**ECE 767**

**Multitarget Tracking and Multisensor Information Fusion**

Faculty of Engineering – McMaster University – Graduate Studies

Winter 2025

Professor: R. Tharmarasa (Thamas)

**Assignment 1 Report**

**Submitted by**: HaoJun(Henry) Zhao, zhaoh52, 400179164

**Submission Date**: February 24, 2025

**Contents**

[Problem Description 3](#_Toc190820030)

[Implementation Details 4](#_Toc190820031)

[Target Generation 4](#_Toc190820032)

[Measurement Generation 6](#_Toc190820033)

[Convert Measurement 7](#_Toc190820034)

[Data Association 8](#_Toc190820035)

[Simulation Results 9](#_Toc190820036)

[Discussion 9](#_Toc190820037)

[Appendix 10](#_Toc190820038)

[Target Generation 10](#_Toc190820039)

[Measurement Generation 11](#_Toc190820040)

[Convert Measurement 12](#_Toc190820041)

# Problem Description

In this assignment, the objective is to estimate a target position under a constant velocity model using a constant measurement Kalman filter (CMKF) with the presence of measurement from radar and false alarm in a simulated environment. This report assumes a single target movement under constant velocity movement. Secondly, one measurement comes from one target. Furthermore, the radar is stationary, and corresponding measurements return range and azimuth.

The overview of the assignment can be separated into the following steps:

1. **Target Generation**
2. **Measurement Generation**
3. **Covert Measurement**
4. **Data Association**
5. **Kalman Filtering**
6. **Performance evaluation**

Implementation details for those key steps are discussed in the following section. For data association, this assignment uses the nearest-neighbor method. Regarding the performance evaluation, this assignment considers the Monte Carlo process to evaluate the root mean square error (RMSE) regarding X position/Speed, and Y position/Speed across multiple test cases. The RMSE evaluation provides direct insight into tracker performance under different noise levels, for example, initial state guesses, initial covariance guesses, sensor measurement noise level, and so on. This report presents a detailed analysis of the problem description, implementation details for each function, results discussion, and RMSE performance evaluation of the simulated preliminary radar tracking system. This assignment can be used as a foundation for assignment 2 with multi-target and project for this course.

# Implementation Details

In this section, implantation details of key functions which are covered in the above section are discussed. Each function plays an important role in this tracking problem. These discussions include detailed reasons to show why trackers need this function and mathematic derivation for each function. The completed code with proper comments is attached to the appendix section.

## Target Generation

To begin the simulation of the tracker, the mandatory condition is to have a target to track. Without a proper target, the tracker is not necessary. In this simulation, a nearly constant velocity model is considered. This is one of the fundamental motion models in the target tracking field. In this model, it assumes the target moves in nearly constant velocity in other words acceleration is near to 0. To represent this model accurately, the acceleration is treated as random process noise which is independent of and axis. The following formula is based on state-space representation, which provides a mathematical framework for predicting the future state of the target given its current state and system dynamics.

Where is the state transition matrix which represents how the target state changes over each time step. is the process noise coupling matrix that determines how random noise in this case random noise is accelerated in the and axis and affects the next state. is the process noise acceleration which represents the uncertainty of this model. Note here that the and axis acceleration noise are independent therefore the correct implementation method is to use the MATLAB randn command to generate two normally distributed random numbers. and represent the target's next state and the target's current state. this assignment includes four states which can be represented as follows:

According to physics, displacement in axis for a target under constant velocity and target velocity in axis can be represented as follows:

Where and represents the current position and next step position, and represents current velocity and next step velocity.

Therefore, the and can be represented as follows:

The completed nearly constant velocity model is as follows:

The final important output in the target generation is the process noise covariance . The covariance is critical to the further development of Kalman filter and data association. The covariance matrix mainly focuses on representing uncertainty in the motion model which allow the filter algorithm to decide the trust level between the measurements and prediction. The formula of the covariance matrix is the following:

## Measurement Generation

The second key function is to generate measurements. The measurements are based on sensor parameters such as sensor position, range, azimuth, and corresponding noise, which are defined in the assignment manual. In this assignment, the sensor is stationary and one measurement can only be generated by one target. One important parameter is the probability of generating measurements. In the implementation, **randn function** need to be used to generate a random number from 0 to 1 to check whether the sensor will generate measurement or not. The range and azimuth calculation follows the formula:

+ Noise

+ Noise

The measurement generation is straightforward but atan2 must used here for azimuth generation in order to return the correct quadrant. Despite the range and azimuth measurement for the target. The sensor also needs to generate false alarms. number of false alarms follows a Poisson distribution with the mean number of events λ and the values of false alarms follow a uniform distribution. The formula for λ is the following:

In order to generate values of false alarms following a uniform distribution, **poissrnd(λ)** needs use here. Based on the simulation, the number of false alarms is around 9 to 10 for each time step. The position of false alarm does not have a formal formula. They are all personal choices. In this report, the formula for calculating the false alarm is the following:

This formula makes sense because it considers false alarms' randomness and ensures they are in the correct boundary conditions. The completed code with detailed comments on the generated measurement code is attached in the appendix.

## Convert Measurement

This function is actually part of generating measurements but this report separates it out to make sure the hierarchy can be illustrated much clearer. The most important reason behind converting measurement is that this report does not use an extended Kalman filter since EKF handles nonlinearities by linearizing the measurement by applying Jacobian function calculation. therefore it must convert the original measurements in range and angle into cartesian position in order to match the original state representation. This step is crucial for maintaining consistency in your estimation framework. The next important point is that unbiased conversion has to be used here. The naïve conversion introduces bias . The formulas are used in this report are the following:

The next key output for this function is the corresponding covariance matrix because even though the conversion itself is unbiased the distribution of errors changes. The reason behind it is that non-linear conversion is used here, it makes the error more dependent. For each measurement, the covariance matrix is as follows:

Where and

The completed code with detailed comments on the convert measurement code is attached in the appendix.

## Data Association

Data association plays an important role in tracking multiple targets with imperfect sensors. Although this assignment handles one target and one sensor case, data association is still very important for the following reasons. In this assignment, the sensor is not perfect. There is a 90% chance for the sensor to generate measurement and at the same time, it generates a false alarm. If those measurements directly go into the filtering algorithm, the filter cannot identify between true measurements and false alarms. Second, the filter has no prior knowledge of whether the sensor successfully generated a measurement at a given time step. If no measurement is available, the filter may incorrectly update the state based on outdated or erroneous information.

By implementing data association, we ensure that the filter correctly identifies valid measurements, improves estimation accuracy, and enhances overall tracking performance. According to the requirement in this assignment, the method implemented here is called nearest neighbor (NN). The concept of nearest neighbor is that the predicted value is generated and calculated based on the last fused estimation. The distance formula is the following:

is used to compare the gate size which is set in the parameters. is the innovation covariance. If is smaller than the pre-set gate value, that the corresponding measurement is considered as true measurement, otherwise it is a false alarm. The completed code with detailed comments on the data association code is attached in the appendix.

## Kalman Filtering

Kalman Filtering is one of the most important functions in this assignment. From the previous discussion, data association filters out all the false alarms and only keeps the true measurement. The implementation for the Kalman filter can be simply concluded as follows:

Where is the famous Kalman gain, is the predicted state, is the predicted state’s covariance, is the predicted measurement. From the data association section, it is clear that predictions need to be made therefore predictions are all made before the Kalman filter function. In order to initialize the Kalman filter, an initial guess shall be made here as well. Although there is no specific formula related, it will be covered quickly due to the characteristics of the Kalman filter. The initial guess for state and covariance is the following:

Since this assignment needs to implement a Monte Carlo run, the formula for initial state guess can make sure every run initial state guess always has different values. The completed code with detailed comments on the Kalman filtering code is attached in the appendix.

## Performance Evaluation

There are different ways for performance evaluation for example latency, RMSE, false alarm rate, and so on. In this assignment, RMSE which covers distance RMSE, velocity RMSE, and corresponding valid track for the overall Monte Carlo run is considered as the only performance evaluation indicator. The formula of RMSE is the following:

Where denotes the number of Monte Carlo runs. The biggest issue during the implementation is that there are opportunities to lose track. Since Pd is not equal to 1 which means there are possibilities for the sensor to not generate any detection. In this case, if one of the false alarms location is accidentally close enough to the target, it will be considered as a measurement even though it is not. The consequence of false association will cause a loss of track directly. From RMSE's point of view, there is a peak when losing track happens. The limiting method shall be considered to avoid this issue. The performance evaluation gate is applied in this assignment. The idea behind that is the algorithm only considers errors within the evaluation gate into RMSE calculations otherwise considers errors as zero at that time step. If the error is larger than the performance evaluation gate, it also means the algorithm loses track at that time step, zero is assigned to a valid run matrix which records whether lost track or not for each time step in each Monte Carlo run.

# Simulation Results

# Discussion

# Appendix

## Target Generation

A computer screen shot of a program

AI-generated content may be incorrect.

## Measurement Generation

A screenshot of a computer

AI-generated content may be incorrect.

## Convert Measurement

A close-up of a computer screen

AI-generated content may be incorrect.

## Data Association

A screenshot of a computer code

AI-generated content may be incorrect.