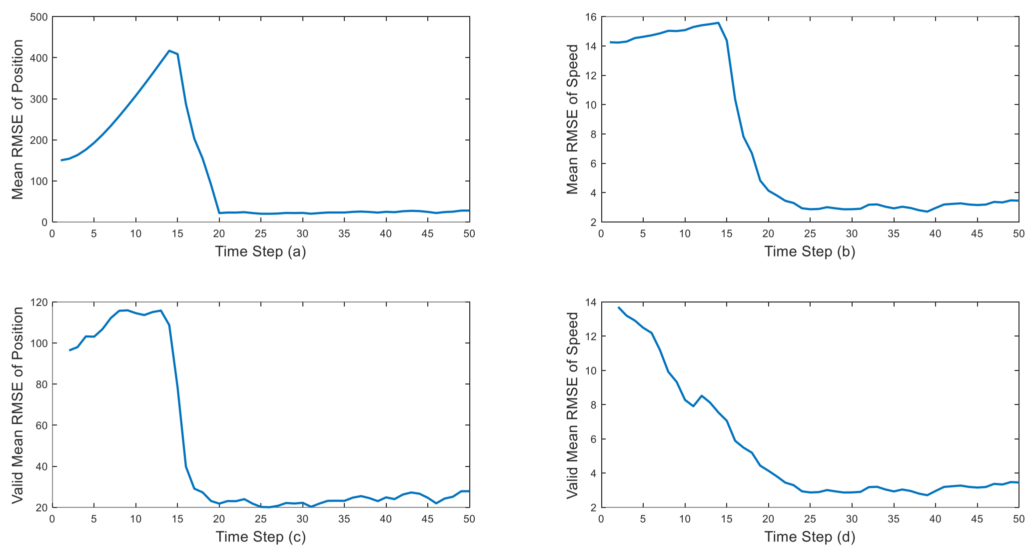

Figure 20: decreasing sensor coverage azimuth RMSE.

In this scenario, the target initially lies outside the sensor's coverage range, leading to difficulties in detection and tracking. As the target moves closer and enters the sensor's coverage area, there's a notable spike in RMSE, indicating tracking challenges due to the target's initial position. However, once the target is within the sensor's coverage, RMSE decreases as tracking accuracy improves with better measurements. This pattern highlights the influence of target proximity to the sensor on tracking performance, emphasizing the significance of coverage range in accurate target tracking.

The effect of increasing tracker covariance

Increasing the value of tracker covariance from $diag([100^2, 10^2, 100^2, 10^2])$ to $diag([200^2, 20^2, 200^2, 20^2])$ to represents a significant change in the initial covariance matrix used by the tracker. This matrix defines the initial uncertainty or error associated with the estimated target state. By doubling the variance values in the diagonal elements of the covariance matrix, we effectively increase the initial uncertainty in the target's position and velocity estimates. This adjustment implies that the tracker starts with a less confident estimate of the target's state, allowing for more flexibility in accommodating potential deviations from the true target trajectory. Consequently, the tracker may exhibit greater tolerance for initial tracking errors and uncertainties, which could result in smoother tracking performance, particularly during the early stages of the tracking process. However, it's important to note that this increased initial uncertainty may also lead to slower convergence and potentially larger tracking errors over time, as the tracker adjusts its estimates based on incoming measurements. However, as the tracking process progresses and more measurements are incorporated into the estimation, the tracker would gradually refine its estimates and reduce the RMSE values. Overall, adjusting tracker covariance to higher values introduces a trade-off between initial robustness and long-term tracking accuracy, highlighting the importance of carefully tuning this parameter to optimize tracking performance based on specific tracking requirements and environmental conditions.

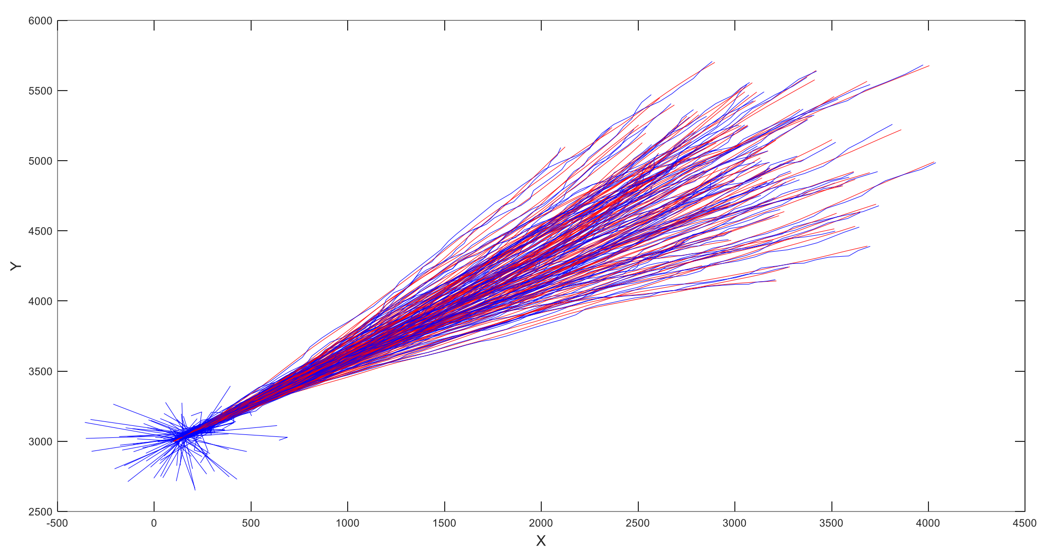

Figure 21: increasing tracker covariance trajectories.

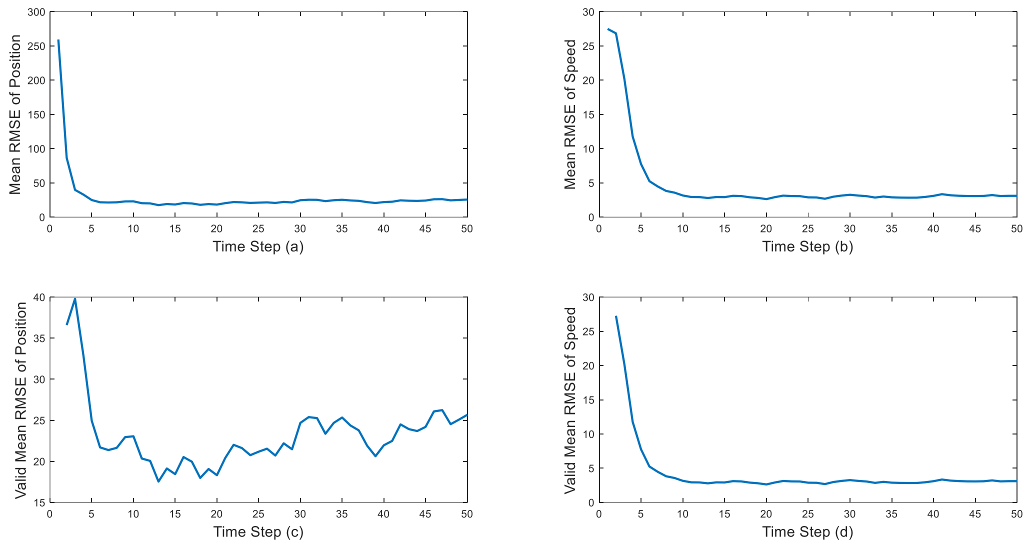

Figure 22: increasing tracker covariance RMSE.