

**Multi-Sensor Fusion for
Autonomous Driving (AD) and
Advance Driver Assistance System (ADAS) functions**

Concept and Design Document

Authored By:
UDIT BHASKAR TALUKDAR
Date : 01-01-2021

Table of Contents

1. INTRODUCTION	5
2. PROJECT SCOPE	5
3. DOCUMENT OVERVIEW	5
4. ASSUMPTIONS.....	5
5. ARCHITECTURE	6
5.1 Fusion Architecture Overview.....	6
5.2 High-Level Fusion Module Architecture	7
5.3 Detailed Module Architecture / Execution Flow	8
5.3.1 Pre Processing	8
5.3.2 Radar Track Fusion.....	9
5.3.3 Camera Track Fusion.....	10
5.3.4 Radar and Camera Track Fusion	11
6. SENSOR LAYOUT.....	12
6.1 Introduction	12
6.2 Sensor Parameters.....	13
6.2.1 Radar Sensor Parameters.....	13
6.2.2 Camera Sensor Parameters.....	14
6.3 Measurement Parameters	14
6.3.1 Radar Measurement Parameters.....	15
6.3.2 Camera Measurement Parameters.....	15
6.4 Measurement Noise Covariance for Radar.....	16
6.5 Measurement Noise Covariance for Camera.....	20
7. SENSOR FOV GRID	23
7.1 Introduction	23
7.2 Grid Concepts.....	25
7.2.1 Grid Definition.....	25
7.2.2 Grid Cell Index Computation from a point.....	26
7.2.3 Grid Centre Computation from Grid Cell Index	27
7.2.4 Grid Cell Coordinates Transformation from Vehicle Frame to Sensor Frame	28
7.2.5 Region Confidence Computation	28
7.3 High Level Design	29
7.4 Detailed Design	30
7.4.1 Flowchart of Grid Cell Index Computation from a point.....	30
7.4.2 Flowchart of Sensor FoV Overlap and Region Confidence Computation - part 1	31
7.4.3 Flowchart of Sensor FoV Overlap and Region Confidence Computation - part 2	32
7.5 Visualization	33
8. SPATIAL ALIGNMENT.....	34
8.1 Introduction	34
8.2 Coordinate Transformation Concepts.....	34
8.3 Sensor Measurement Structure.....	36
8.3.1 Radar Measurement Matrix.....	36
8.3.2 Camera Measurement Matrix.....	37
8.4 Plots and Analysis.....	38

9. OBJECT FUSION.....	39
9.1 Introduction	39
9.2 Object Fusion Concept & Design.....	39
9.3 Sensor Measurement Clustering.....	40
9.3.1 Module Architecture/Execution Flow.....	40
9.3.2 High - Level Design Of Radar Measurement Clustering	41
9.3.3 High - Level Design Of Camera Measurement Clustering.....	42
9.3.4 Detailed Design : Simplified DBSCAN Clustering.....	43
9.3.5 Detailed Design : NN Clustering	46
9.3.6 Plots.....	47
9.4 State Prediction.....	48
9.4.1 Introduction	48
9.4.2 Formulation.....	48
9.5 Measurements To Predicted Track Ellipsoidal Gating	49
9.5.1 Introduction	49
9.5.2 Module Architecture/Execution Flow.....	50
9.5.3 Formulation.....	51
9.5.4 Plots.....	52
9.6 Single Sensor Measurement To Track Data Association.....	54
9.6.1 Introduction	54
9.6.2 Association Probability Computation	55
9.6.3 Single Sensor Data Association Flowchart	56
9.7 Multiple Sensor Fusion	57
9.7.1 Homogeneous Sensor Fusion Flowchart.....	57
9.7.2 Heterogeneous Sensor Fusion Flowchart	58
9.8 Plots.....	59
9.8.1 Radar Track Fusion Estimates	59
9.8.2 Camera Track Fusion Estimates	61
9.8.3 Radar and Camera Track To Track Fusion Estimates	63
10. TRACK MANAGEMENT	65
10.1 Overview/Requirements.....	65
10.2 Deterministic Finite State Automata as Track State Machine	66
10.2.1 Track States	66
10.2.2 Track State Transitions.....	67
10.2.3 Current Implementation	68
10.3 Plots :	69
10.3.1 Track ID & Track Status	69
10.3.2 Sensor Catch Status for TV1 : time plot of sensor contributions to the estimation.....	71
10.3.3 Sensor Catch Status for TV2 : time plot of sensor contributions to the estimation.....	72
11. CONCLUSION	73
12. References	74

This Page is intentionally Left Blank

1. INTRODUCTION

This project '**Multi-Sensor Fusion for Autonomous Driving (AD) and Advance Driver Assistance System (ADAS) functions**' is the project 1 of a series of 4 Sensor Fusion Project.

The other 3 projects are as follows :

- Road Model Fusion by Multiple Cameras
- Occupancy Grid Map Fusion of Multiple Radars and Cameras.
- Localization & Mapping by Multiple Radar and Camera Fusion.

2. PROJECT SCOPE

Scope of the project is to design & develop a proof of concept (POC) of sensor fusion system, which fuses camera and radar sensor information to perform Object Tracking.

3. DOCUMENT OVERVIEW

This document provides the reader with a detailed description on system overview, mathematical formulations, implementation steps in the form of flowcharts, and relevant plots for visualizations and analysis of Multi modal Sensor Fusion System.

4. ASSUMPTIONS

The following assumptions are made in the **version 1** of the project. (**Current Version**)

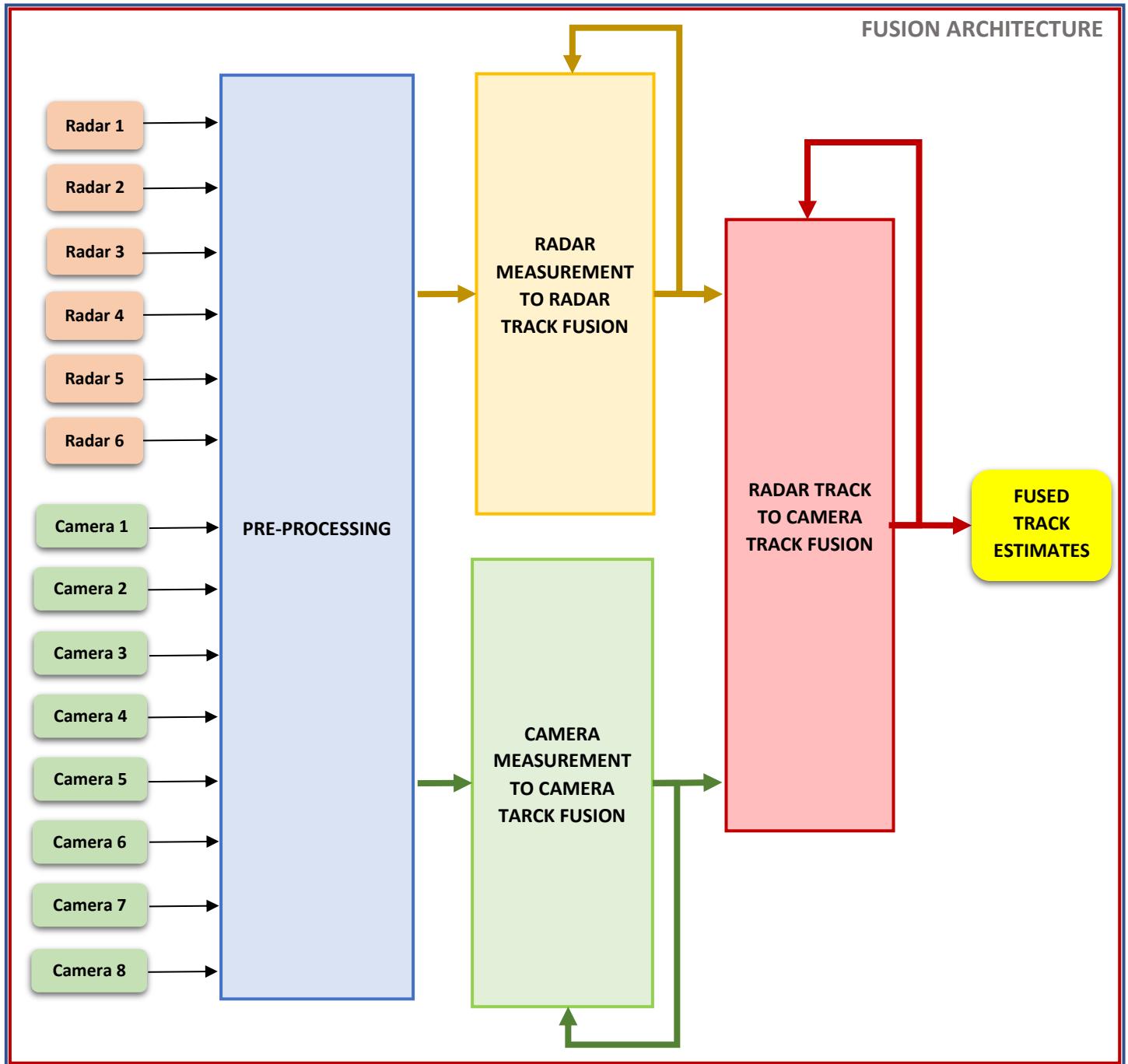
- The number of radar sensors considered is 6
- The number of camera sensors considered is 8
- The sensors are synchronized (Temporally aligned)
- Range and velocity parameters of the radar sensor measurements shall be in cartesian coordinates. The position and velocity vector of the measurement shall have longitudinal and lateral component.
- Both Radar and Camera measurements shall have position and velocity.
- Only the kinematic parameters from the sensor measurements are used for fusion which includes longitudinal and lateral range components, longitudinal and lateral velocity components.
- The measurements from each of the sensors shall be with respect to its own sensor frame.
- The Fusion System shall be validated with simulated data.

The following shall be considered for **version 2** of the project (**Future Version**)

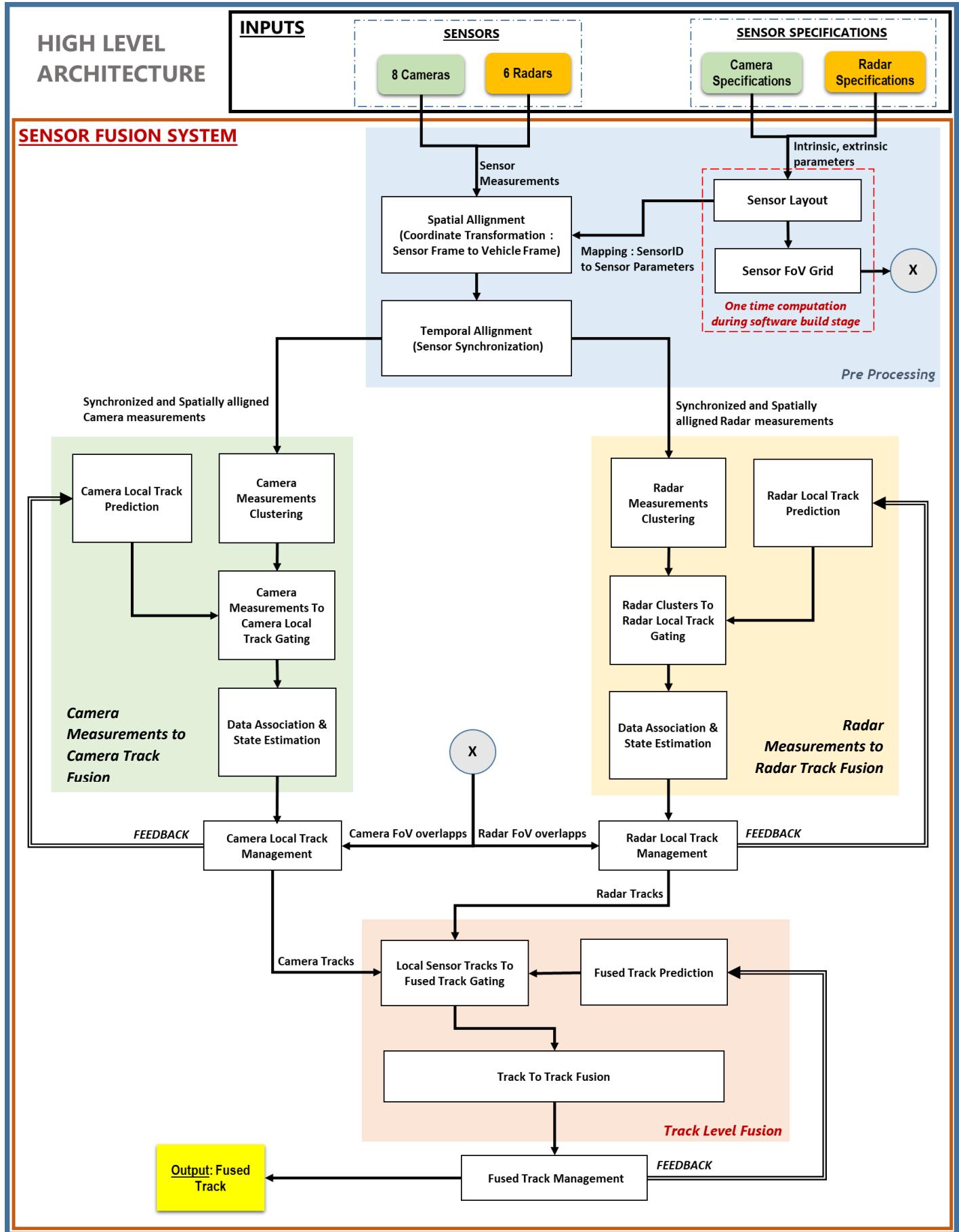
- The number of radars and cameras shall be kept as a variable.
- The sensors shall be unsynchronized (Temporally misaligned)
- Position and velocity parameters of the radar sensor measurements shall be in polar coordinates.
- For radars, kinematic measurement parameters shall be radial range, radial range rate and azimuth.
- For cameras, the kinematic measurement parameters shall be lateral and longitudinal range.
- For cameras, the longitudinal and lateral velocity shall NOT be considered.
- From Radar, reflection intensity parameters shall be utilized for fusion.
- For Camera, bounding box and classification information shall be utilized for fusion.
- The Fusion System shall be validated with the open sourced NuScenes data set.
- Extended Object Tracking Techniques based on Random Finite Set based methods shall be incorporated in the fusion library.
- Integration of IMM (Interacting Multiple Model Filter) with the fusion framework.
- Bayesian Dynamic Grid Based fusion techniques shall be explored.

5. ARCHITECTURE

5.1 Fusion Architecture Overview

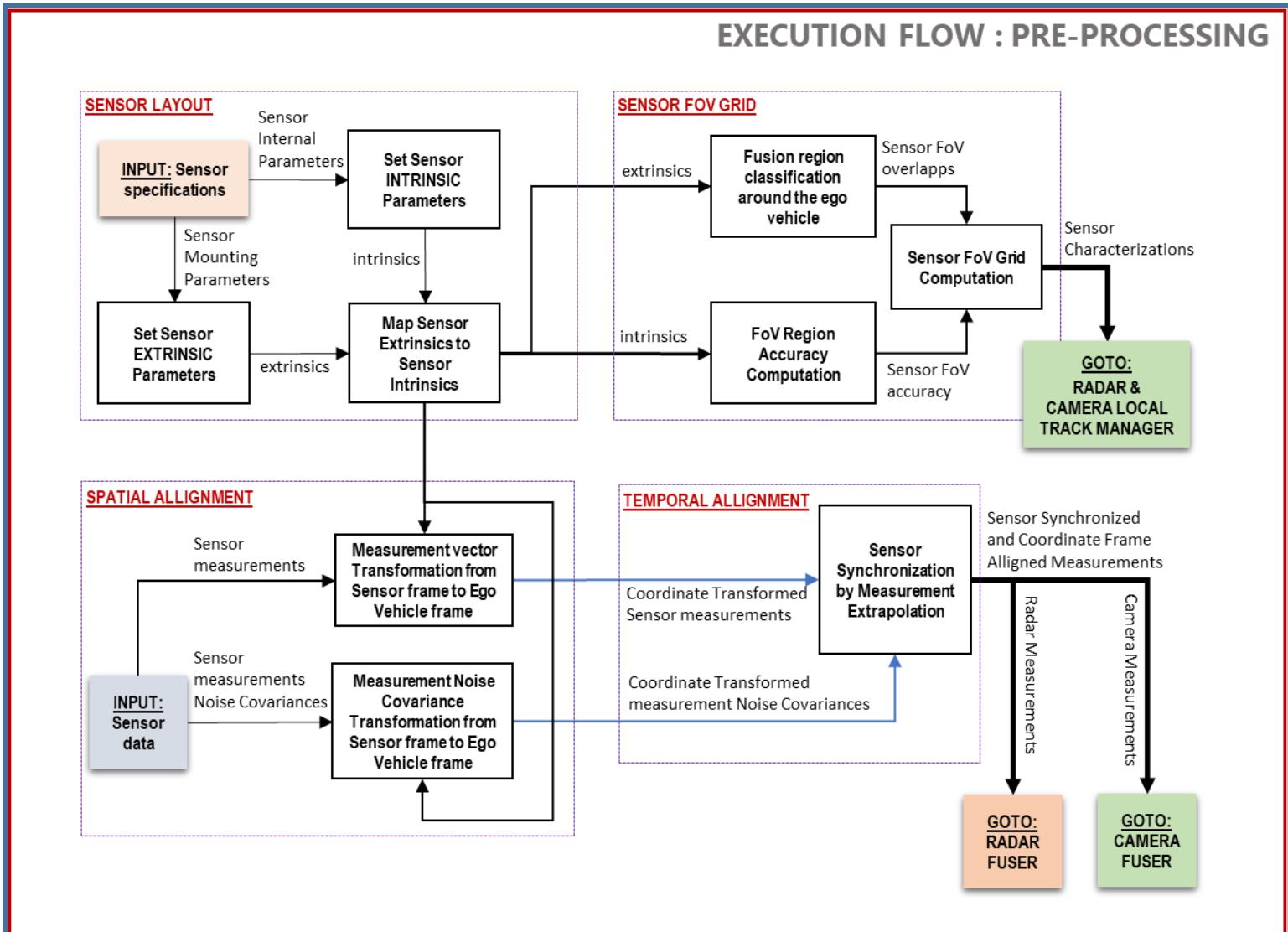


5.2 High-Level Fusion Module Architecture

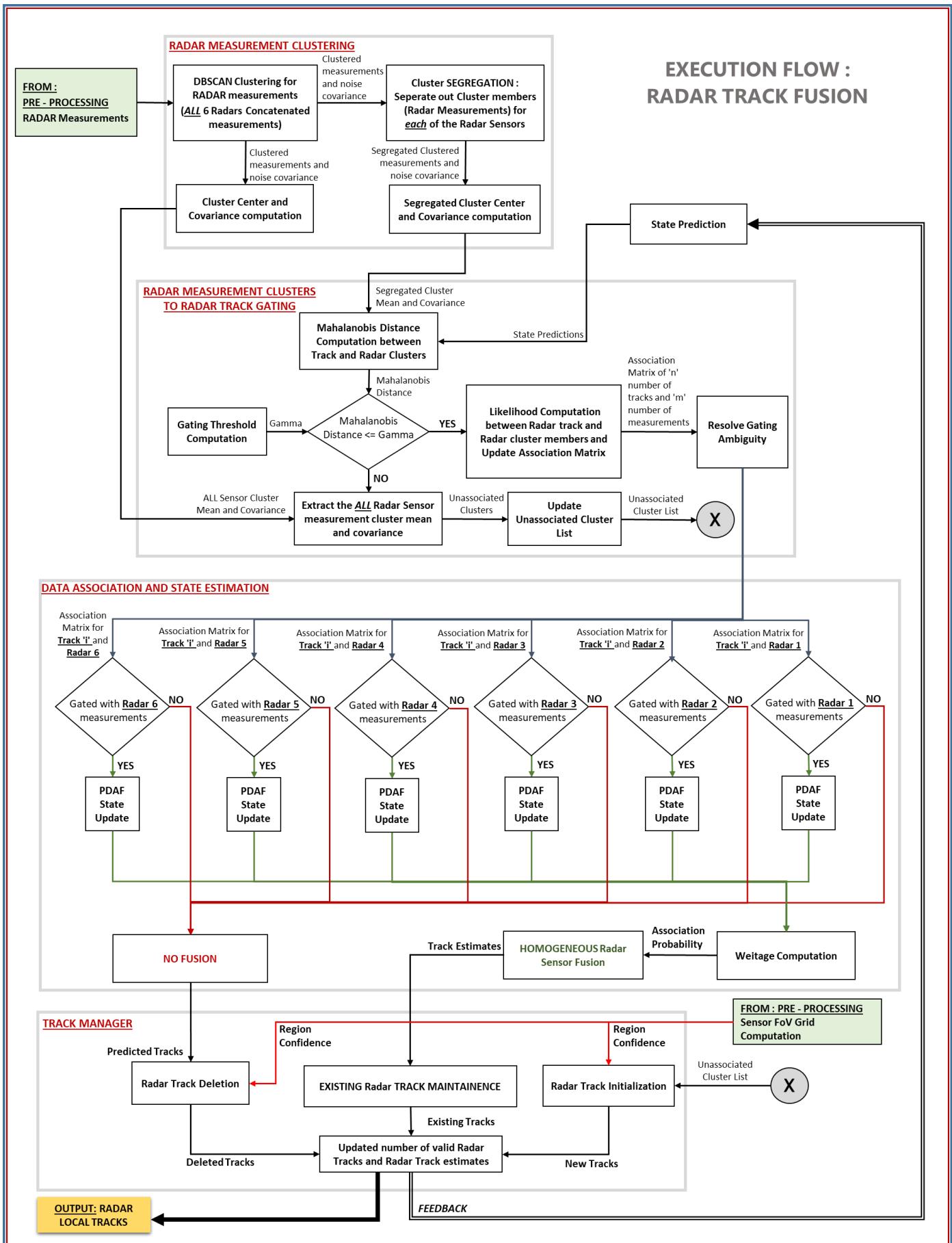


5.3 Detailed Module Architecture / Execution Flow

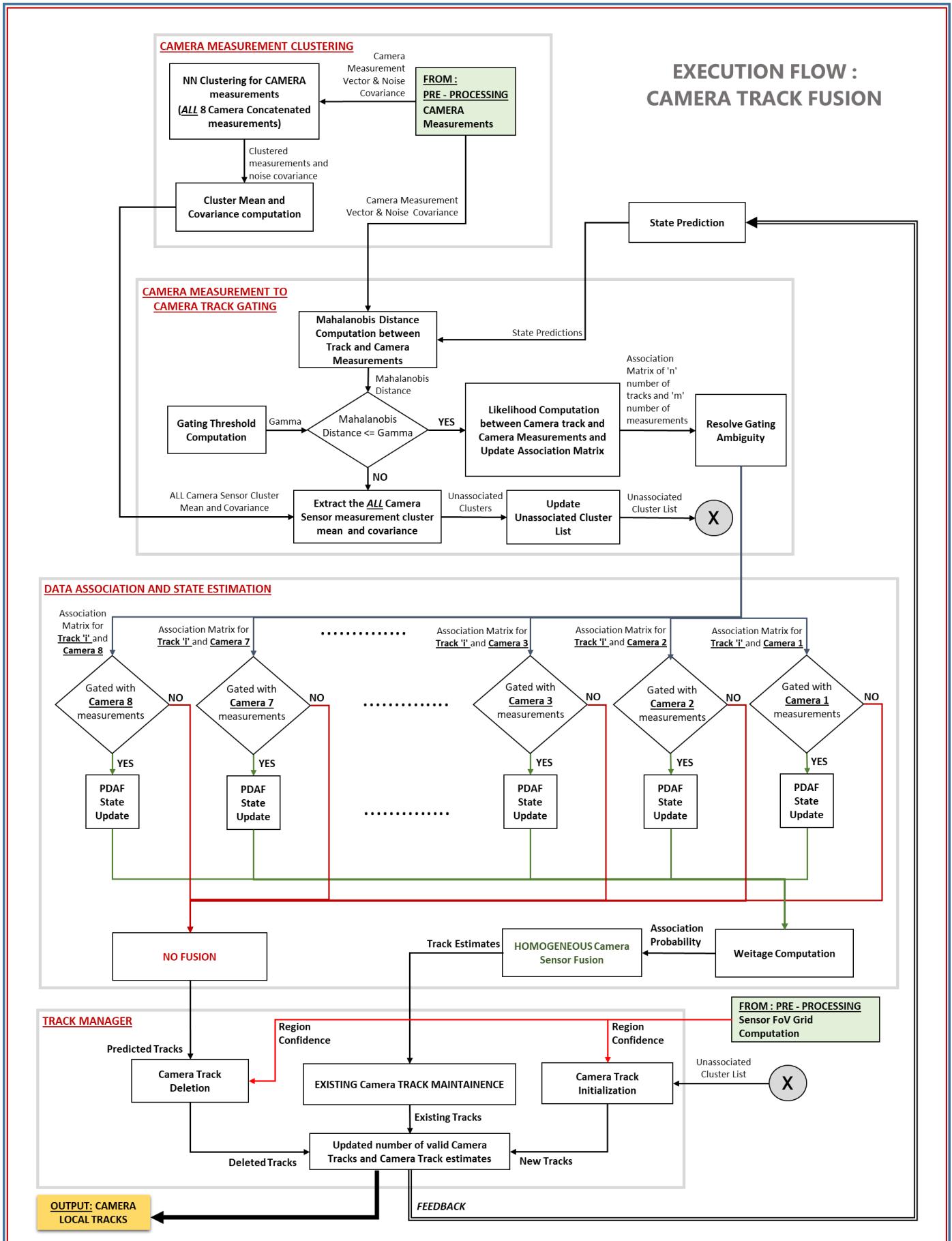
5.3.1 Pre Processing



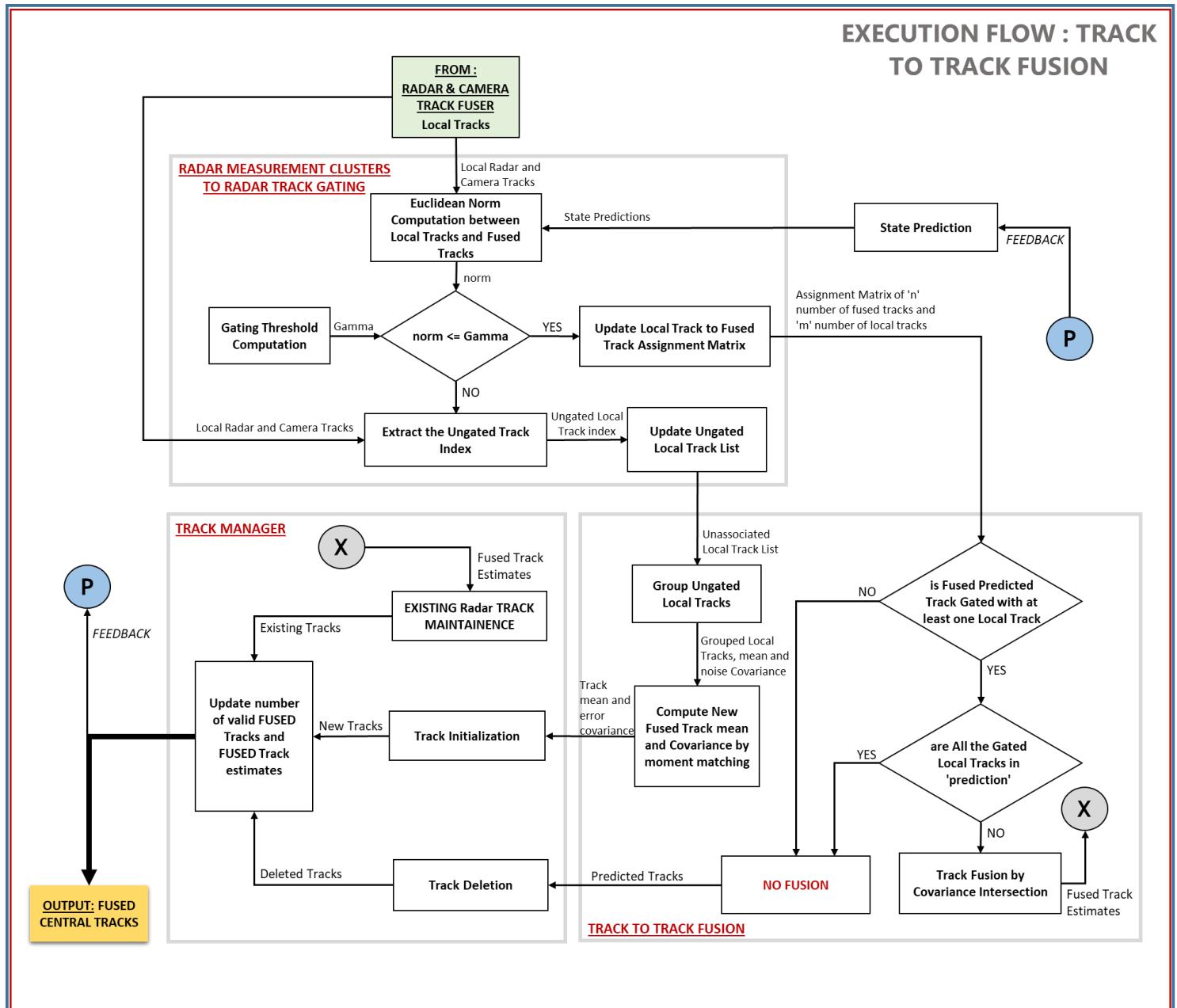
5.3.2 Radar Track Fusion



5.3.3 Camera Track Fusion



5.3.4 Radar and Camera Track Fusion



6. SENSOR LAYOUT

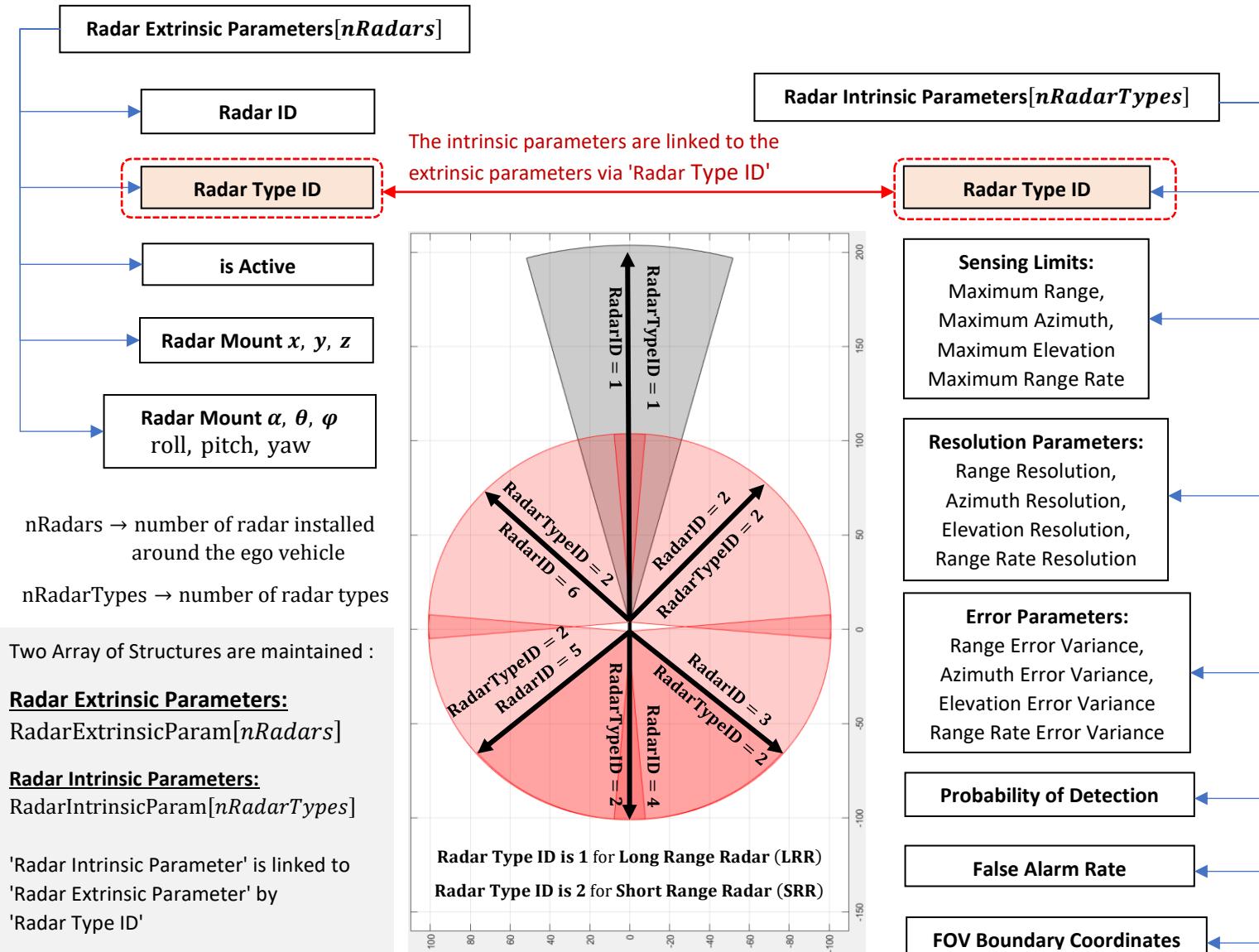
6.1 Introduction

Sensor Layout is responsible for creating a many to one mapping between the sensor extrinsic and intrinsic parameters. In this project 6 Radars , 8 Cameras Sensor Suite is considered for Object Tracking feature. Each of the radars considered here can be of two types : Long Range Radar (LRR) and Short Range Radar (SRR) , and each of the camera can be : Narrow FoV (NFOV) and Wide FoV (WFOV). Thus each of the sensors can have different modes like Radar, Camera and each of the sensor modalities can have subtypes like LRR, SRR in case of Radar and NFOV , WFOV in case of Camera.

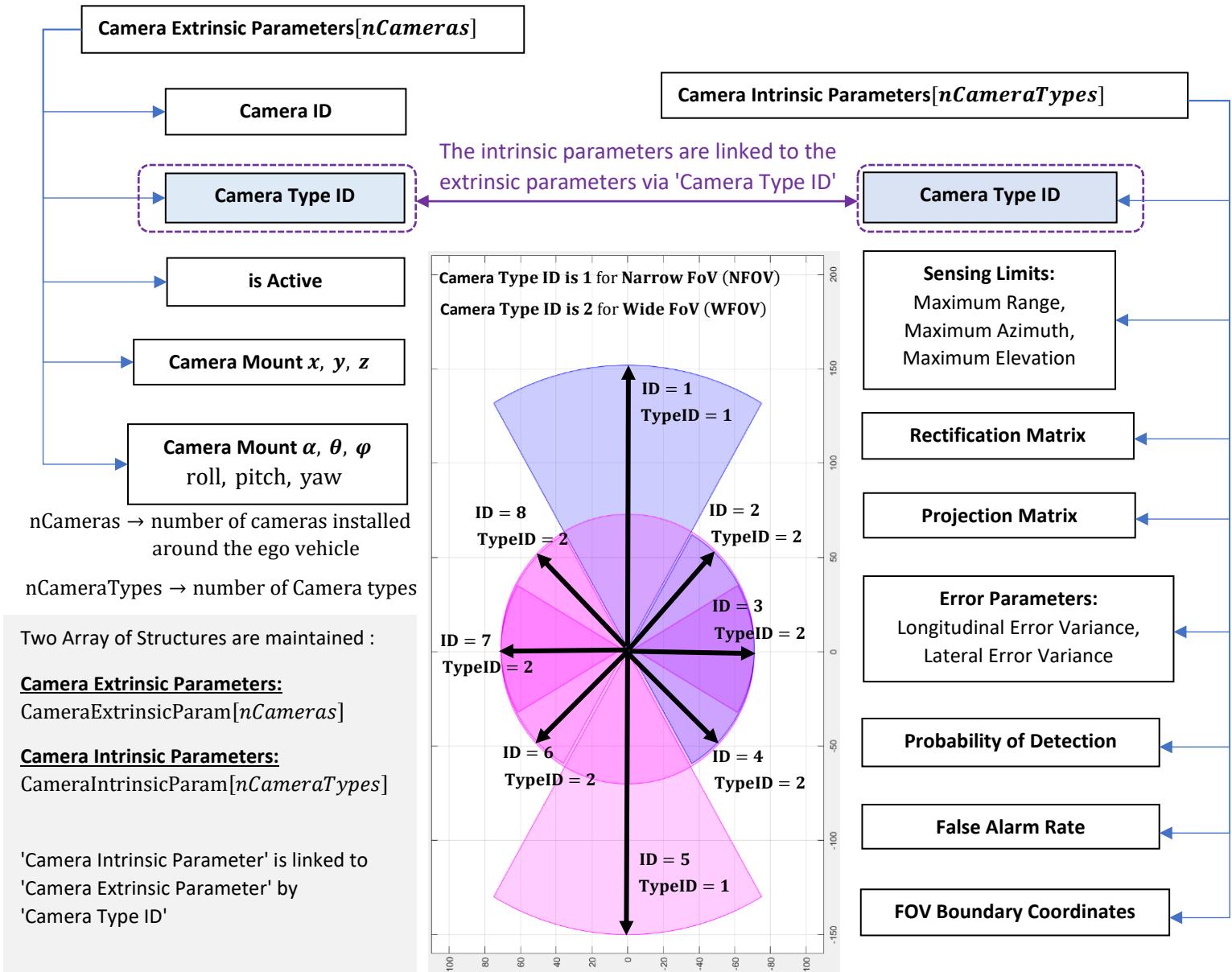
Considering the above variations in installed sensor types, an efficient data structure is required such that for each of the measurement received from the sensors, the appropriate sensor parameters can be identified and correct transformations w.r.t sensor parameters can be applied. Sensor Layout Module is responsible for creating and maintaining such data structure.

6.2 Sensor Parameters

6.2.1 Radar Sensor Parameters



6.2.2 Camera Sensor Parameters

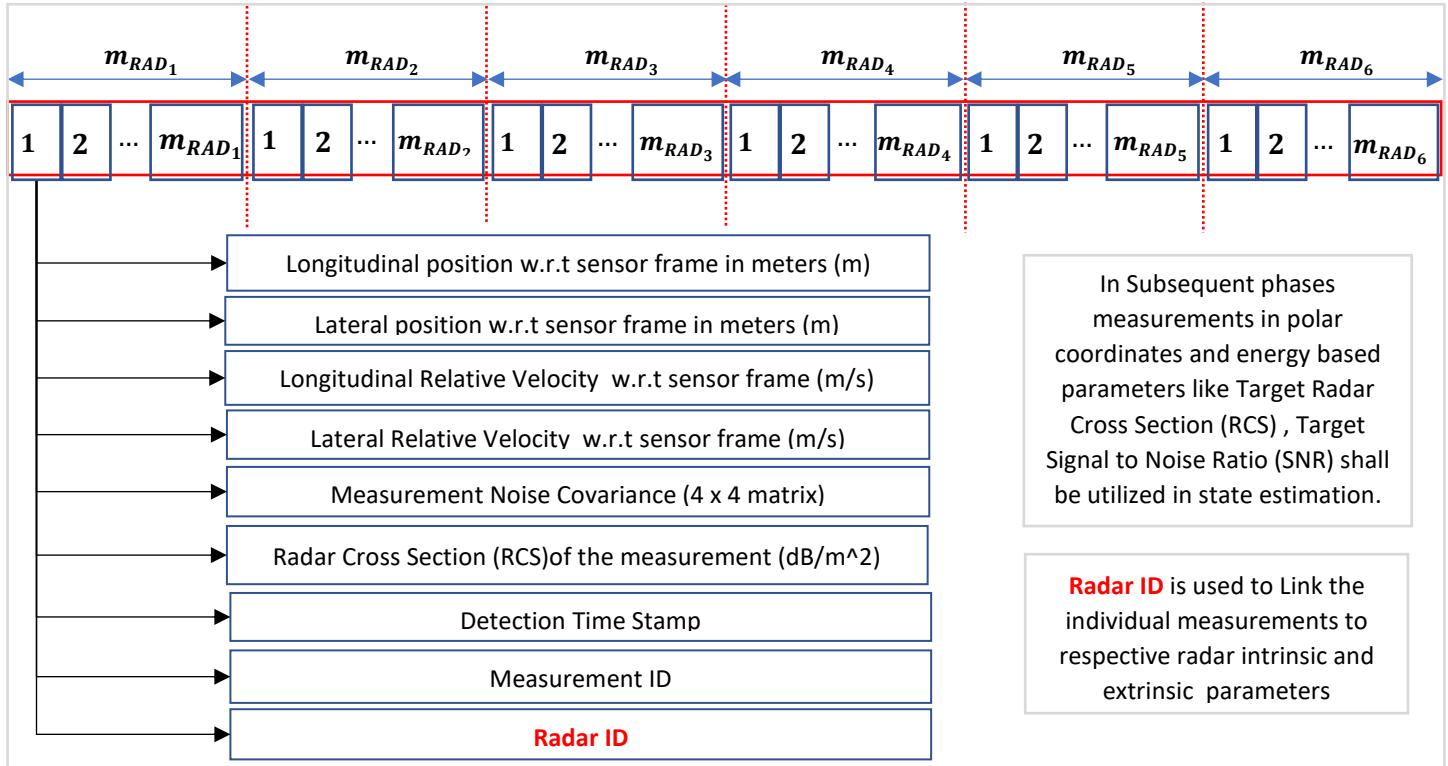


6.3 Measurement Parameters

Camera and radar shall have different set of measurements. Some of the measurement parameters might be common and some might be unique to a particular sensor type. Considering the complementary nature of these two sensor modalities, a combination of these parameters has the capability of improving the consistency and accuracy of an estimate.

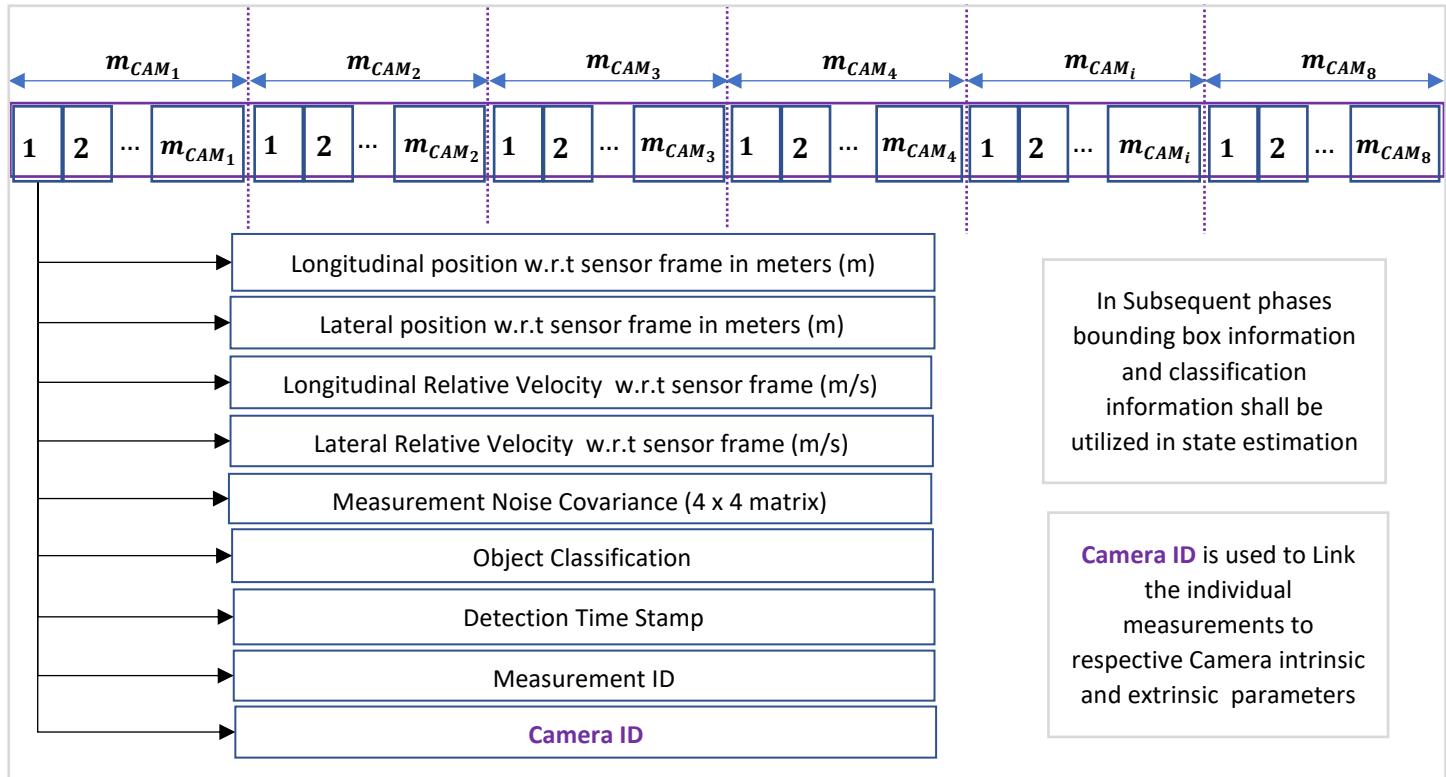
6.3.1 Radar Measurement Parameters

For each of the radars i the input is assumed to be in the form of an array of structure of size m_{RAD_i}



6.3.2 Camera Measurement Parameters

For each of the cameras i the input is assumed to be in the form of an array of structure of size m_{CAM_i}



6.4 Measurement Noise Covariance for Radar

The below description is for lateral and longitudinal position measurement errors. Similar rational shall be applied for determining error in measurement longitudinal and lateral velocity by considering errors in doppler measurements from radar specification sheet.

Typically the Measurement noise variances are reported in the sensor manufacturer's specification sheet. For Radar the measurement noise is reported as variance in range and azimuth σ_{range} and σ_{azimuth} . Assuming that the errors in range and azimuth is uncorrelated , the measurement noise covariance can be represented in the polar coordinate as follows.

$$R_{r,\theta} = \begin{pmatrix} \sigma_{\text{range}}^2 & 0 \\ 0 & \sigma_{\text{azimuth}}^2 \end{pmatrix}$$

In this project the sensor measurements are in cartesian coordinates, thus the above covariance matrix needs to be transformed to cartesian space. Transforming the measurement noise covariance from polar to cartesian has two challenges.

- Since the polar to cartesian conversion is a non-linear operation, the measurement noise covariance cannot be linearly transformed from polar to cartesian. The non-linear function needs to be linearized at the measurement mean value so that the measurement noise can be represented as a gaussian in cartesian space.
- The measurement noise covariance is constant in polar coordinate space. Once the covariance is transformed to cartesian coordinate space, it shall no longer be constant in cartesian. The measurement noise covariance shall vary in volume with lateral and longitudinal positions. For avoiding issues in filter convergence, measurement noise covariance whose ellipsoidal volume is constant is often desired.

The subsequent section shall discuss on the methods to address the above issues.

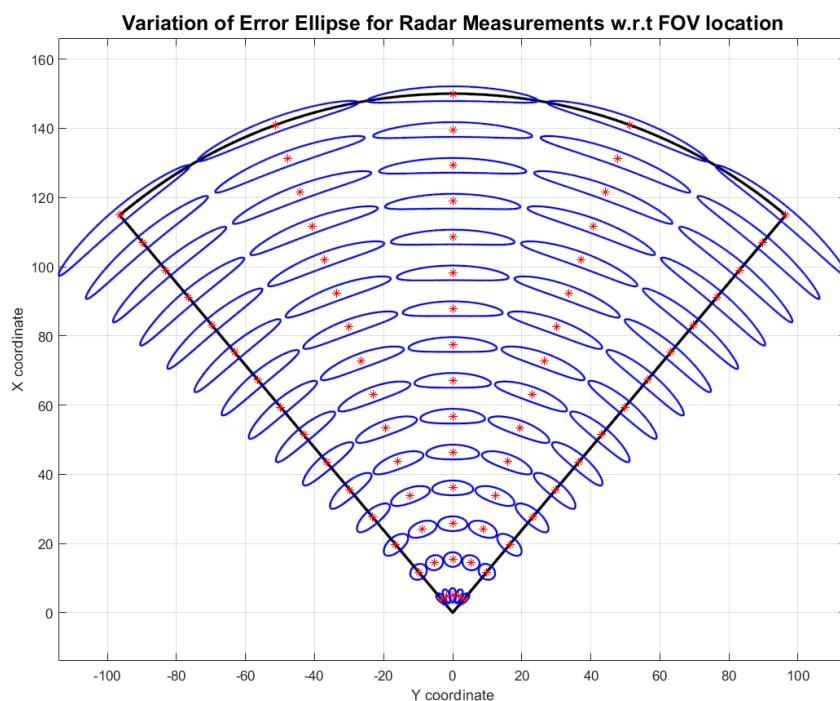


Figure : Variation in the area of Error Ellipse for different locations within FOV

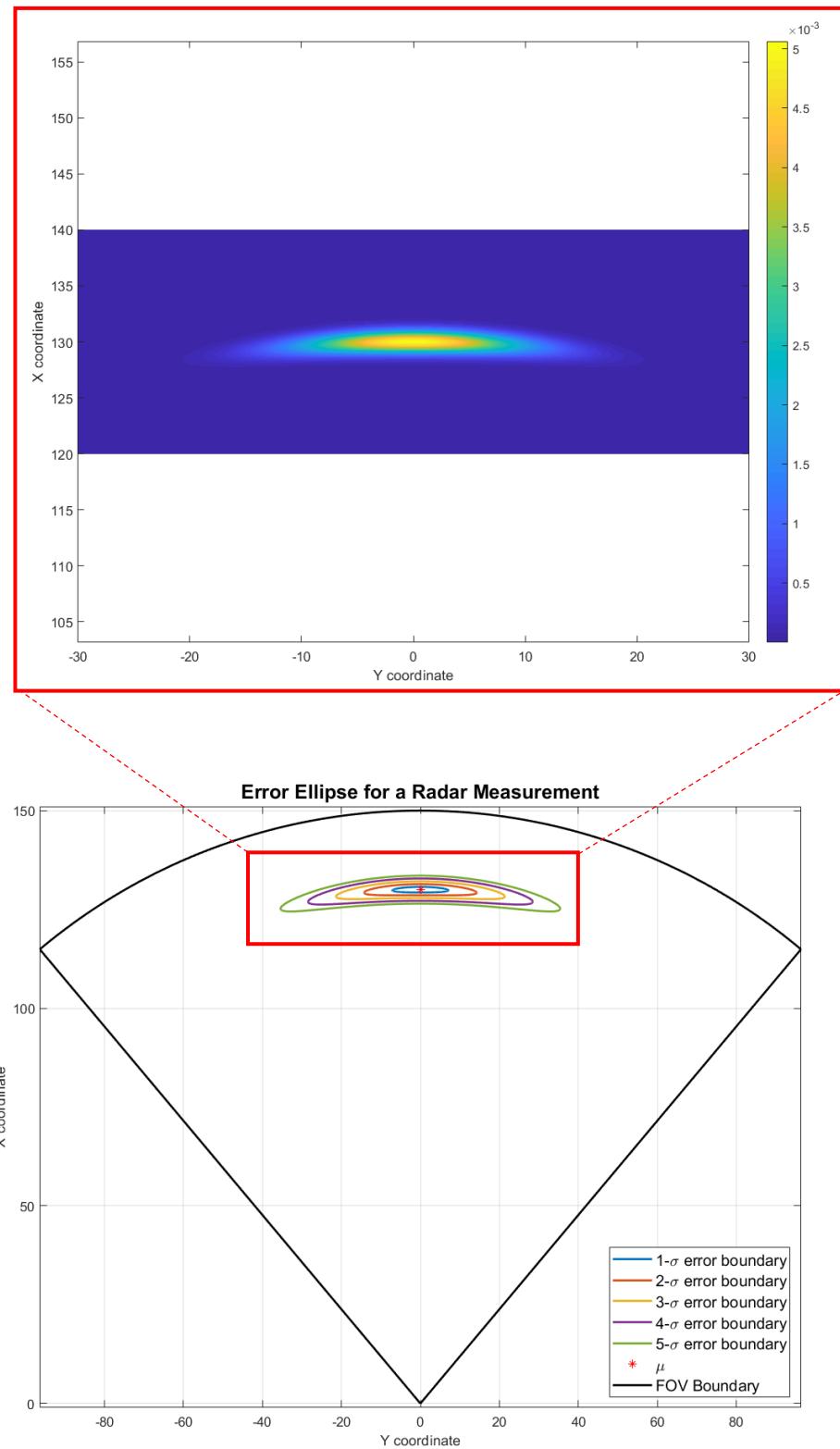


Figure : Effect of nonlinearity in polar to cartesian conversion - the error ellipse when plotted in cartesian space resembles a 'banana' like shape.

The measurement noise covariance is transformed from polar to cartesian by linearization using jacobian and then it is bounded and transformed to preserve correlation in error.

Linear Transformation of Measurement Noise covariance from polar to cartesian

$$\left. \begin{array}{l} x = r\cos\theta \\ y = r\sin\theta \end{array} \right\} \text{Polar to Cartesian Transformation (equation 1)}$$

$$\left. \begin{array}{l} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{pmatrix} \begin{pmatrix} r \\ \theta \end{pmatrix} \\ \Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix} \begin{pmatrix} r \\ \theta \end{pmatrix} \end{array} \right\} \text{linearization of equation 1 by jacobian (equation 2)}$$

$$\left. \begin{array}{l} R_{x,y} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix} R_{r,\theta} \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}^T \end{array} \right\} \text{Computation of Measurement Noise covariance from polar to cartesian (equation 3)}$$

Upper Bound on Transformed Measurement Noise covariance

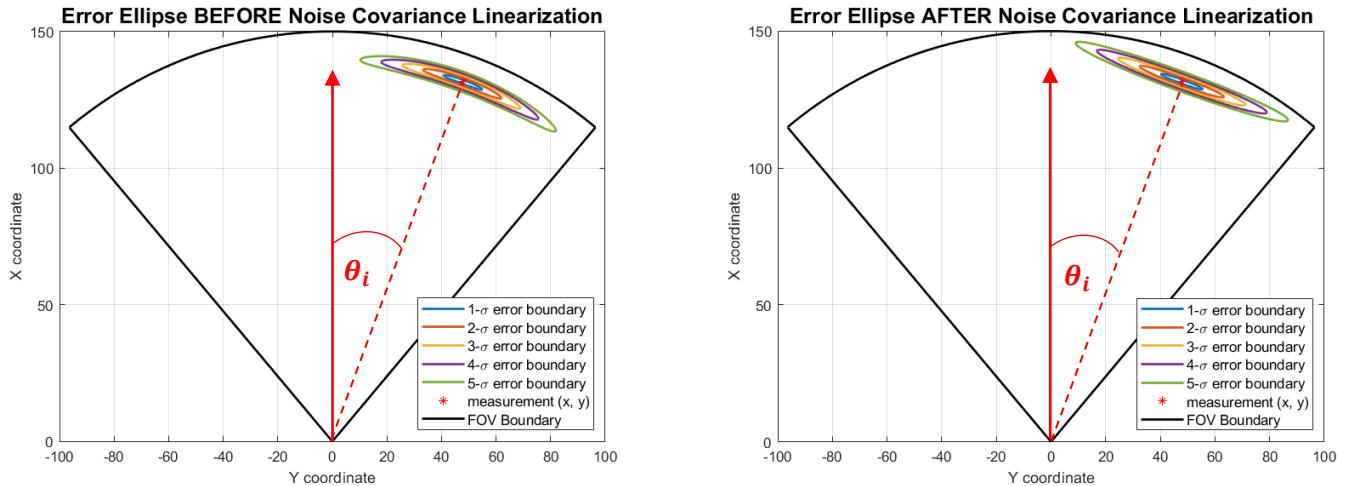
$$\left. \begin{array}{l} \text{Let } r = \alpha \\ \text{Let } \theta = 0 \end{array} \right\} \text{Measurement noise covariance is bounded at } r = \alpha, \text{ where } \alpha \text{ is tunable}$$

$$\left. \begin{array}{l} R_{x,y} = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} R_{r,\theta} \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix}^T \\ \Rightarrow R_{x,y} = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \\ \Rightarrow R_{x,y} = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & (\alpha\sigma_\theta)^2 \end{pmatrix} \end{array} \right\} \text{Measurement noise covariance Upper Limit}$$

Rotational Transformation to preserve correlated errors in the Transformed Measurement Noise covariance

$$\left. \begin{array}{l} \text{Let } z_i = (x_i, y_i)^T \\ \theta_i = \tan^{-1} \frac{y_i}{x_i} \end{array} \right\} \text{Radar measurement 'i' w.r.t sensor frame}$$

$$R_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & (\alpha\sigma_\theta)^2 \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}^T \right\} \text{Measurement noise covariance for measurement } z_i$$



$z_i \rightarrow$ Radar measurement 'i' w.r.t sensor frame
 $\alpha \rightarrow$ tunable parameter
 $\sigma_r \rightarrow$ error in FOV range from sensor specification sheet
 $\sigma_\theta \rightarrow$ error in FOV azimuth from sensor specification sheet

Let $z_i = (x_i, y_i)^T$

$$\theta_i = \tan^{-1} \frac{y_i}{x_i}$$

$$R_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & (\alpha\sigma_\theta)^2 \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}^T$$

measurement noise covariance of constant area in cartesian coordinate for a radar measurement in the sensor frame

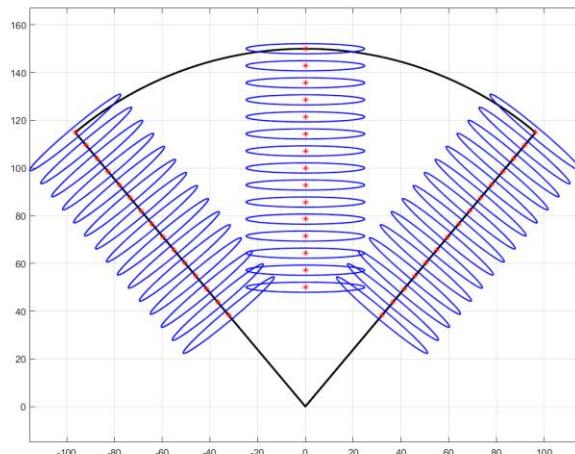


Figure : Variation of Error Ellipse for different locations within FOV after approximation

6.5 Measurement Noise Covariance for Camera

For Camera the measurement noise is reported as variance in pixel lateral and longitudinal position $\sigma_{\Delta x}$ and $\sigma_{\Delta y}$. Assuming that the errors are uncorrelated, the measurement noise covariance can be represented in the image pixel coordinate as follows.

$$R_{r,\theta} = \begin{pmatrix} \sigma_{\Delta x}^2 & 0 \\ 0 & \sigma_{\Delta y}^2 \end{pmatrix}$$

Since in this project the camera measurements are in cartesian coordinate and assuming the depth information is known, the camera measurement noise covariance is represented in cartesian coordinate space.

Due to perspective transformation in imaging geometry the errors in the cartesian frame is much higher in the longitudinal range than in the lateral range. The error covariance in cartesian coordinate is computed by modelling the errors in range and azimuth as follows:

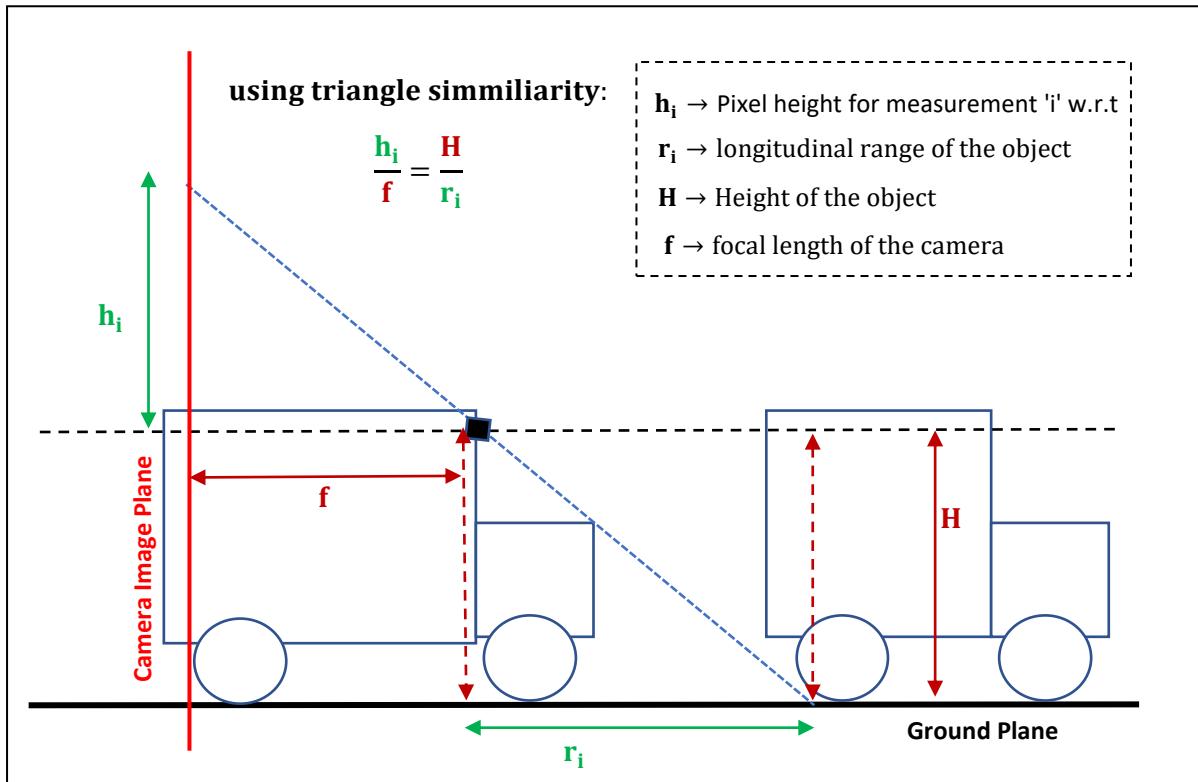


Figure : Projective Geometry for determining relation between pixel height and longitudinal distance

$$r_i = \frac{fH}{h_i} \rightarrow \text{relation between pixel height and longitudinal distance}$$

Determination of error in range from error in pixel height

$\sigma_{\Delta x}$ → error in pixel height (constant, determined from sensor specification sheet)

$\sigma_{\Delta y}$ → error in pixel width (constant, determined from sensor specification sheet)

σ_r → error in range

σ_θ → error in constant

$$r = \frac{k}{h} \quad \left. \right\} \text{range is inversely proportional to pixel height (equation 1)}$$

$$\frac{dr}{dh} = -\frac{k}{h^2}$$

$$\sigma_r^2 = \left(-\frac{k}{h^2} \right)^2 \sigma_{\Delta x}^2 \quad \left. \right\} \text{variance in range from variance in pixel height (equation 2)}$$

$$\Rightarrow \sigma_r^2 = \left(\frac{k}{h^2} \right)^2 \sigma_{\Delta x}^2$$

$$\Rightarrow \sigma_r^2 = \left(\frac{k}{(\frac{k}{r})^2} \right)^2 \sigma_{\Delta x}^2 \quad \left. \right\} \text{variance in range as a function of range (equation 3)}$$

$$\Rightarrow \sigma_r^2 = \left(\frac{r^2}{k} \right)^2 \sigma_{\Delta x}^2$$

$$\Rightarrow \sigma_r^2 = r^4 \left(\frac{\sigma_{\Delta x}}{k} \right)^2$$

Enforcing error limits in range

The error σ_r^2 is restricted by the constraint : $\sigma_{rMin}^2 \leq \sigma_r^2 \leq \sigma_{rMax}^2$

$$\sigma_r^2 = Cr^4 \quad \left. \right\} \text{Equation 3 can be further simplified (equation 4)}$$

The proportionality constant is determined as :

$$\sigma_{rMax}^2 = R_{max}^4 C \quad \left. \right\} \text{(equation 5)}$$

$$\Rightarrow C = \frac{\sigma_{rMax}^2}{R_{max}^4}$$

$$\sigma_r^2 = \left(\frac{\sigma_{rMax}^2}{R_{max}^4} \right) r^4 \quad \left. \right\} \text{from Equation 5 (equation 6)}$$

Bounded error in range

$$\sigma_r^2 = \min \left((\sigma_{rMax}^2), \max \left((\sigma_{rMin}^2), \left(\left(\frac{\sigma_{rMax}^2}{R_{max}^4} \right) r^4 \right) \right) \right)$$

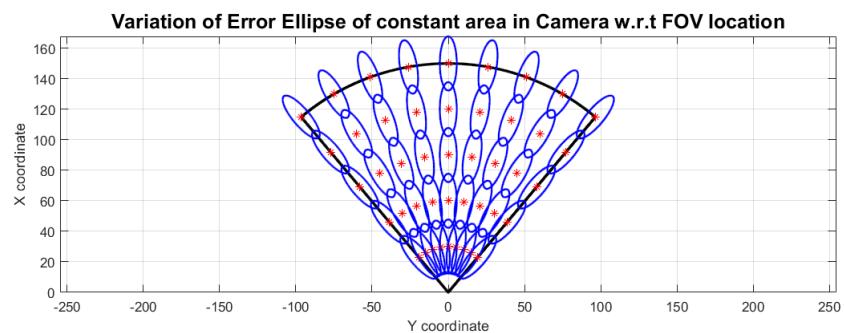
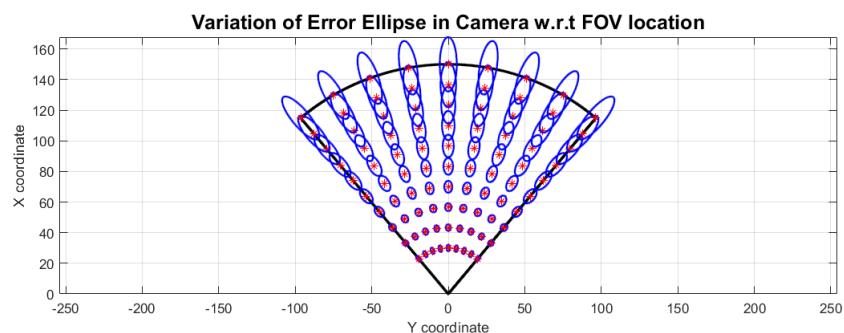
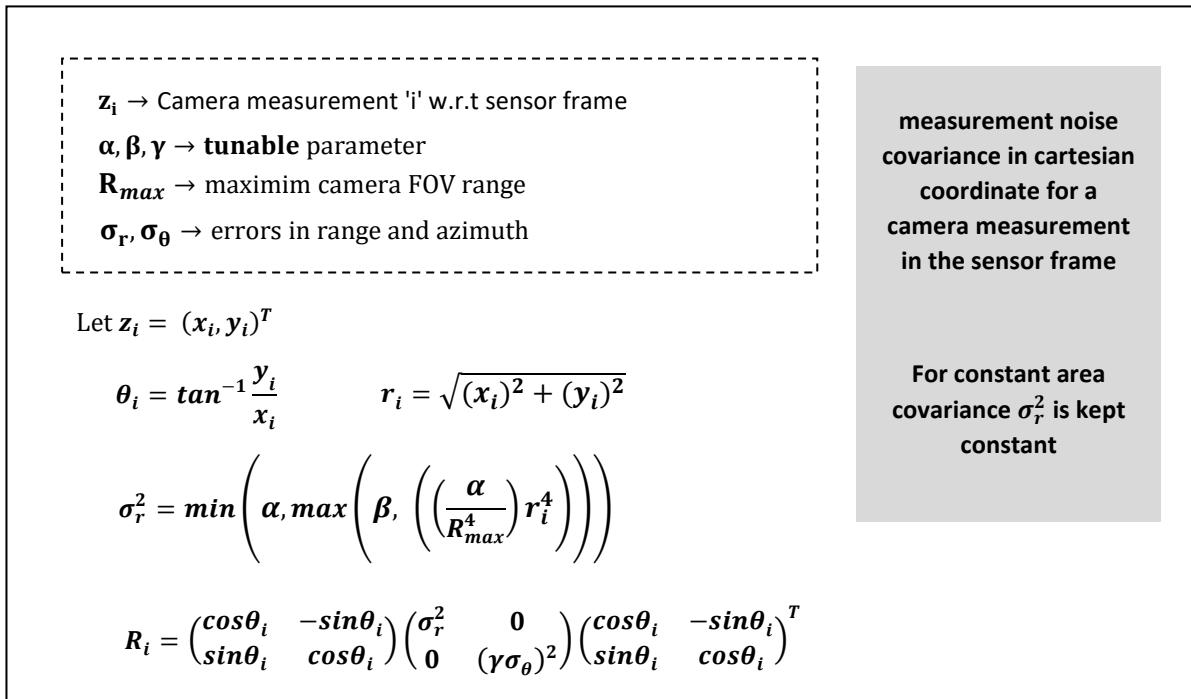


Figure : Measurement Noise Covariance at various Camera FOV locations

7. SENSOR FOV GRID

7.1 Introduction

Sensor Field of View Grid is a discretized representation of the operational environment around the ego vehicle. The Grid essentially characterizes the perceptual environment in terms of various sensor parameters like accuracy, resolution , Field of View boundary Limits , different sensor Field of View Overlaps etc. The Current scope of the Grid in this project is limited to Track Management. To enable support for other AD/ADAS features in subsequent versions like Free Space Estimation, Grid Fusion, Localization, Road Model Fusion, Situation Assessment and Path Planning, a rectangular shaped full grid is designed.

A Grid is needed for determining why a track got mis-detected with respect to environment parameters (miss detection might happen for reasons like : object out of the sensor FoV, object in occlusion, object in blind zone etc.). The Target Tracks might be required to be maintained in different prediction modes depending on its position and the scene of the environment. Similarly if a new object appears within the FoV, depending on the number sensor overlaps we shall have varying indications about its presence, since simultaneous detections by multiple sensors provides more evidence.

In this project Sensor Grid is used to compute Region Confidence and different sensor FOV zones with respect to vehicle frame efficiently which is utilized by track management for track initialization and track deletion. The grid is a pre cached as a look-up table which holds the position of a grid cell in the form of cell id and the corresponding region confidence.

Region confidence is derived from the sensor specifications and sensor overlaps.

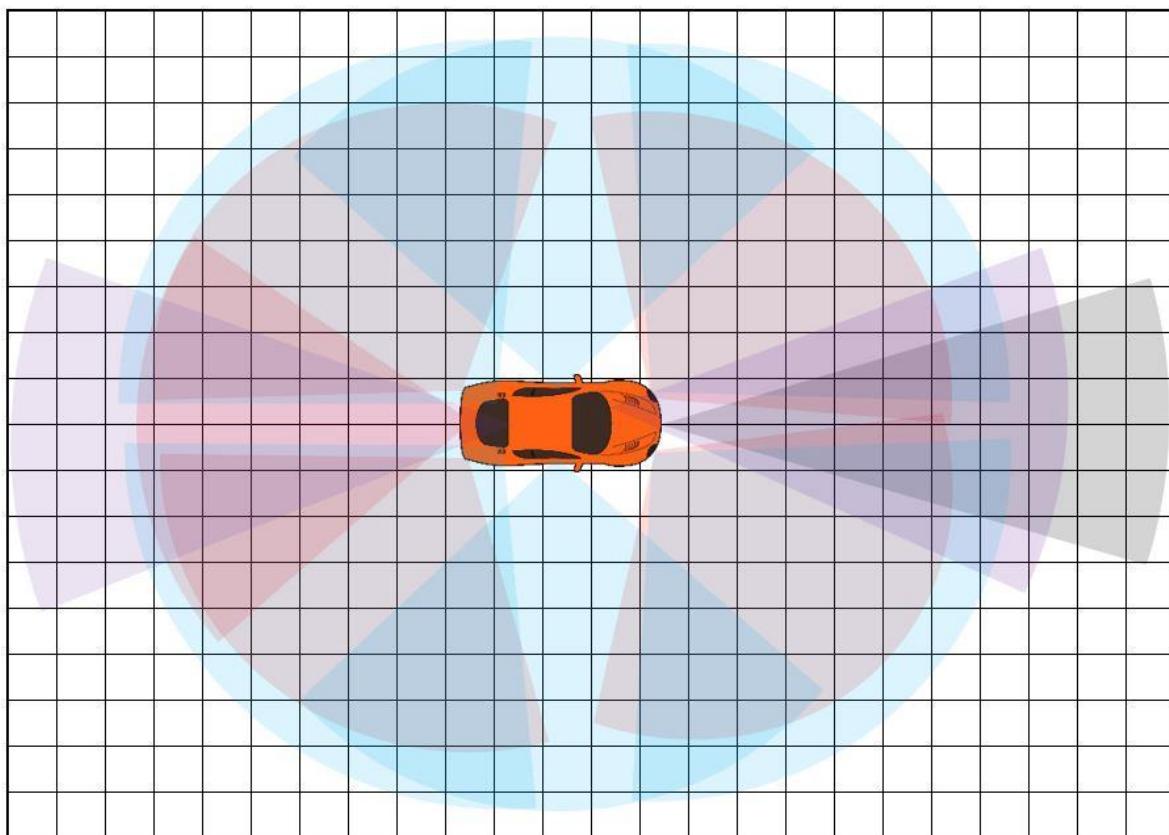


Figure : Visual depiction of a grid

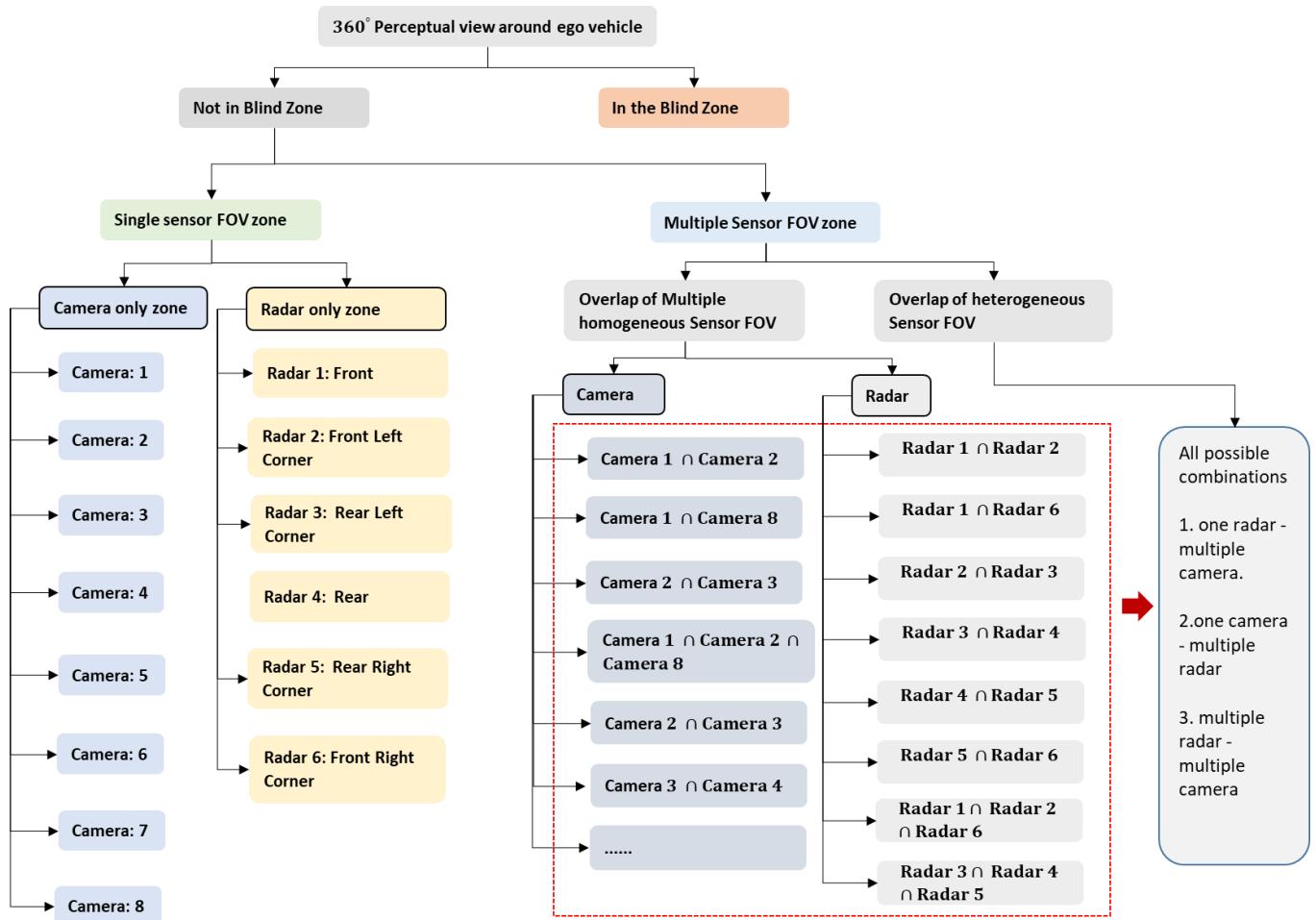


Figure : Different categories of sensor overlapped zones

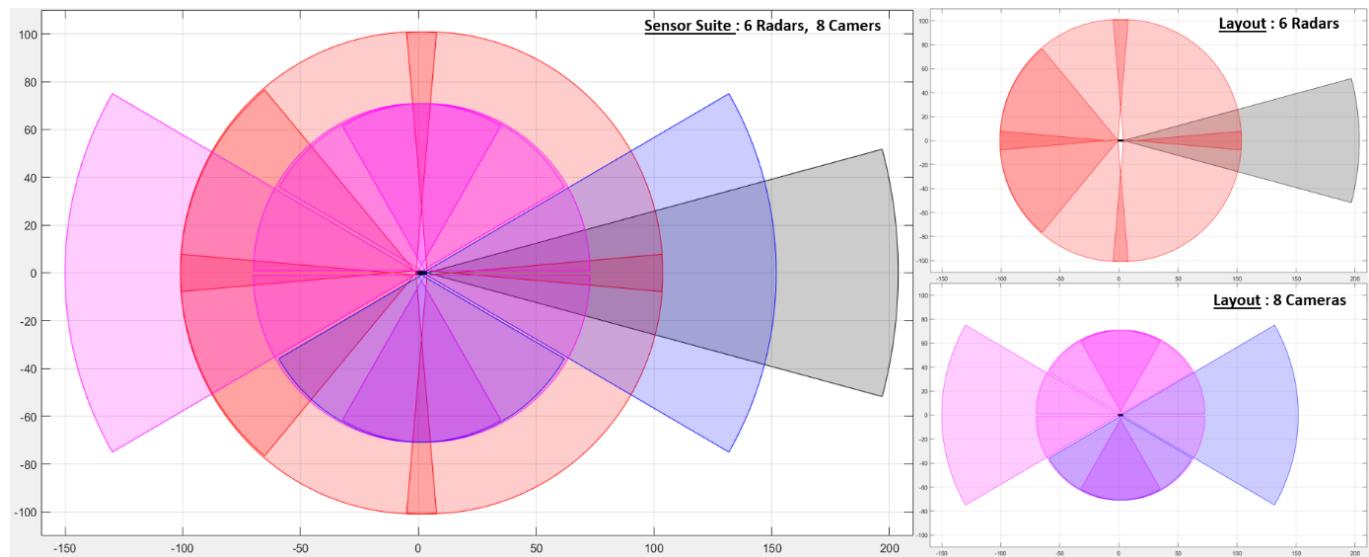


Figure : Different Sensor Layouts : Multiple Radars, Multiple Cameras, Multiple Radars and Cameras

7.2 Grid Concepts

The following section gives a conceptual description on the design of a grid. The concept is presented as an abstract problem of representing a bounded region in the form of a collection of discrete cells. Following which the Grid cell coordinates are derived with respect to the ego vehicle frame. The 2-dimensional Cell coordinates are then transformed to a 1-dimensional Cell index . The Cell index are then transformed from Ego Vehicle frame to each of the installed Local Sensor frame and determined if it falls within a sensor FoV. Finally the region confidence and Sensor FoV overlaps are computed and precached in a lookup Table.

The below topics are discussed/summarized in this section at a concept level.

- Input Parameter Identification for Grid Definition.
- Grid Cell Index Computation from Cartesian Grid Cell Coordinates.
- Conversion of Grid Cell Index to Cartesian Grid Cell Center.
- Coordinate Transformation of Grid Cell Centers from vehicle frame to sensor frame.
- Region Confidence Computation.

7.2.1 Grid Definition

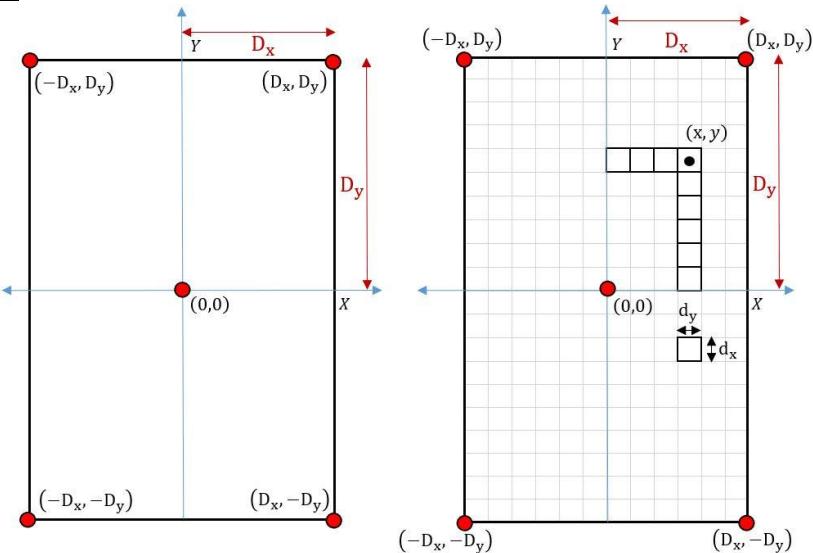


Figure : Grid Limits and Grid Resolution

Let An arbitrary point within the Grid: (x, y)

$$\begin{aligned} \text{Let a region be bounded as : } & -D_x \leq x \leq D_x \dots (1) \\ & -D_y \leq y \leq D_y \dots (2) \end{aligned}$$

$$g_x = \left\lfloor \frac{x}{d_x} \right\rfloor \dots (3)$$

Grid Cell coordinates :

$$g_y = \left\lfloor \frac{y}{d_y} \right\rfloor \dots (4)$$

$d_y \rightarrow$ Grid Cell lateral Dimension

$d_x \rightarrow$ Grid Cell longitudinal Dimension

7.2.2 Grid Cell Index Computation from a point

Let a region be bounded as follows :-

$$\begin{aligned} \text{Longitudinal Grid Limit: } & -D_x \leq x \leq D_x \\ \text{Lateral Grid Limit: } & -D_y \leq y \leq D_y \end{aligned}$$

Grid/bin coordinates

$$d_x \rightarrow \text{Grid Cell longitudinal Dimension}$$

$$d_y \rightarrow \text{Grid Cell lateral Dimension}$$

$$g_x = \left\lfloor \frac{X}{d_x} \right\rfloor$$

$$g_y = \left\lfloor \frac{Y}{d_y} \right\rfloor$$

Transform the Grid Cell coordinates such that it is positive:-

$$-D_x \leq X \leq D_x$$

$$\Rightarrow -\frac{D_x}{d_x} \leq \frac{X}{d_x} \leq \frac{D_x}{d_x}$$

$$\Rightarrow \left\lfloor -\frac{D_x}{d_x} \right\rfloor \leq \left\lfloor \frac{X}{d_x} \right\rfloor \leq \left\lfloor \frac{D_x}{d_x} \right\rfloor$$

$$\Rightarrow \left\lfloor -\frac{D_x}{d_x} \right\rfloor \leq g_x \leq \left\lfloor \frac{D_x}{d_x} \right\rfloor$$

$$\text{let, } k_x = \left\lfloor -\frac{D_x}{d_x} \right\rfloor$$

$$g'_x = g_x - k_x$$

$$\Rightarrow g'_x = \left\lfloor \frac{X}{d_x} \right\rfloor - \left\lfloor -\frac{D_x}{d_x} \right\rfloor$$

$$-D_y \leq Y \leq D_y$$

$$\Rightarrow -\frac{D_y}{d_y} \leq \frac{Y}{d_y} \leq \frac{D_y}{d_y}$$

$$\Rightarrow \left\lfloor -\frac{D_y}{d_y} \right\rfloor \leq \left\lfloor \frac{Y}{d_y} \right\rfloor \leq \left\lfloor \frac{D_y}{d_y} \right\rfloor$$

$$\Rightarrow \left\lfloor -\frac{D_y}{d_y} \right\rfloor \leq g_y \leq \left\lfloor \frac{D_y}{d_y} \right\rfloor$$

$$\text{let, } k_y = \left\lfloor -\frac{D_y}{d_y} \right\rfloor$$

$$g'_y = g_y - k_y$$

$$\Rightarrow g'_y = \left\lfloor \frac{Y}{d_y} \right\rfloor - \left\lfloor -\frac{D_y}{d_y} \right\rfloor$$

Maximum Grid Size:-

Since maximum longitudinal Grid Limit $x_{\max} = D_x$

$$\text{maximum value of longitudinal cells : } g'_{x_{\max}} = \left\lfloor \frac{D_x}{d_x} \right\rfloor - \left\lfloor -\frac{D_x}{d_x} \right\rfloor$$

So, maximum number of lateral cells : $C_x = g'_{x_{\max}} + 1$

$$\Rightarrow C_x = \left\lfloor \frac{D_x}{d_x} \right\rfloor - \left\lfloor -\frac{D_x}{d_x} \right\rfloor + 1$$

$$\text{maximum number of lateral cells : } C_y = \left\lfloor \frac{D_y}{d_y} \right\rfloor - \left\lfloor -\frac{D_y}{d_y} \right\rfloor + 1$$

Total number of cells : $C_n = C_x * C_y$

Generate 1D unique Grid cell ID from 2D Grid coordinates :-

$$\text{index} = C_x * g'_y + g'_x$$

$$\text{index} = \left(\left\lfloor \frac{D_x}{d_x} \right\rfloor - \left\lfloor -\frac{D_x}{d_x} \right\rfloor + 1 \right) * \left(\left\lfloor \frac{Y}{d_y} \right\rfloor - \left\lfloor -\frac{D_y}{d_y} \right\rfloor \right) + \left(\left\lfloor \frac{X}{d_x} \right\rfloor - \left\lfloor -\frac{D_x}{d_x} \right\rfloor \right)$$

7.2.3 Grid Centre Computation from Grid Cell Index

Inverse transformation from Grid Cell index to Cell Coordinates: –

INPUTS : Grid Cell Index : index
 Grid Parameters : $d_x \rightarrow$ Grid Cell longitudinal Dimension
 $d_y \rightarrow$ Grid Cell lateral Dimension
 $D_x \rightarrow$ Grid longitudinal Upper Limit
 $D_y \rightarrow$ Grid lateral Upper Limit
 $C_x \rightarrow$ Maximum number of longitudinal grid cells
 $C_y \rightarrow$ Maximum number of lateral grid cells

Grid Cell Coordinate Extraction: –

Since we derived previously that : $\text{index} = C_x * g'_y + g'_x$

$$\text{we can extract Lateral Grid Cell Coordinate as : } g'_y = \left\lfloor \frac{\text{index}}{C_x} \right\rfloor$$

Similarly we can extract Longitudinal Grid Cell Coordinate as :

$$g'_x = \text{index} - C_x g'_y$$

$$\Rightarrow g'_x = \text{index} - C_x \left\lfloor \frac{\text{index}}{C_x} \right\rfloor$$

Grid Cell Cartesian Coordinate Extraction: –

$$\text{Since we derived previously that : } g'_x = \left\lfloor \frac{x}{d_x} \right\rfloor - \left\lfloor -\frac{D_x}{d_x} \right\rfloor \text{ & } g'_y = \left\lfloor \frac{y}{d_y} \right\rfloor - \left\lfloor -\frac{D_y}{d_y} \right\rfloor$$

$$\Rightarrow \left\lfloor \frac{x}{d_x} \right\rfloor = g'_x + \left\lfloor -\frac{D_x}{d_x} \right\rfloor$$

$$\Rightarrow \left\lfloor \frac{x}{d_x} \right\rfloor = \text{index} - C_x \left\lfloor \frac{\text{index}}{C_x} \right\rfloor + \left\lfloor -\frac{D_x}{d_x} \right\rfloor$$

$$\Rightarrow x = d_x \left(\text{index} - C_x \left\lfloor \frac{\text{index}}{C_x} \right\rfloor + \left\lfloor -\frac{D_x}{d_x} \right\rfloor \right)$$

$$\Rightarrow \left\lfloor \frac{y}{d_y} \right\rfloor = g'_y + \left\lfloor -\frac{D_y}{d_y} \right\rfloor$$

$$\Rightarrow \left\lfloor \frac{y}{d_y} \right\rfloor = \left\lfloor \frac{\text{index}}{C_x} \right\rfloor + \left\lfloor -\frac{D_y}{d_y} \right\rfloor$$

$$\Rightarrow y = d_y \left(\left\lfloor \frac{\text{index}}{C_x} \right\rfloor + \left\lfloor -\frac{D_y}{d_y} \right\rfloor \right)$$

To compute the center of the Grid Cell, an offset of 0.5 is added to the Cell Coordinate

$$x = d_x \left(\text{index} - C_x \left\lfloor \frac{\text{index}}{C_x} \right\rfloor + \left\lfloor -\frac{D_x}{d_x} \right\rfloor + 0.5 \right)$$

$$y = d_y \left(\left\lfloor \frac{\text{index}}{C_x} \right\rfloor + \left\lfloor -\frac{D_y}{d_y} \right\rfloor + 0.5 \right)$$

7.2.4 Grid Cell Coordinates Transformation from Vehicle Frame to Sensor Frame

$$X_{SF} = R_\theta^T(X_{VF} - T)$$

$$\Rightarrow \begin{pmatrix} X_{SF} \\ Y_{SF} \end{pmatrix} = \begin{pmatrix} \cos(\theta_{mount}) & \sin(\theta_{mount}) \\ -\sin(\theta_{mount}) & \cos(\theta_{mount}) \end{pmatrix} \left(\begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} X_{mount} \\ Y_{mount} \end{pmatrix} \right)$$

7.2.5 Region Confidence Computation

The Region confidence of a sensor i is computed from the below function :

$$p(\theta; a, b) = \frac{1}{1 + e^{(a\theta+b)}}$$

$$x \rightarrow \text{longitudinal coordinate w.r.t sensor frame}$$

$$y \rightarrow \text{lateral coordinate w.r.t sensor frame}$$

$$p(r; c) = \frac{1}{e^{(cr^2)}}$$

$$r = \sqrt{x^2 + y^2}$$

$$a, b, c \rightarrow \text{tunable parameters for sensor } i$$

$$\theta = \tan^{-1} \frac{y}{x}$$

$$f(r, \theta; a, b, c) = p(\theta; a, b) + p(r; c)$$

$$\Rightarrow f(r, \theta; a, b, c) = \frac{1}{1+e^{(a\theta+b)}} + \frac{1}{e^{(cr^2)}}$$

Illustration

The below figures are the heat map plots of the above region confidence function plotted for Long Range Radar FoV and Short Range Radar FoV. The confidence decreases as the color changes from red to yellow to green. It can be observed that as the range increases the confidence decreases and as the azimuth increases the confidence also decreases. By using different values of parameters (a,b,c) corresponding to different sensors , confidence functions suited for specific sensor types can be created.

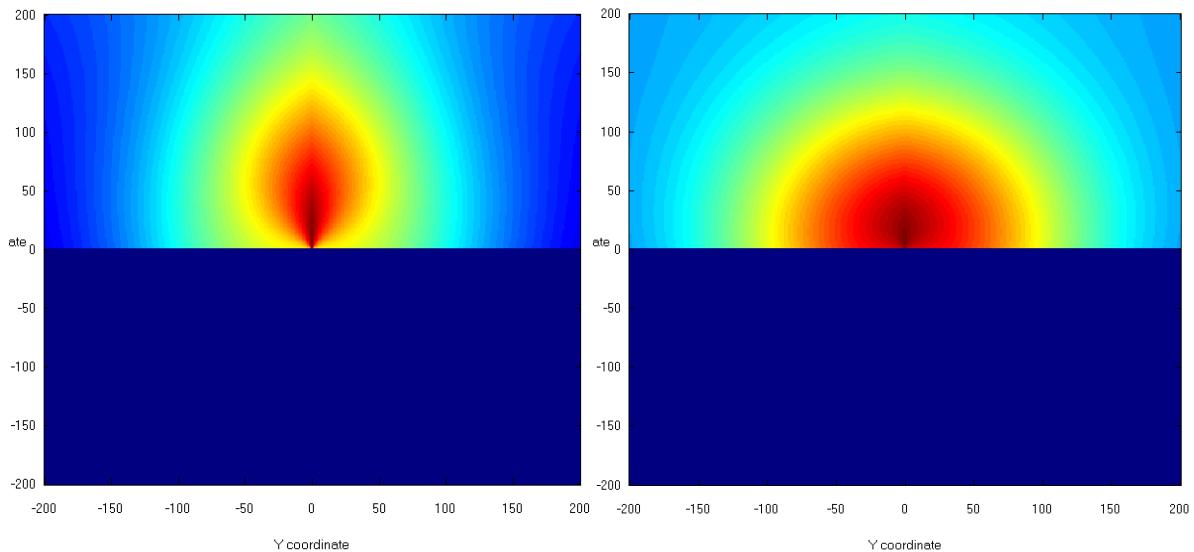
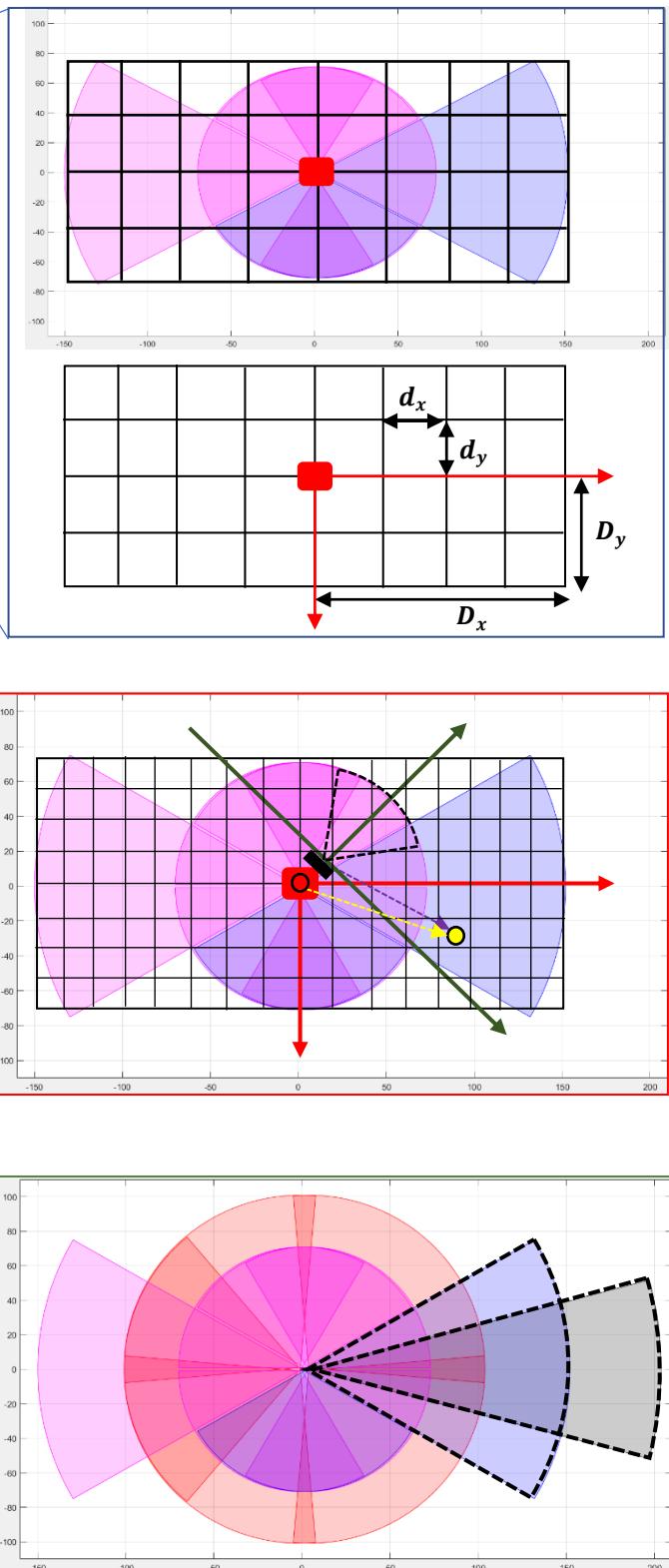
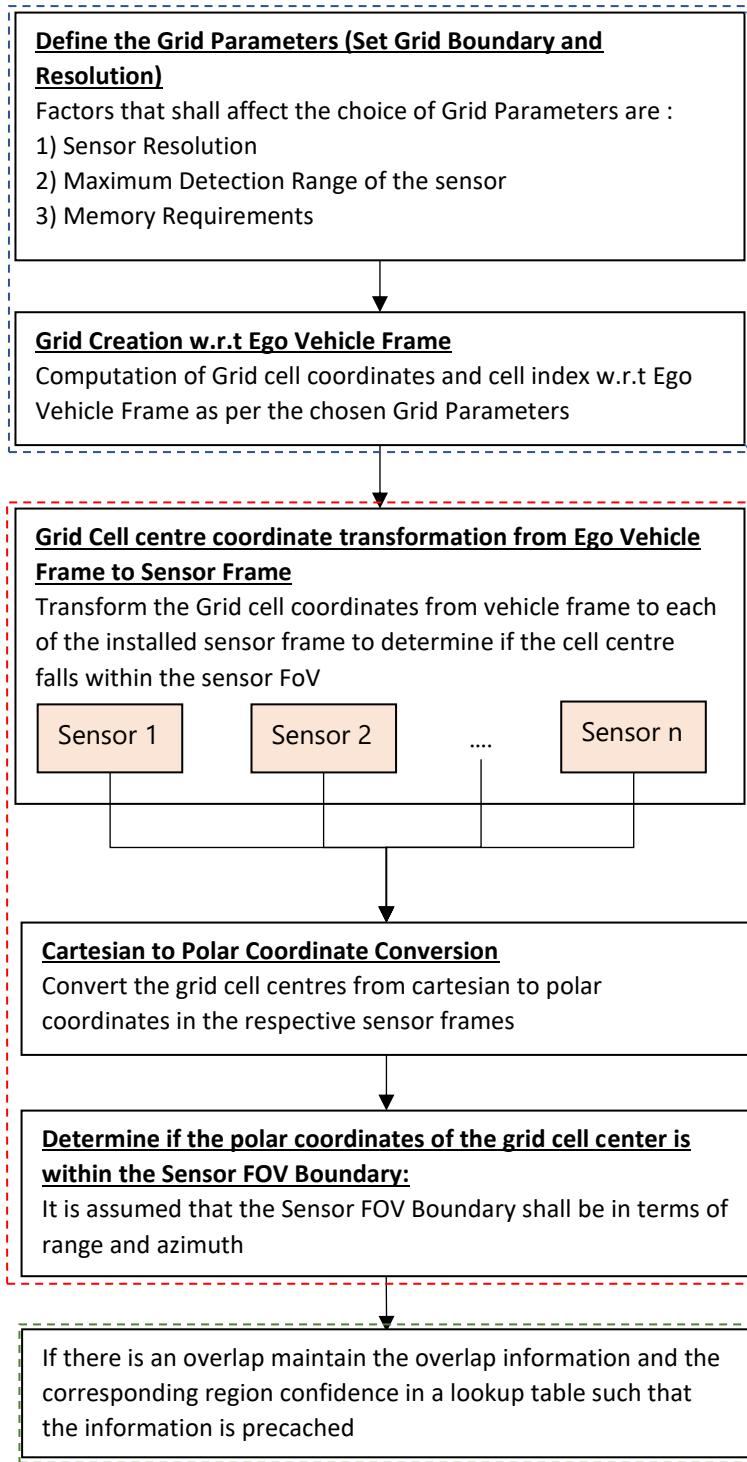


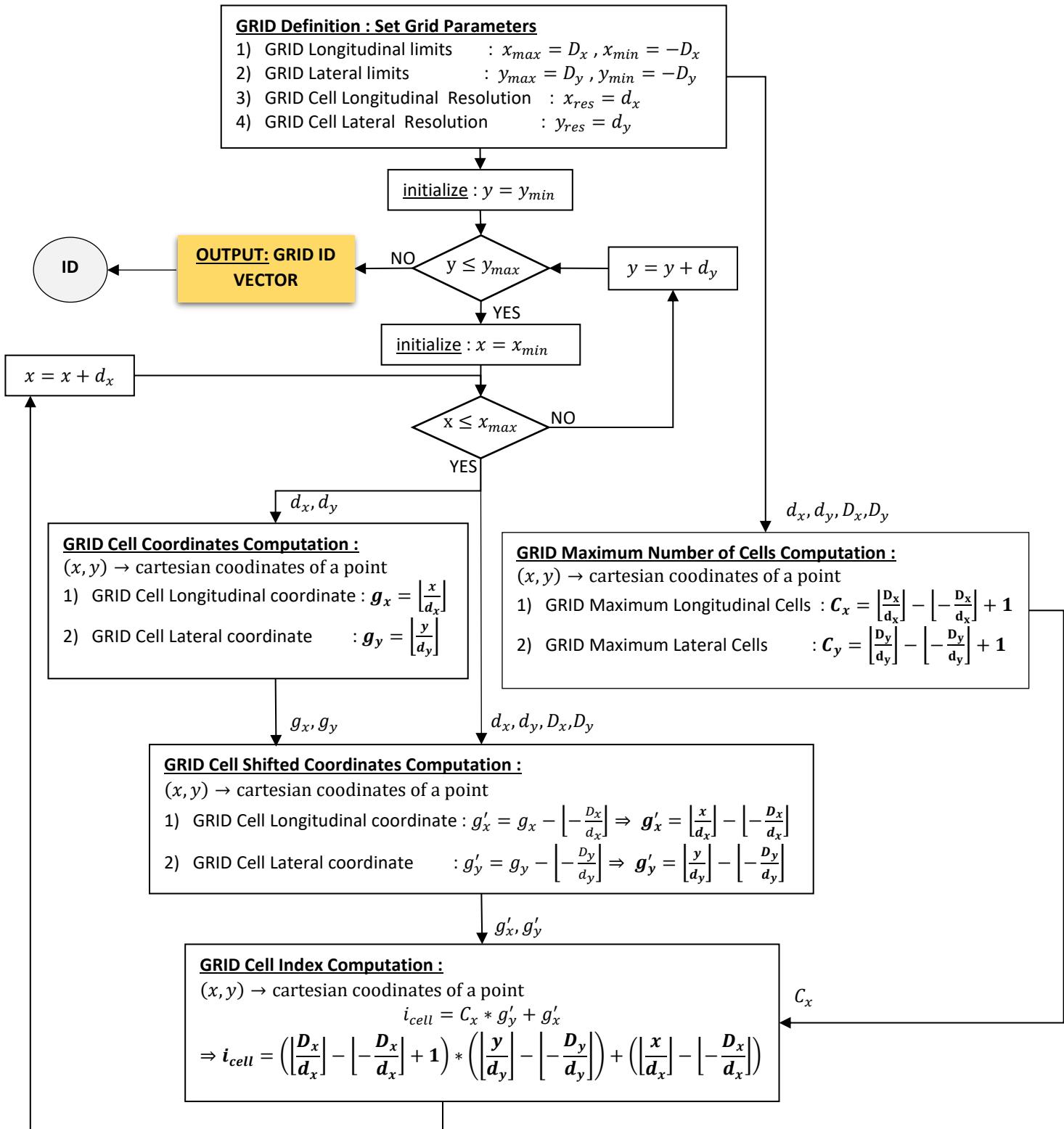
Figure : Variation in confidence in LRR and MRR as HeatMaps

7.3 High Level Design

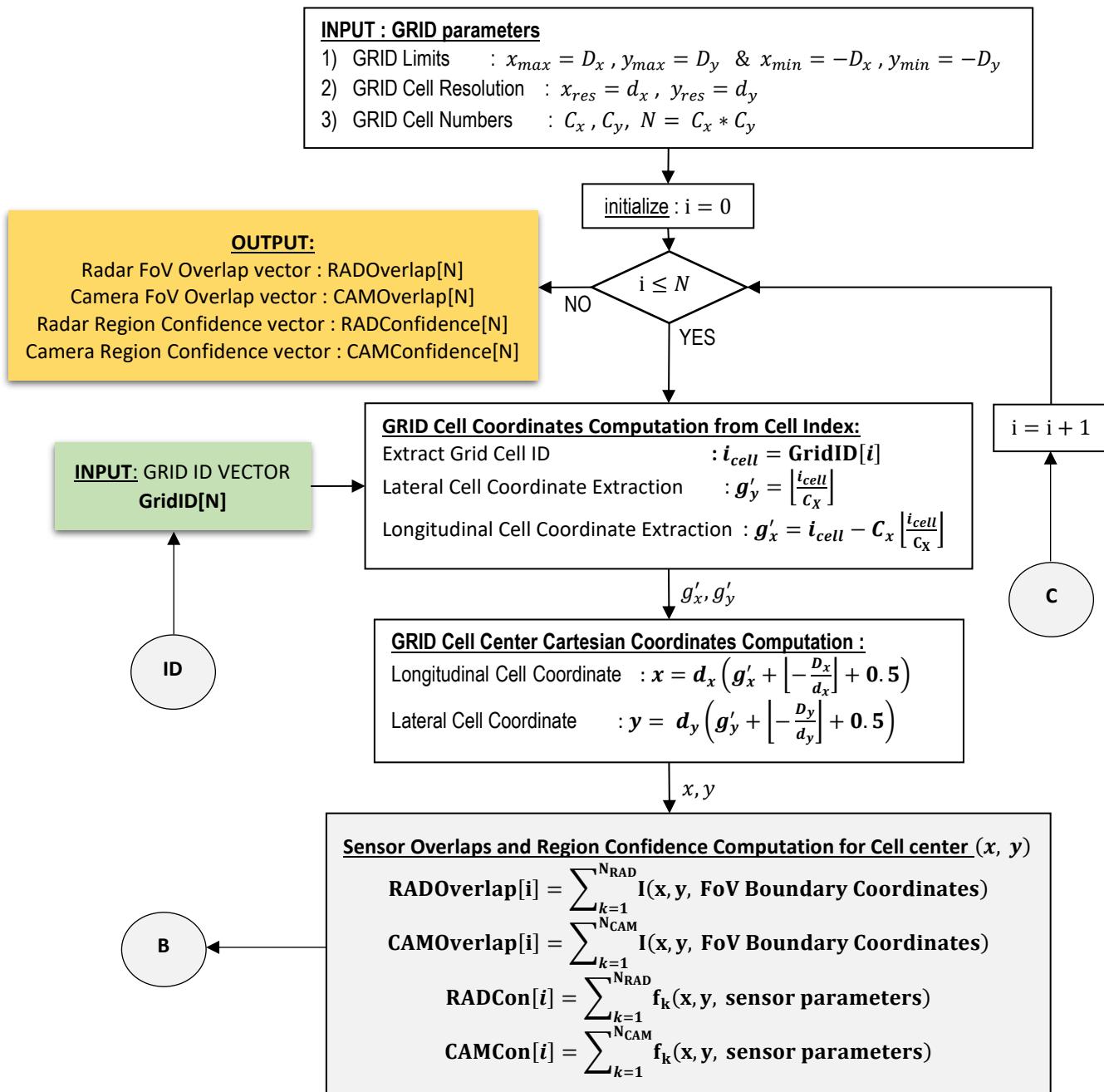


7.4 Detailed Design

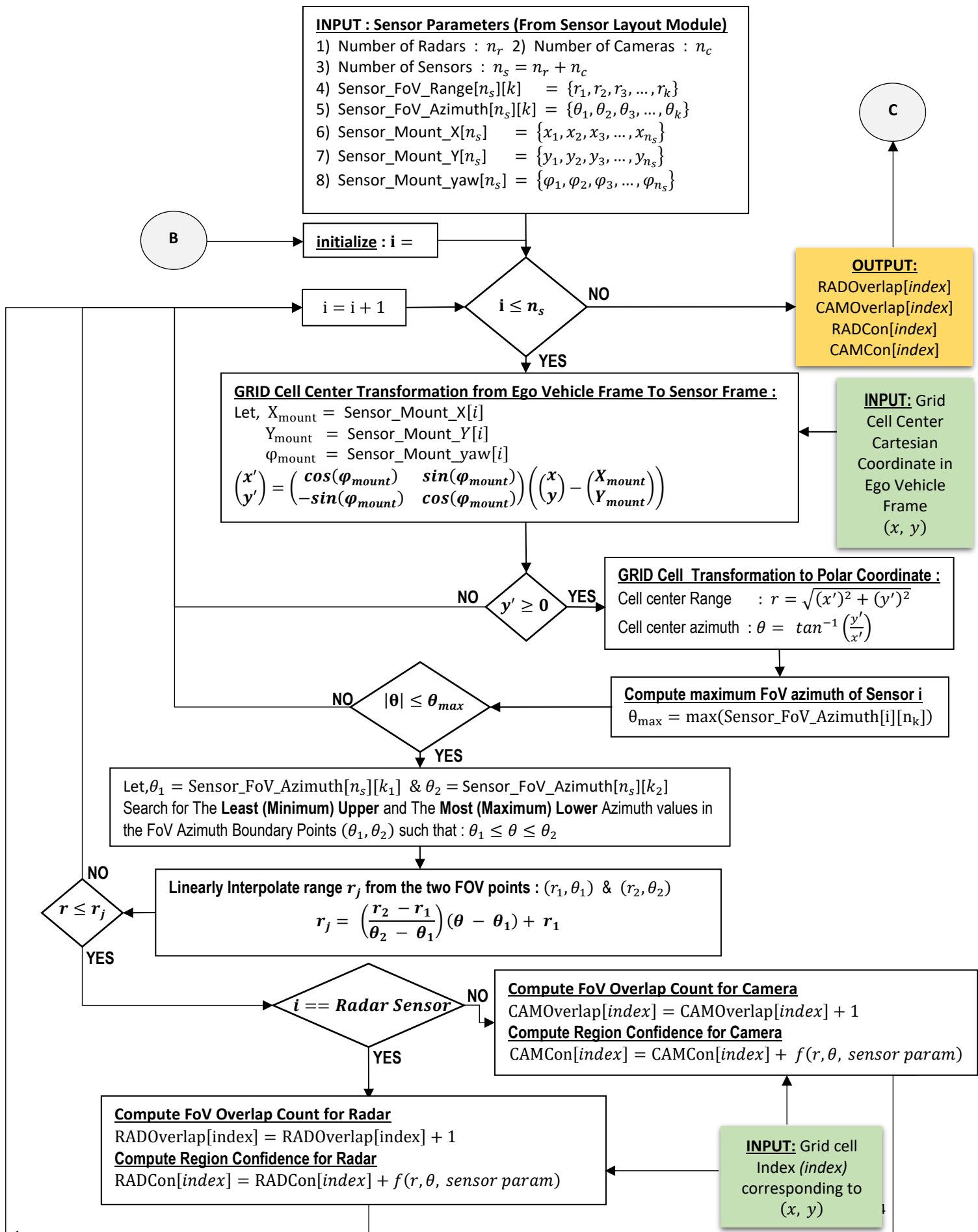
7.4.1 Flowchart of Grid Cell Index Computation from a point



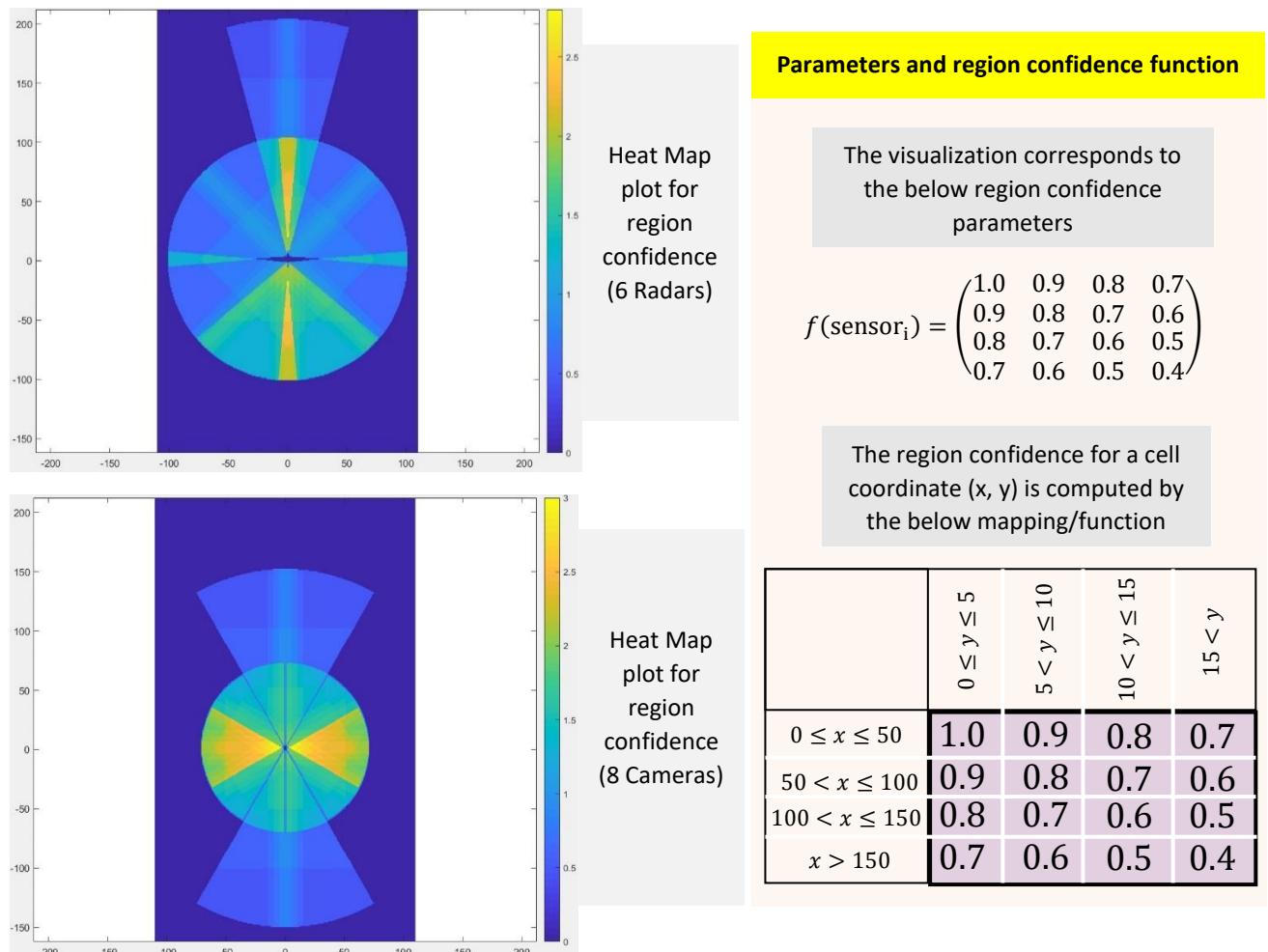
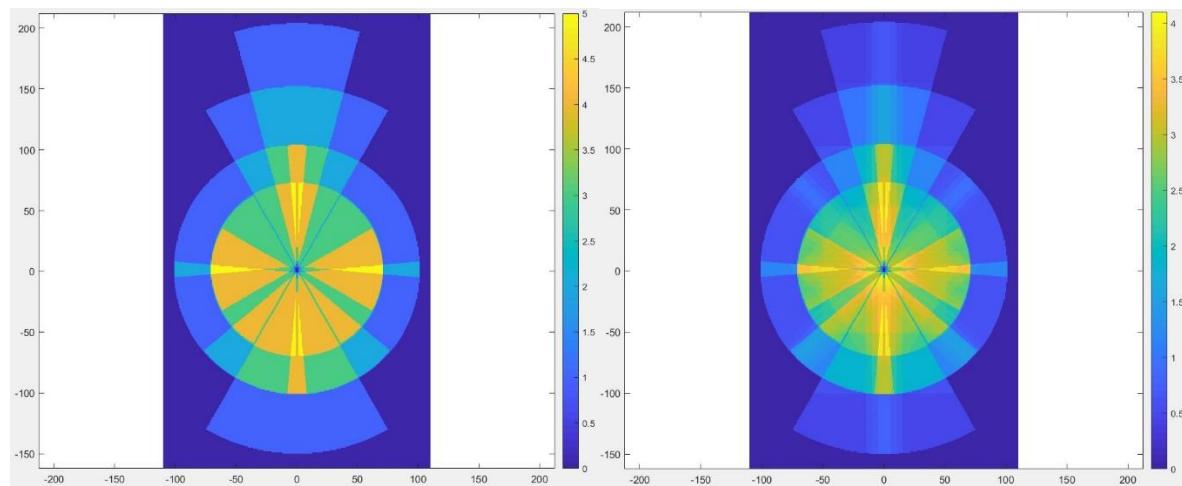
7.4.2 Flowchart of Sensor FoV Overlap and Region Confidence Computation - part 1



7.4.3 Flowchart of Sensor FoV Overlap and Region Confidence Computation - part 2



7.5 Visualization



8. SPATIAL ALIGNMENT

8.1 Introduction

The Measurements and the Noise covariance of the sensors shall be with respect to the respective local sensor frame. In order to perform data fusion , the measurement and the noise covariance needs to be transformed to a common frame of reference.

Also the measurement are received as an array of structures from the respected sensors. For efficient vectorized operation, the valid measurements after coordinate transformation are cached as an concatenated array of sensor measurements (all sensors).

Spatial Alignment module is responsible for the following functions :

- Coordinate Transformation of Sensor **Measurement Vector** from respective Sensor frames to Ego Vehicle Frame (Vehicle Wheel Base Center)
- Coordinate Transformation of Sensor **Measurement Noise Covariance** from respective Sensor frames to Ego Vehicle Frame
- Maintain a matrix of coordinate transformed **valid measurements** such that the matrix is a **concatenation of measurements from all sensors**
- Maintain a matrix of sensor IDs which are the sources of the respective measurements and measurement CAN index, such that various relevant sensor properties can be extracted and extra measurement parameters like 'snr', 'classification' etc. can also be extracted.

8.2 Coordinate Transformation Concepts

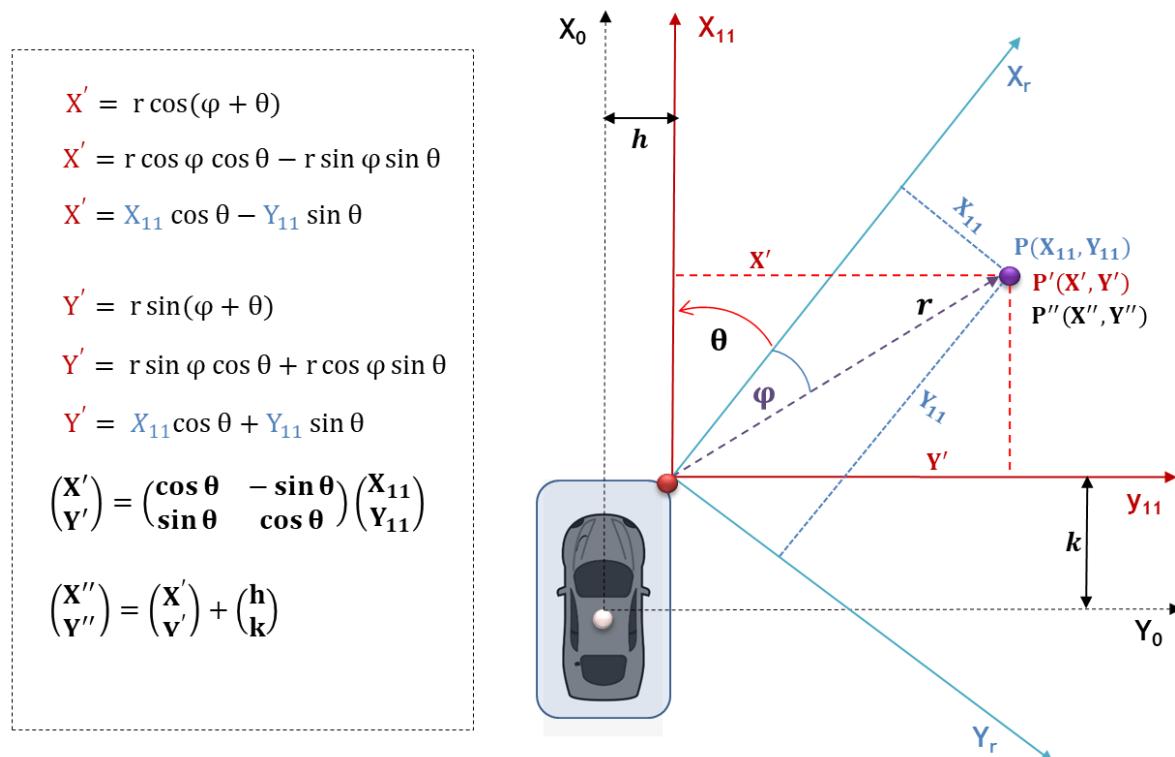


Figure : 2D-coordinate transformation from sensor frame to vehicle frame

INPUT : measurement vector in sensor frame: $(px, py, vx, vy)^T$

measurement Noise Covariance in sensor frame: $\begin{pmatrix} \sigma_{px}^2 & 0 & 0 & 0 \\ 0 & \sigma_{py}^2 & 0 & 0 \\ 0 & 0 & \sigma_{vx}^2 & 0 \\ 0 & 0 & 0 & \sigma_{vy}^2 \end{pmatrix}$

position measurement vector in vehicle frame:

$$\begin{pmatrix} px' \\ py' \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} px \\ py \end{pmatrix} + \begin{pmatrix} X_i \\ Y_i \end{pmatrix}$$

$\theta_i \rightarrow$ sensor mount azimuth angle

$X_i \rightarrow$ sensor mount x coordinate w.r.t WBC

$Y_i \rightarrow$ sensor mount y coordinate w.r.t WBC

velocity measurement vector in vehicle frame:

$$\begin{pmatrix} vx' \\ vy' \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} vx \\ vy \end{pmatrix}$$

Position measurement Noise Covariance in vehicle frame:

$$\Sigma_{pos} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \sigma_{px}^2 & 0 \\ 0 & \sigma_{py}^2 \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}^T$$

velocity measurement Noise Covariance in vehicle frame:

$$\Sigma_{vel} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \sigma_{vx}^2 & 0 \\ 0 & \sigma_{vy}^2 \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}^T$$

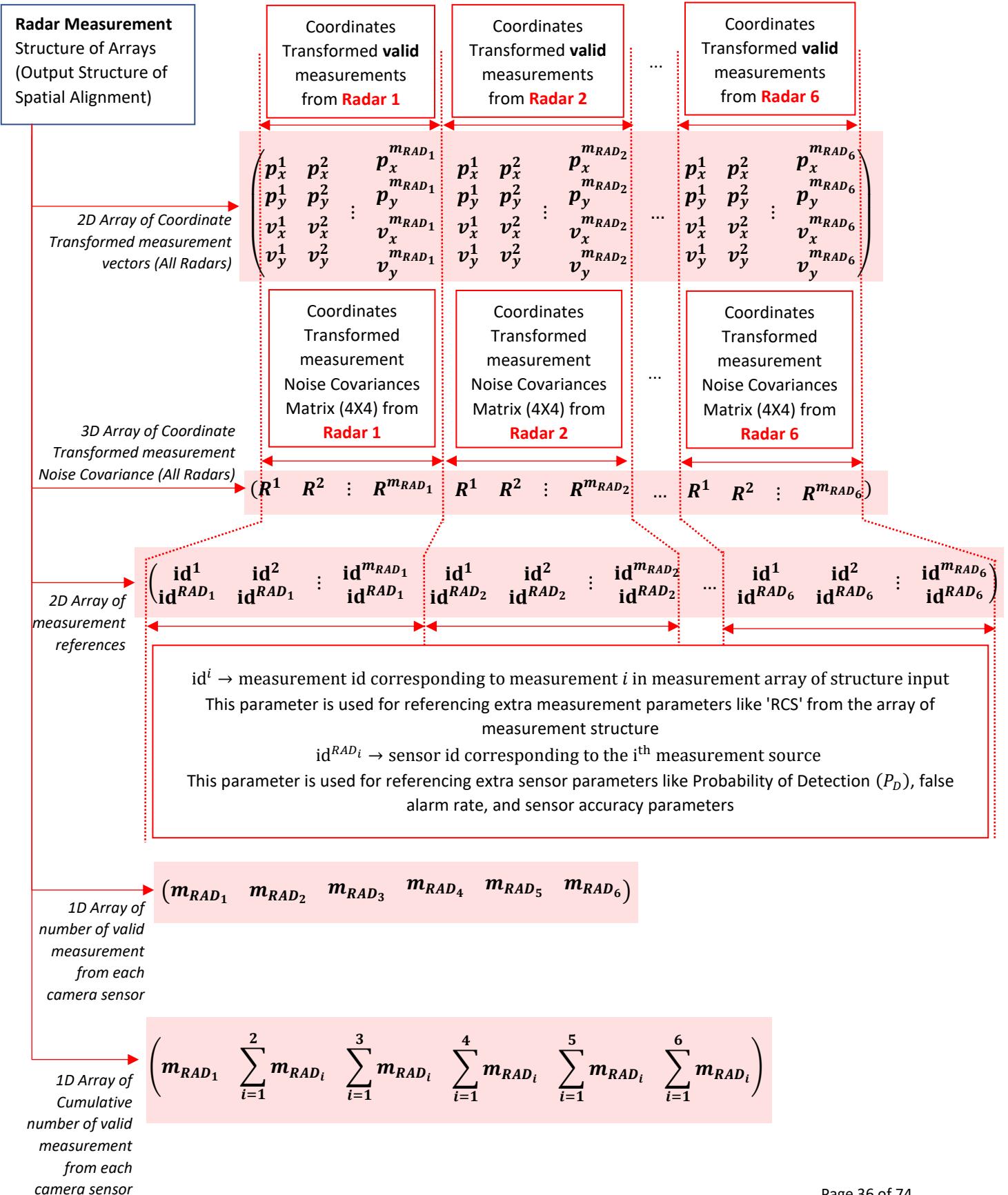
So, Measurement Noise Covariance in vehicle frame:

$$\begin{pmatrix} (\Sigma_{pos})_{2 \times 2} & (0)_{2 \times 2} \\ (0)_{2 \times 2} & (\Sigma_{vel})_{2 \times 2} \end{pmatrix}_{4 \times 4}$$

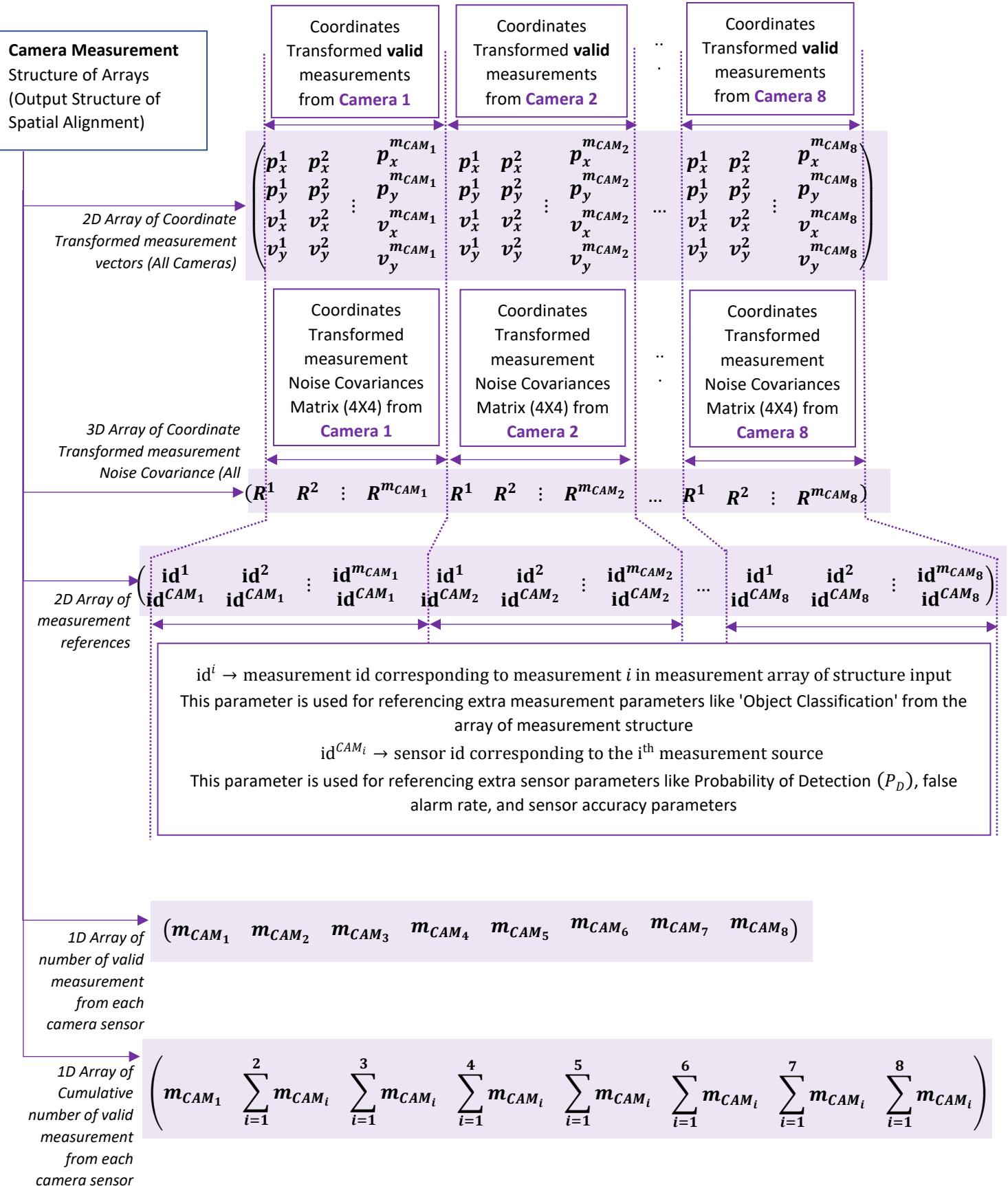
$$\Rightarrow \begin{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \sigma_{px}^2 & 0 \\ 0 & \sigma_{py}^2 \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}^T & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \sigma_{vx}^2 & 0 \\ 0 & \sigma_{vy}^2 \end{pmatrix} \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}^T \end{pmatrix}$$

8.3 Sensor Measurement Structure

8.3.1 Radar Measurement Matrix

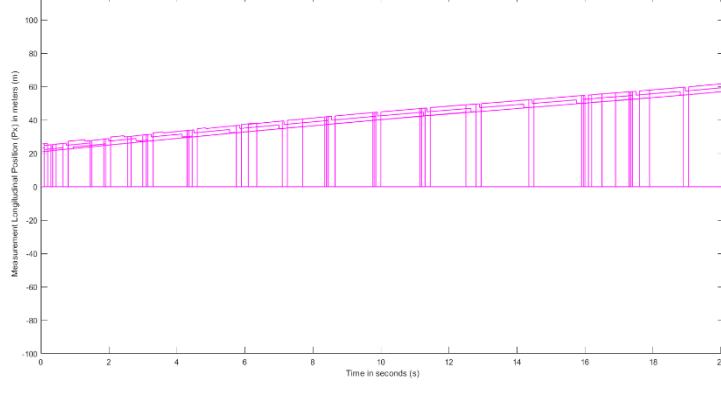
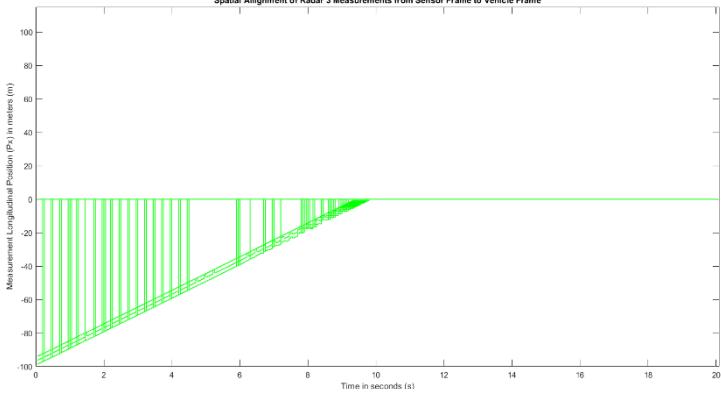
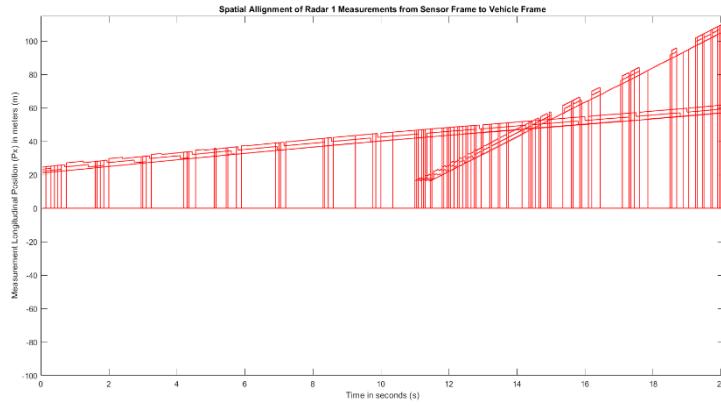
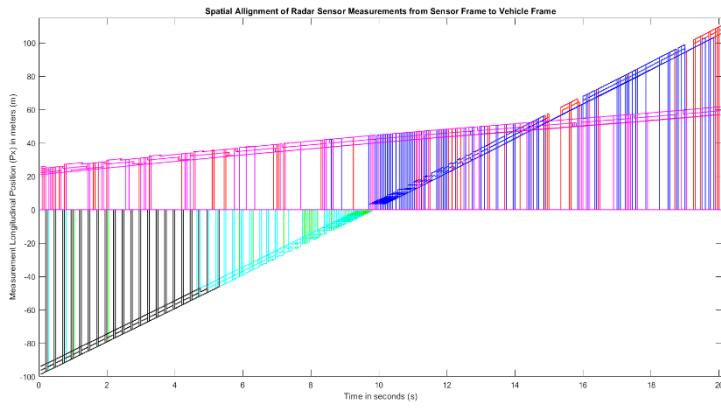


8.3.2 Camera Measurement Matrix



8.4 Plots and Analysis

The below plot is for Radar Longitudinal position. Similar plots can be generated and relevant observations can be made for lateral position and the Camera measurements



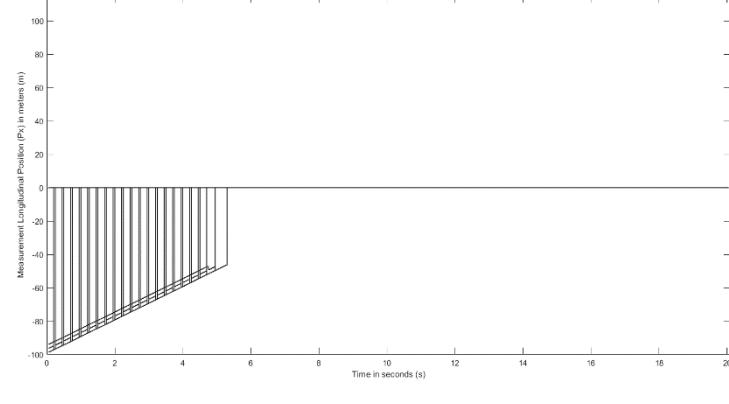
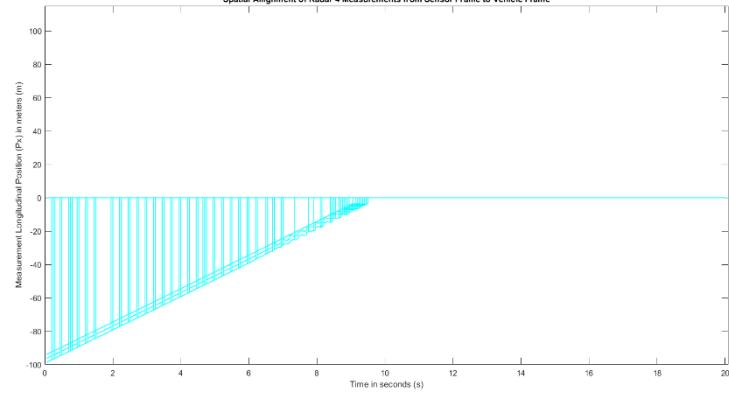
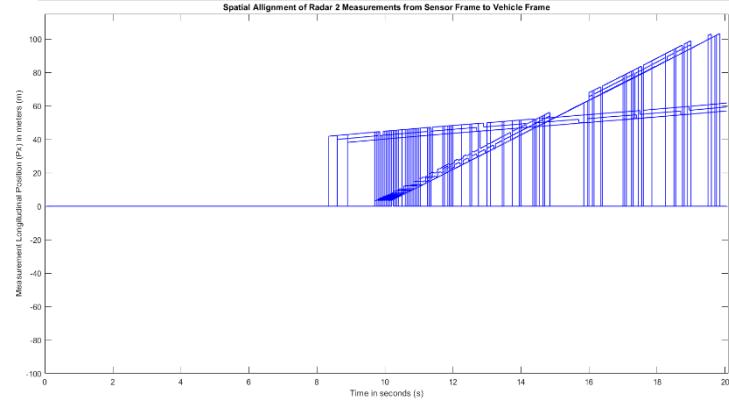
The following additional observations can be made can be made from the below plots

Depending on the Field Of View of the mounted Radars , mounting parameters and the kinematics of the Traffic Object. The sensors might detect the objects for a specific durations which varies from one sensor to the other

Variation of Longitudinal position of the radar measurements with respect to time

The following can be observed from the plots.

1. Two Traffic objects are present for entire the duration of the simulation (Two distance consistent motion patterns).
2. Multiple measurement are returned by the radar sensor for each of the objects.
3. The measurements are consistent after Spatial Alignment.
4. The longitudinal position of one object changes very less w.r.t time indicating the object could be moving almost with the same velocity as the ego vehicle.
5. The longitudinal position of another object changes rapidly w.r.t time. It comes from behind and moves at the front of the ego vehicle.
6. The variation in position is almost linear, indicating almost constant relative velocity of the Traffic Object



9. OBJECT FUSION

9.1 Introduction

Object fusion is the process of fusion of data from multiple sources. In the context of object tracking algorithm, the received data is in the form of measurements from different sensors. Each of the sensors has its own properties like sensor accuracy and resolution. Object fusion shall consider the sensor properties and undergo measurement fusion accordingly such that complementary sensor properties can be incorporated in the process of data fusion.

9.2 Object Fusion Concept & Design

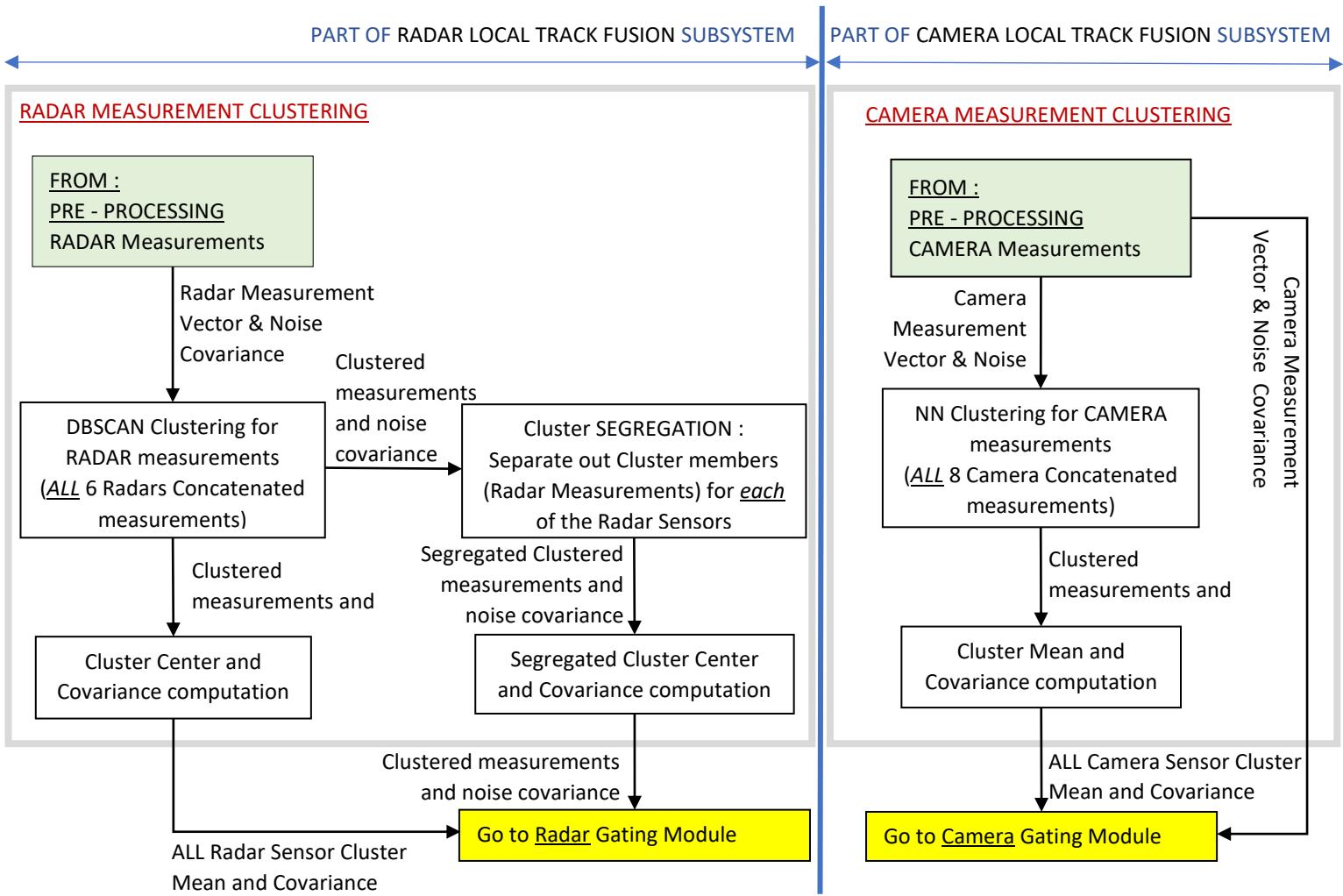
The following section gives a conceptual description of different stages of Object Fusion. Object Fusion is one of the core components of multi object tracking using multiple sensors. Object Fusion involves fusion of measurements from multiple sensors (same modality and different modality)

The object fusion module can be separated as following based on their functionality:

- Sensor Measurements Clustering : [DBSCAN for Radar and NN for Camera](#)
- State prediction : [CA Linear process model](#)
- Gating of Sensor Measurements to Predicted Tracks : [Mahalanobis Distance](#)
- Data association in each of the sensors : [PDAF](#)
- Homogeneous Sensor Fusion : [Combine track estimates from same sensor type modalities](#)
- Heterogeneous Sensor Fusion : [Track to Track Fusion](#)

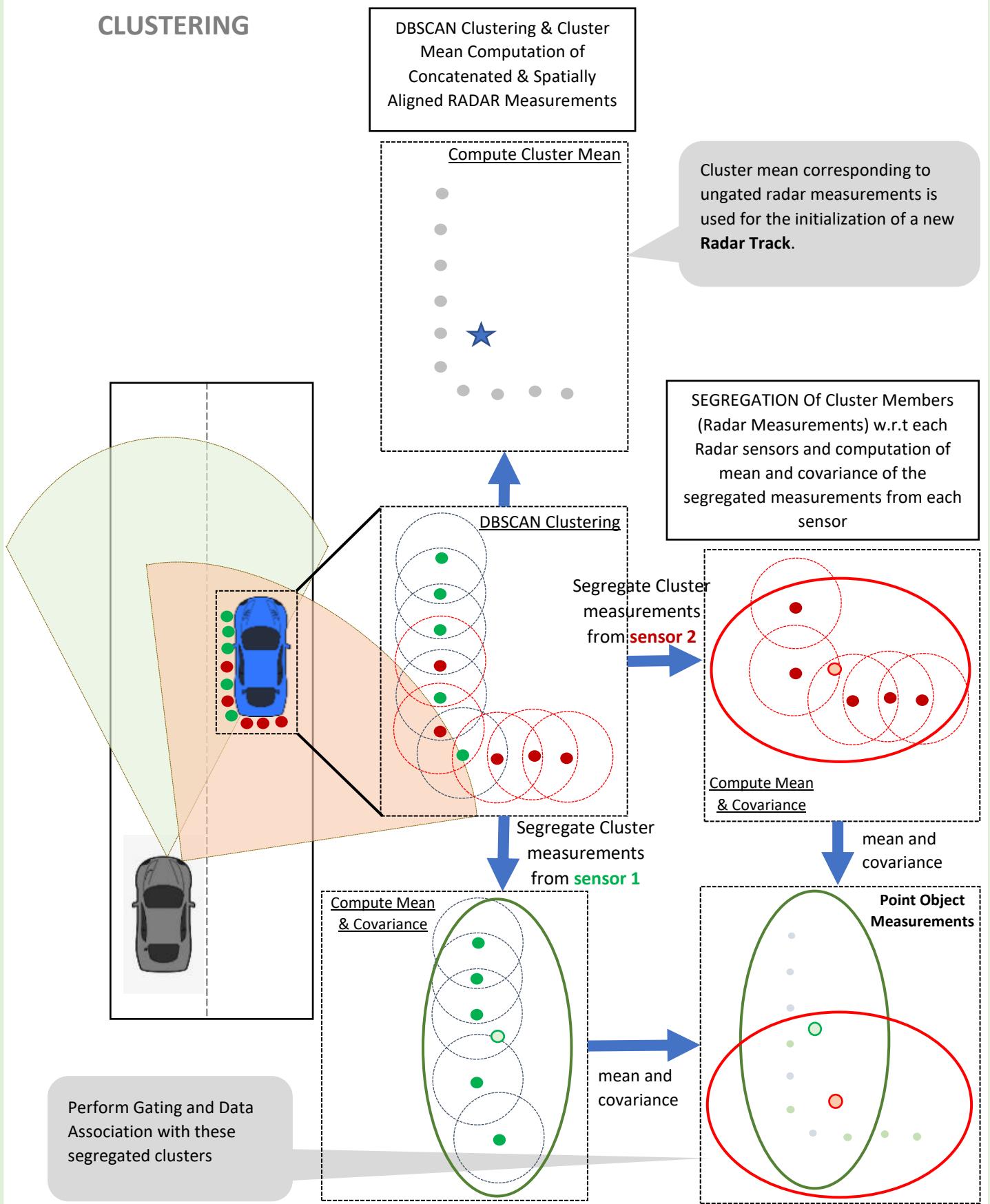
9.3 Sensor Measurement Clustering

9.3.1 Module Architecture/Execution Flow

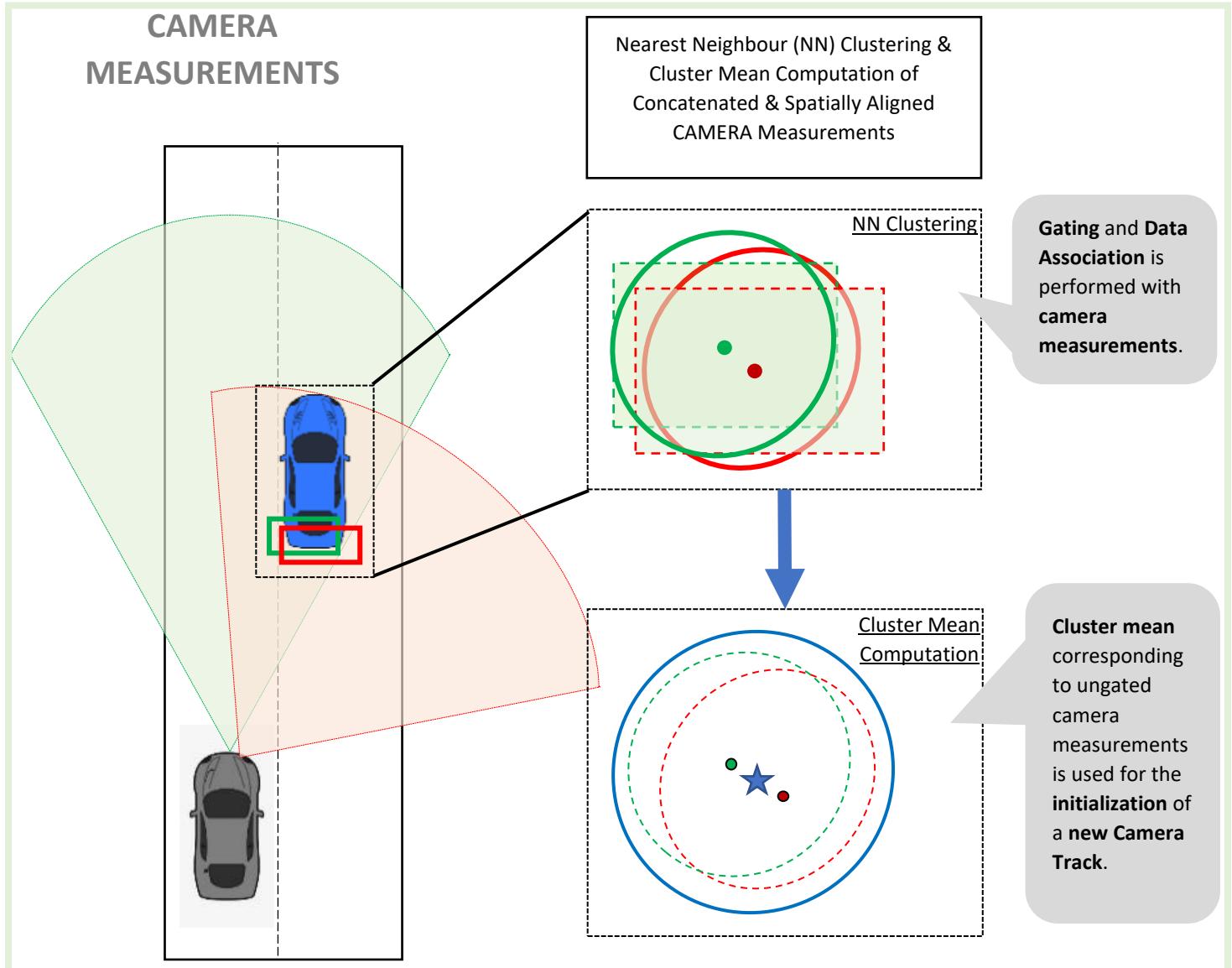


9.3.2 High - Level Design Of Radar Measurement Clustering

RADAR MEASUREMENTS CLUSTERING

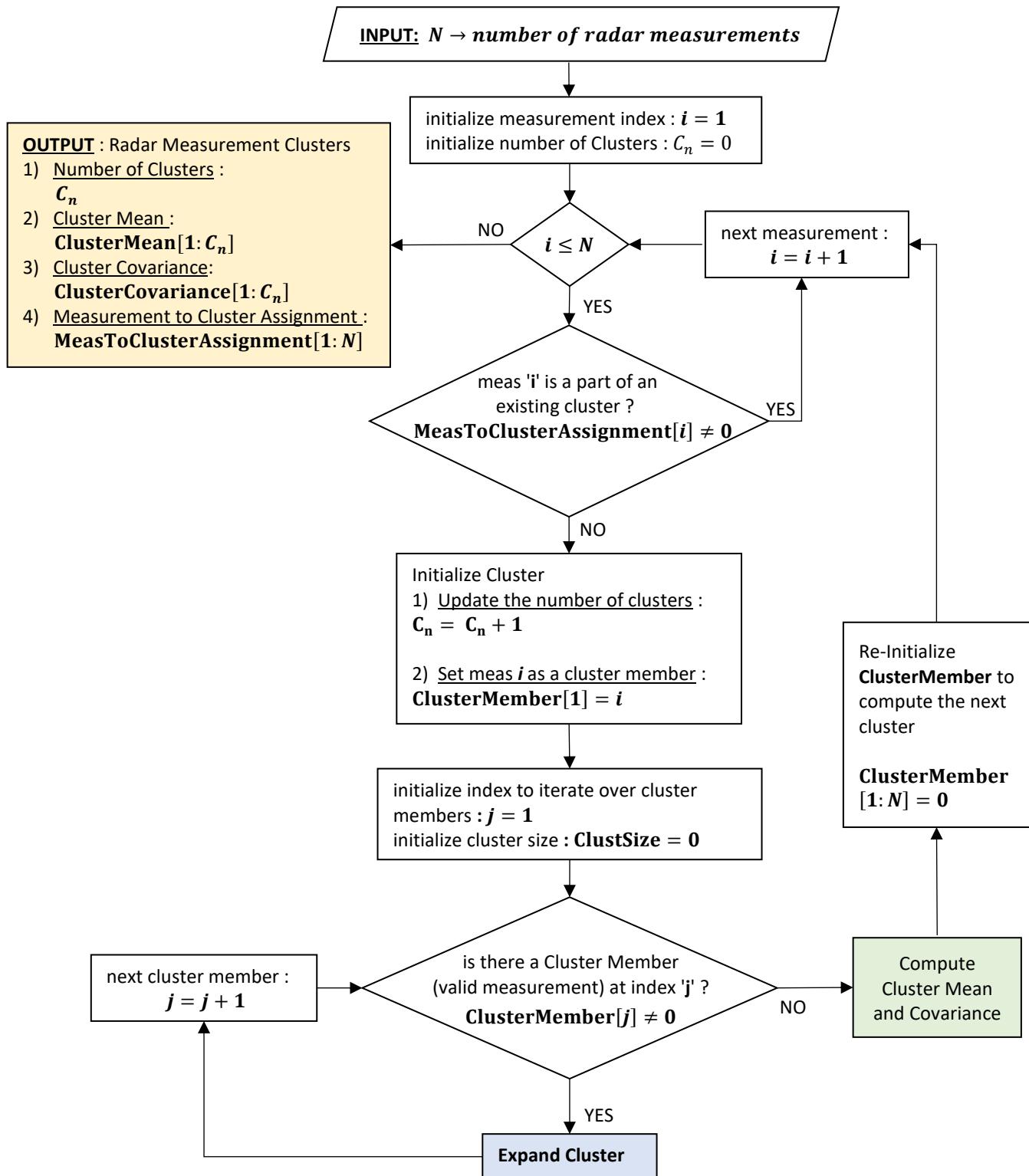


9.3.3 High - Level Design Of Camera Measurement Clustering

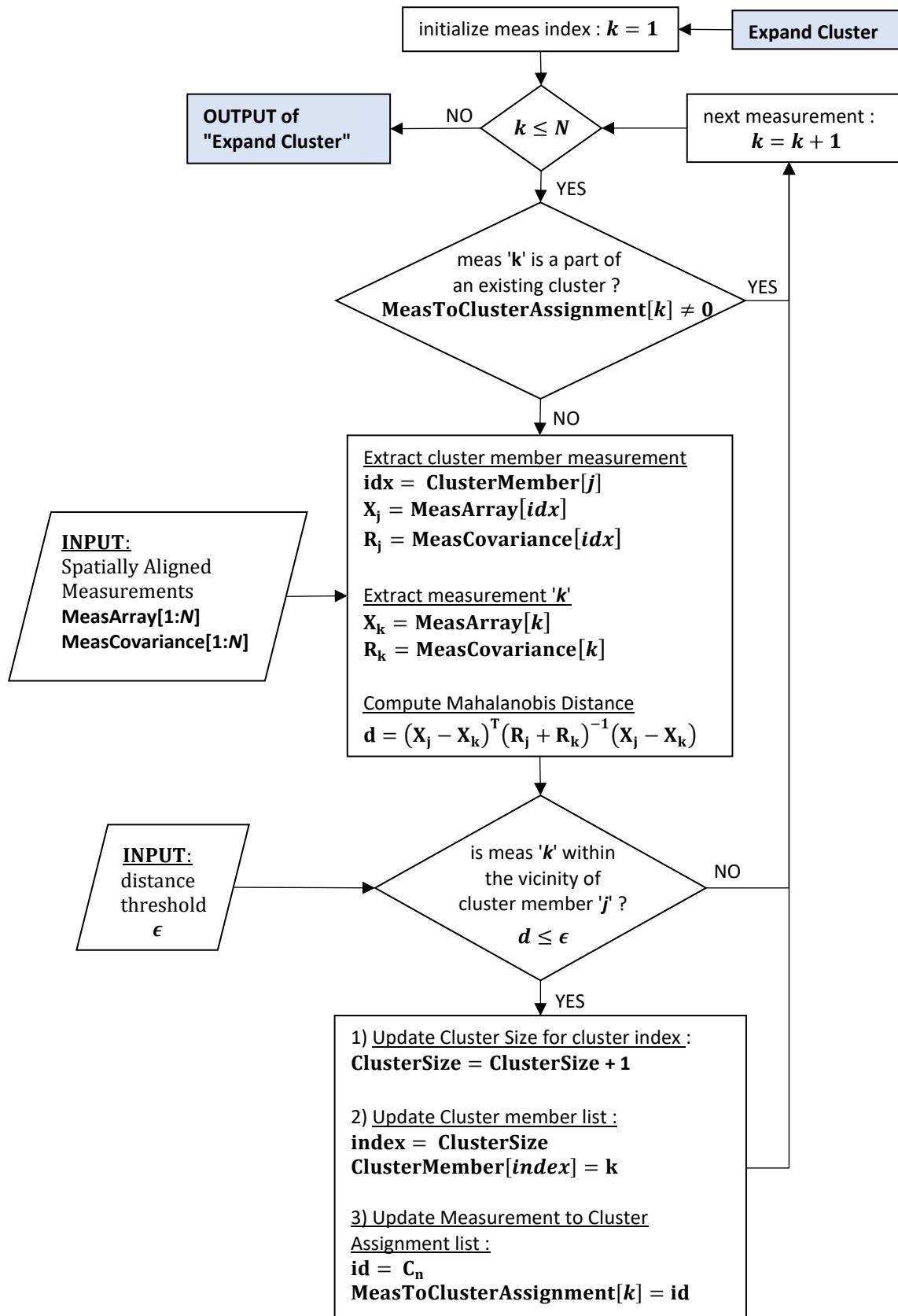


9.3.4 Detailed Design : Simplified DBSCAN Clustering

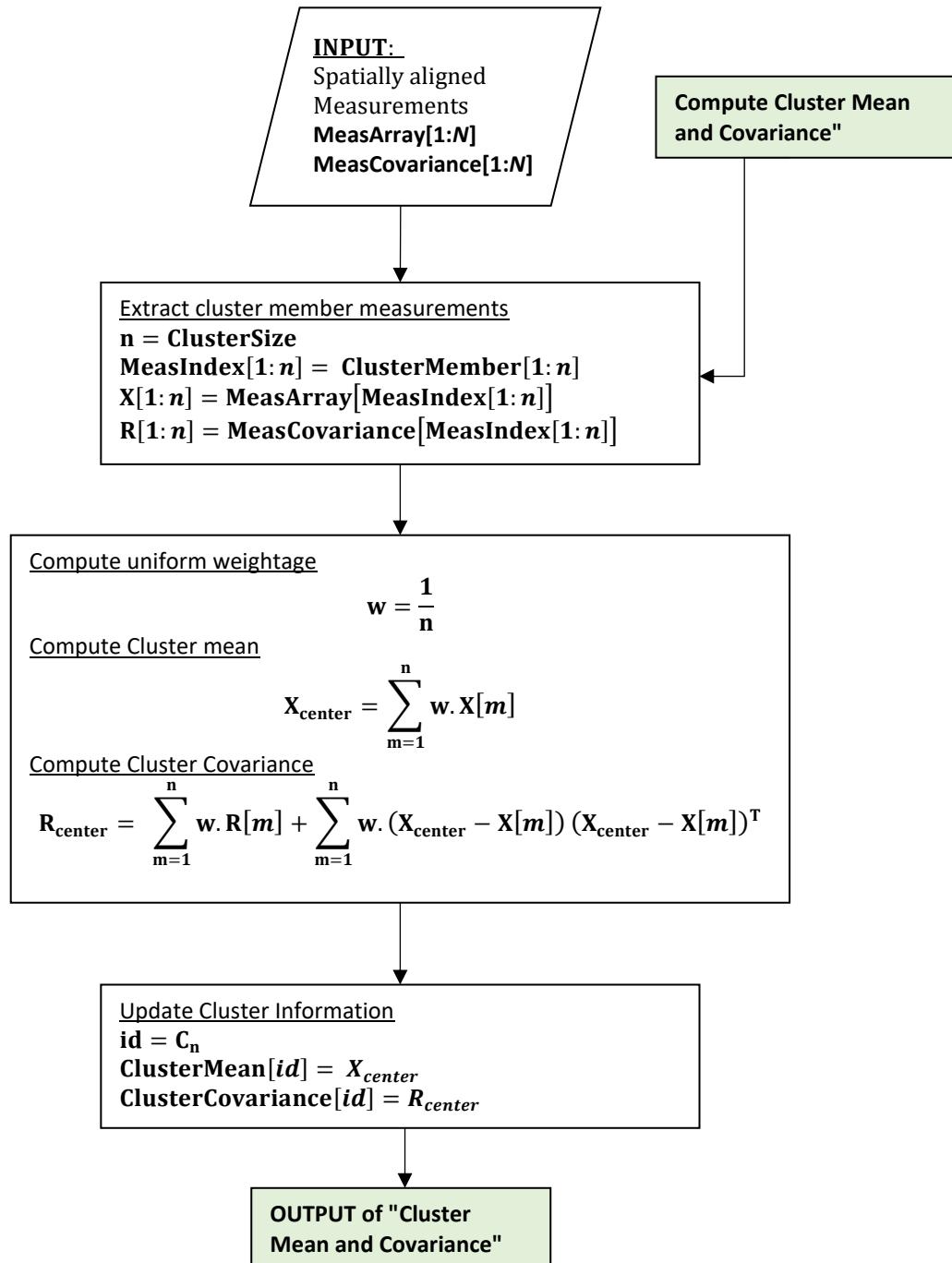
9.3.4.1 Part – 1 : Initialize Cluster



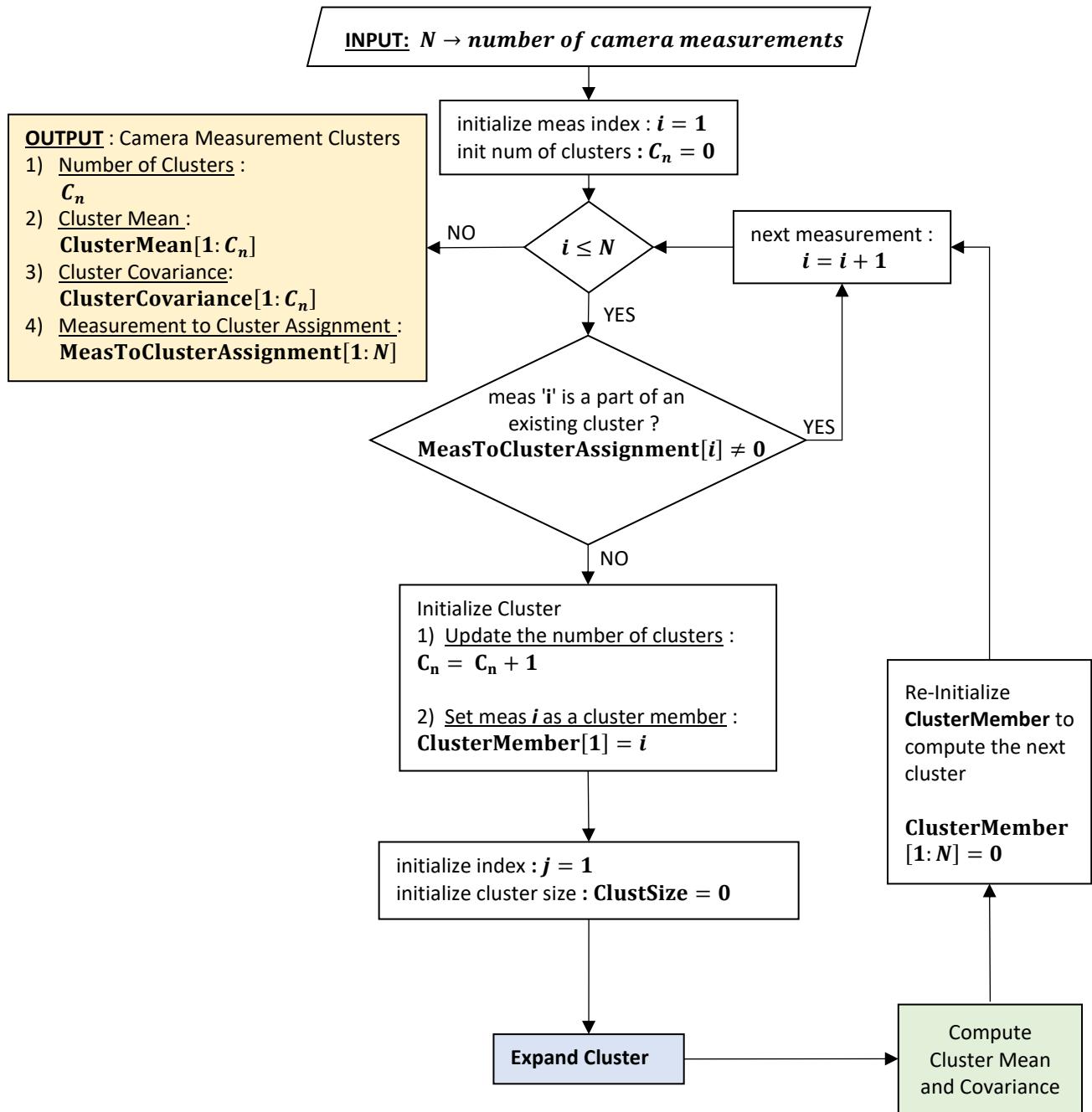
9.3.4.2 Part – 2 : Expand Cluster



9.3.4.3 Part – 3 : Compute Cluster Mean and Covariance



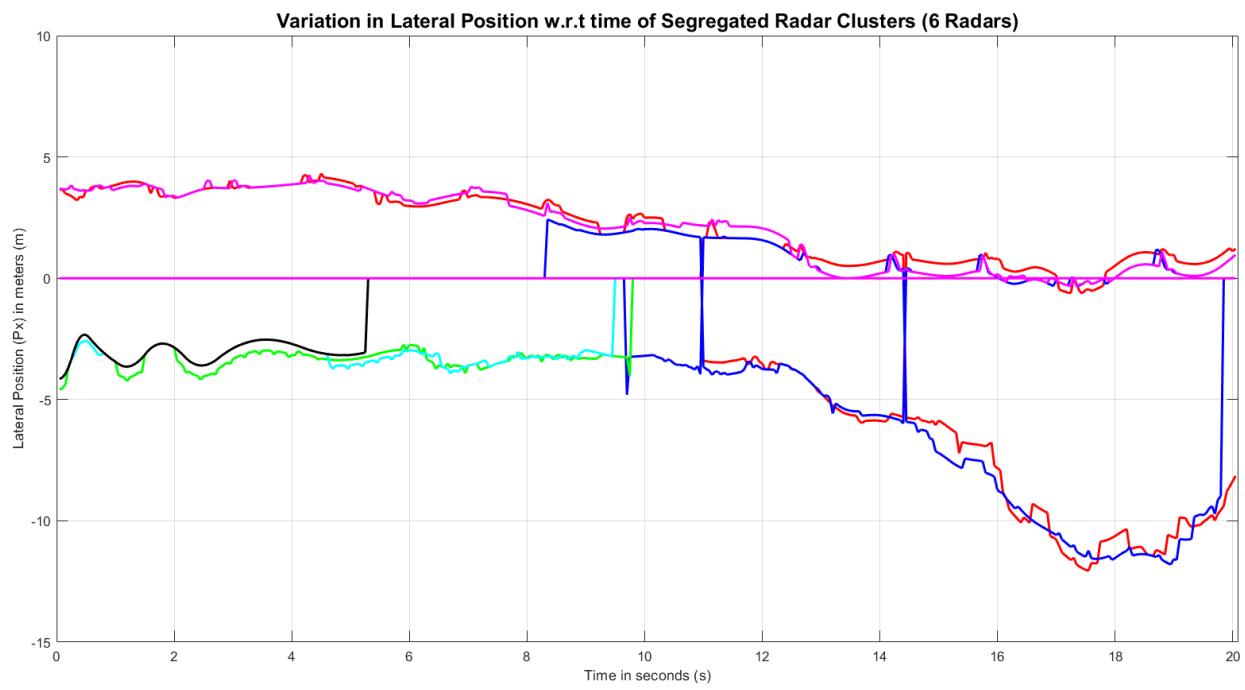
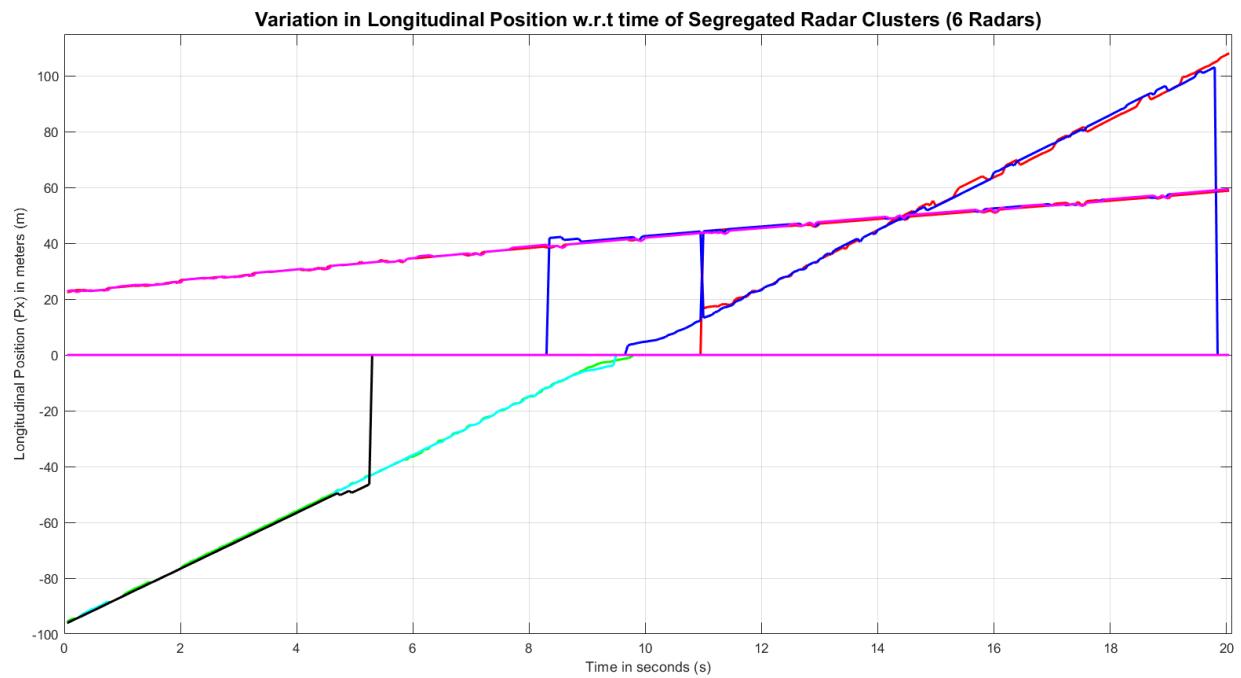
9.3.5 Detailed Design : NN Clustering



Note :

- “Expand Cluster” : Same as **Section 9.3.4.2** (Part – 2 : Expand Cluster for Simplified DBSCAN)
- “Compute Cluster Mean and Covariance” : Same as **Section 9.3.4.3** (Part – 3 : Compute Cluster Mean and Covariance for Simplified DBSCAN)
- *In version 2 of this project soft NN clustering shall be performed.*

9.3.6 Plots



9.4 State Prediction

9.4.1 Introduction

- Local Track estimates from Radar and Camera are maintained and the final estimated track parameters are the track to track fusion of Radar Track and Camera Track.
- In each of these 3 track estimates (Radar Local Track, Camera Local Track, Fused Track), the track parameters are predicted for the next fusion cycle.
- In the present scope Constant Acceleration Motion Model is used as a Process Model.
- The motion model describes our belief of how a tracked object's state evolves over time.

9.4.2 Formulation

INPUTS : Previous State Estimates of i^{th} object (Each for **Radar, Camera and Fused Tracks**)

1) state vector from the previous cycle : $\hat{X}_{k-1|k-1}^i$

$$\hat{X}_{k-1|k-1}^i = \begin{pmatrix} p_x \\ v_x \\ a_x \\ p_y \\ v_y \\ a_y \end{pmatrix}_{k-1|k-1}^i$$

$p_x \rightarrow$ longitudinal position of object 'i'
 $v_x \rightarrow$ longitudinal velocity of object 'i'
 $a_x \rightarrow$ longitudinal acceleration of object 'i'
 $p_y \rightarrow$ lateral position of object 'i'
 $v_y \rightarrow$ lateral velocity of object 'i'
 $a_y \rightarrow$ lateral acceleration of object 'i'

2) error covariance of the previous state estimate : $\hat{P}_{k-1|k-1}^i$

INPUTS : Process Model Parameters (for all of the **Radar, Camera and Fused Tracks**)

3) State Transition Model : A_k

$$A_k = \begin{pmatrix} \begin{pmatrix} 1 & \Delta T & 0.5\Delta T^2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & \Delta T & 0.5\Delta T^2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{pmatrix} \end{pmatrix}_k$$

$\Delta T \rightarrow$ fusion update cycle interval

4) Process Noise Covariance : Q_k

$(\sigma_{ax}^2, \sigma_{ay}^2) \rightarrow$ Variance
(errors in the constant acceleration assumption)

$$Q_k = \begin{pmatrix} \sigma_{ax}^2 \begin{pmatrix} \frac{\Delta T^5}{20} & \frac{\Delta T^4}{8} & \frac{\Delta T^3}{6} \\ \frac{\Delta T^4}{8} & \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^3}{6} & \frac{\Delta T^2}{2} & \Delta T \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \sigma_{ay}^2 \begin{pmatrix} \frac{\Delta T^5}{20} & \frac{\Delta T^4}{8} & \frac{\Delta T^3}{6} \\ \frac{\Delta T^4}{8} & \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^3}{6} & \frac{\Delta T^2}{2} & \Delta T \end{pmatrix} \end{pmatrix}_k$$

9.5 Measurements To Predicted Track Ellipsoidal Gating

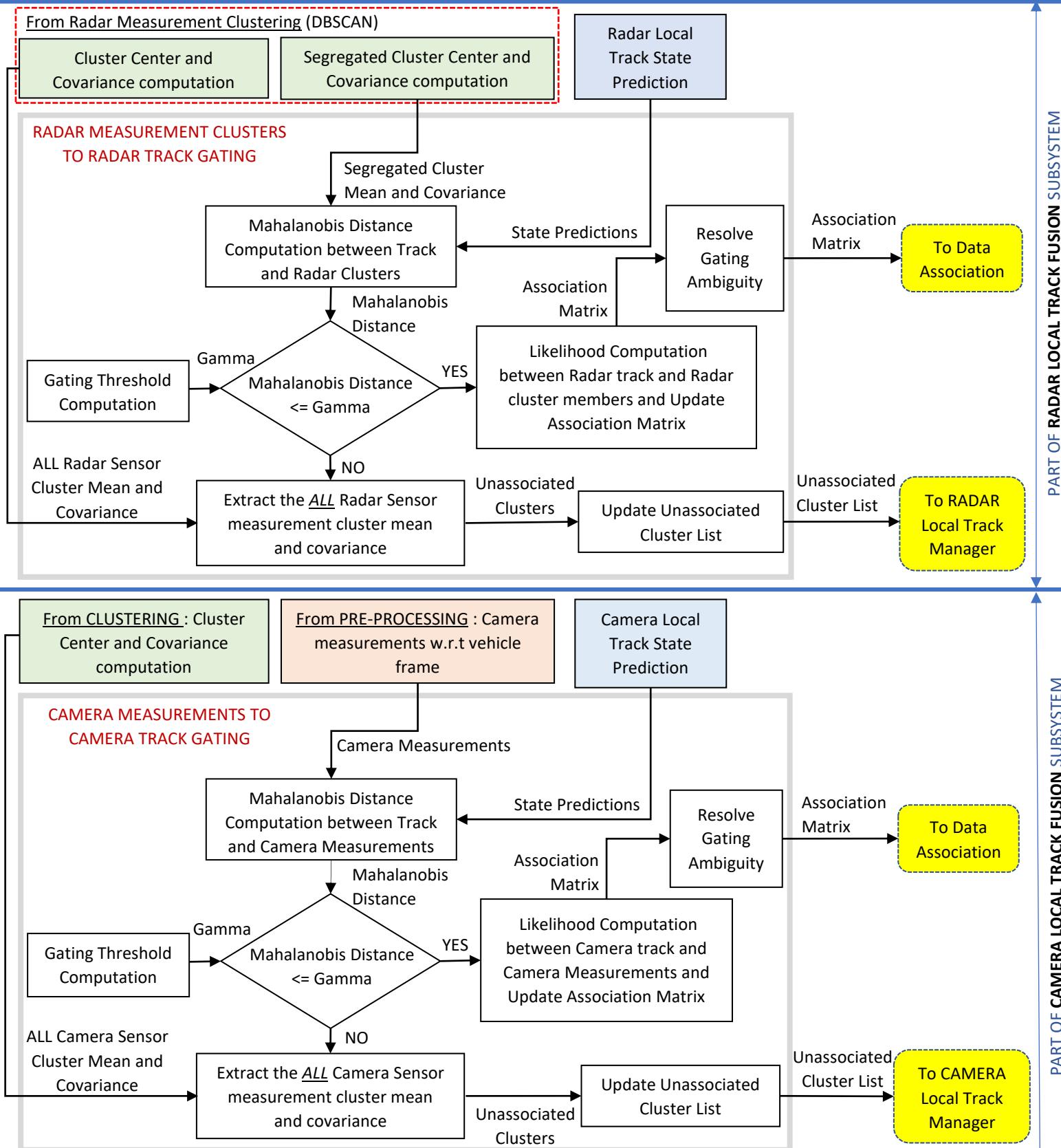
9.5.1 Introduction

- Gating is performed for each track.
- Same target may have multiple measurements from one sensor due to high sensor resolution. Likewise, same target may have measurements from multiple sensors if the sensor FOV's overlap and the traffic object is detected in the overlap region.
- To prevent association of false measurements (clutters) and incorrect track to measurement association a valid region around the track in the measurement space is defined, this process of determining the valid region by computing a threshold is known as validation gating.
- Depending on the measurement generated from track state parameters, sensor measurement, and innovation covariance (which is the error in the similarity between predicted track and gated measurement), a statistical similarity measure is used, which is known as Mahalanobis distance.
- The Mahalanobis distance is compared with the threshold commonly referred as gamma to determine if the measurement shall be associated with the track.

Likelihood Calculation and Resolve Gating Ambiguity.

- The measurement likelihood with respect to a track indicates how much likely the measurement has originated from the track.
- If a measurement is Gated with multiple tracks , then the gating ambiguity is resolved by associating to that track whose measurement likelihood is highest.

9.5.2 Module Architecture/Execution Flow



9.5.3 Formulation

INPUT

- Track State Prediction vector from k – 1 to k: $\hat{x}_{k|k-1}$
- Track State Prediction Error Covariance : $\hat{P}_{k|k-1}$
- measurement vector in vehicle frame: z_k^i
- measurement Noise Covariance in vehicle frame: R_k
- $H_k \rightarrow$ Measurement Model

Generated measurement from the predicted state

$$y_{k|k-1}^i = H_k \hat{x}_{k|k-1}$$

Squared of Euclidean distance between the measurement and hypothesised measurement

$$(z_k^i - y_{k|k-1}^i)^T (z_k^i - y_{k|k-1}^i)$$

$(z_k^i - y_{k|k-1}^i)$ is commonly referred as innovation

Error covariance computation of the innovation

$$S_k = H_k \hat{P}_{k|k-1} H_k^T + R_k$$

Mahalanobis distance computation

$$d_k = (z_k^i - H_k \hat{x}_{k|k-1})^T (H_k \hat{P}_{k|k-1} H_k^T + R_k)^{-1} (z_k^i - H_k \hat{x}_{k|k-1})$$

Threshold Computation (Gamma)

$$\gamma_k = f(S_k)$$

Gating

if $d_k \leq \gamma_k$

then Measurement z_k^i is gated with Track $\hat{x}_{k|k-1}$

Measurement Likelihood probability Computation for gated measurement

where,

$P_D^j \rightarrow$ probability of detection of sensor 'j'

$m_k^i \rightarrow$ number of gated measurements with track i

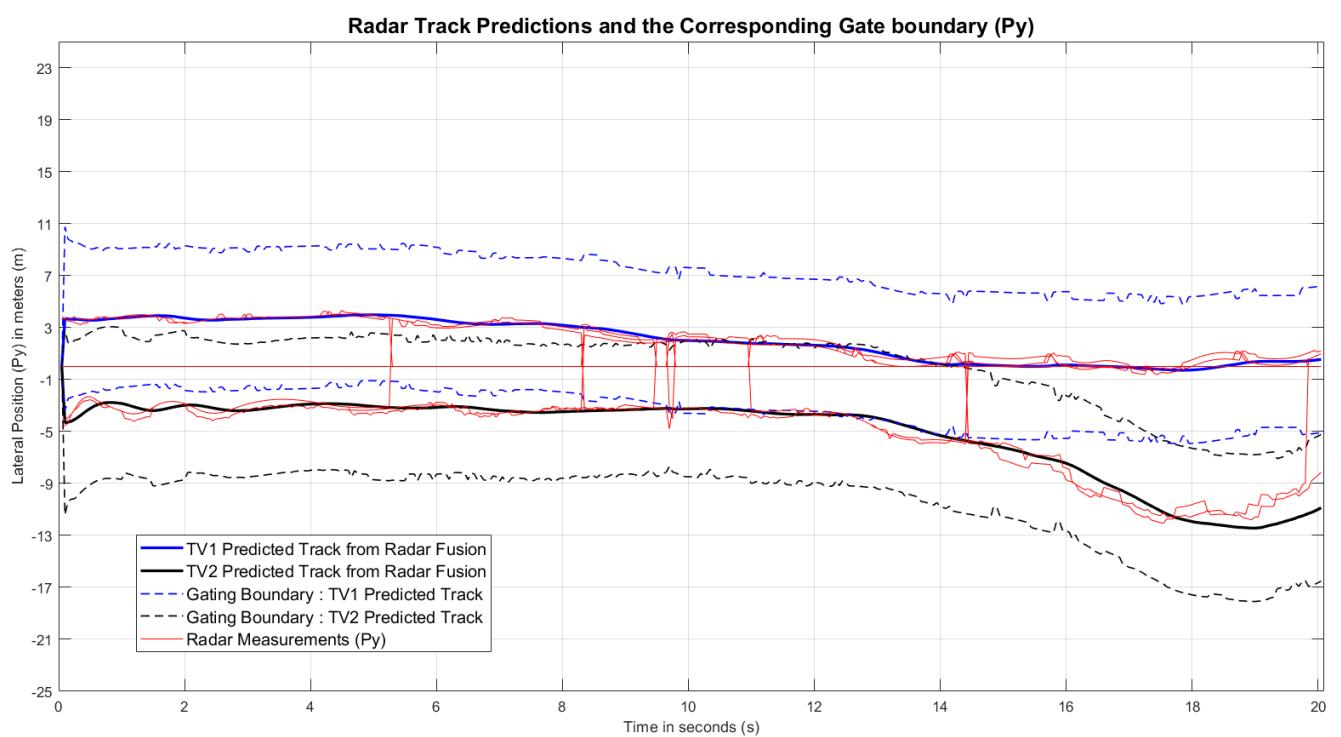
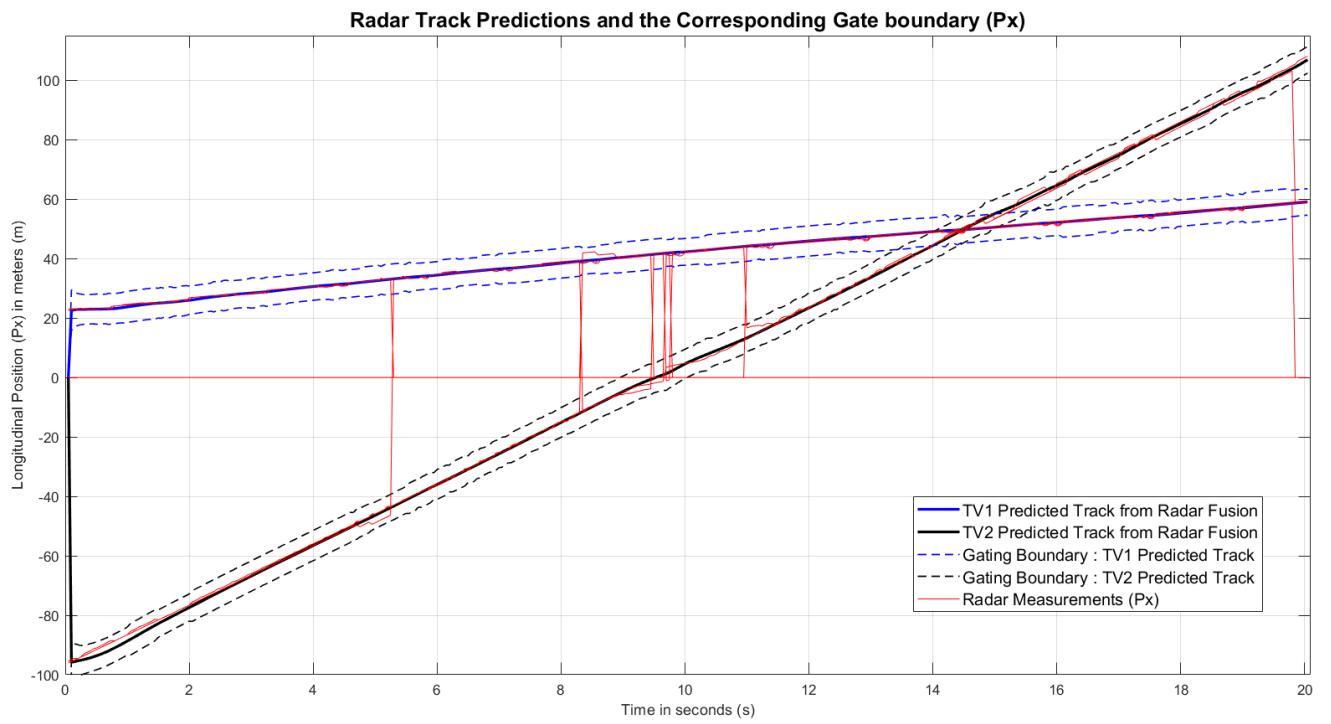
$$w_\theta = \frac{P_D^j \mathcal{N}(z_k^i; H_k \hat{x}_{k|k-1}, S_k)}{\lambda}$$

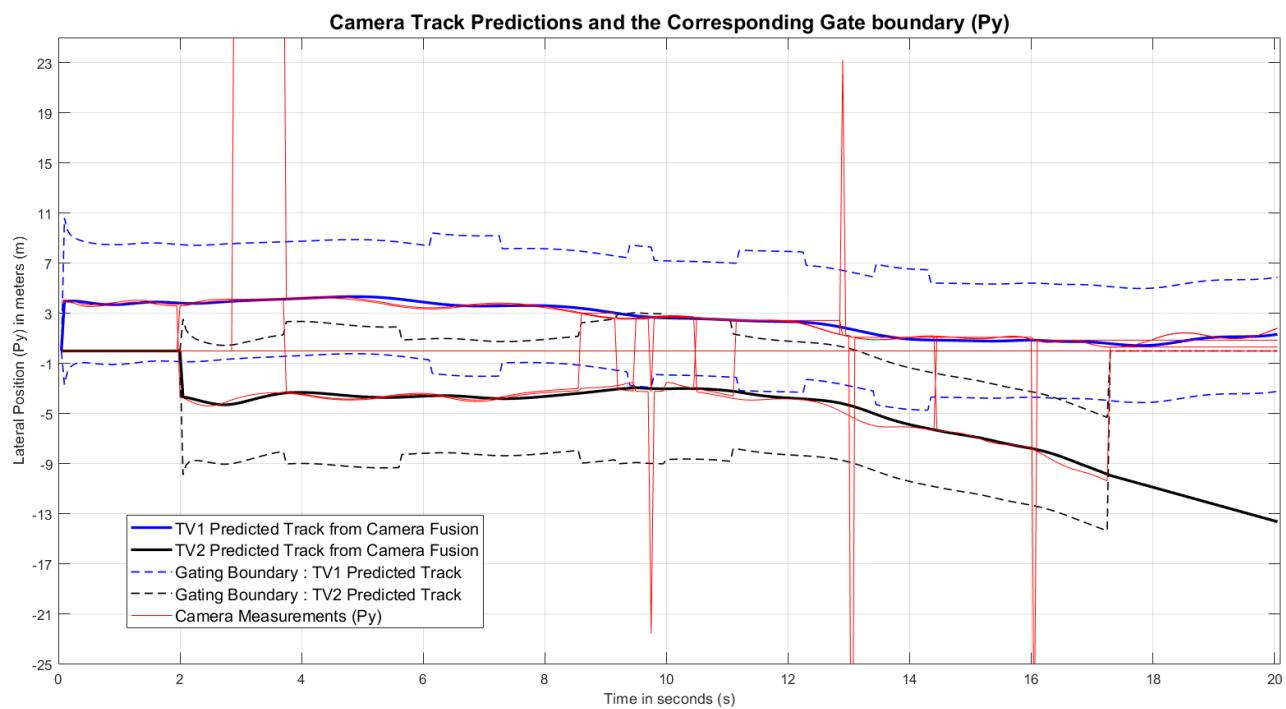
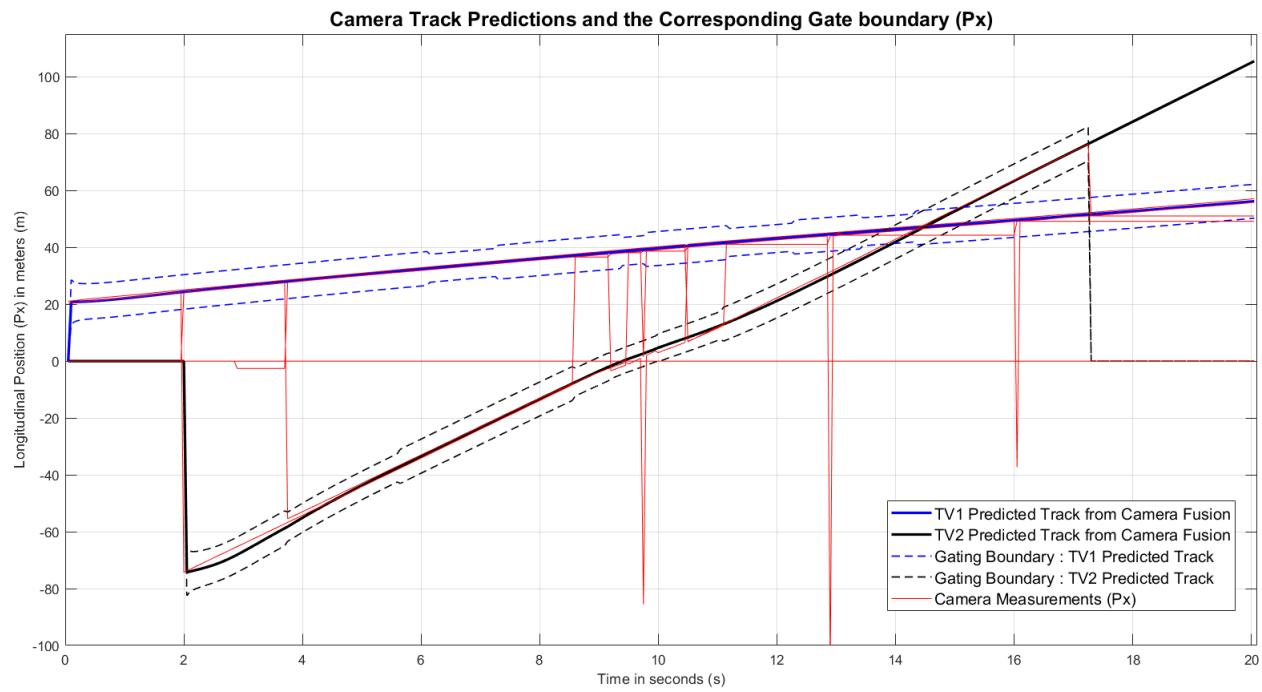
$$\mathcal{N}(z_k^i; H_k \hat{x}_{k|k-1}, S_k) = \frac{1}{\sqrt{|2\pi S_k|}} e^{-(d_k)}$$

$$\text{Gate Volume : } V_k = \frac{\pi^2}{2} \gamma_k \sqrt{S_k}$$

$$\text{Uniform Clutter density : } \lambda = \frac{m_k^i}{V_k}$$

9.5.4 Plots





9.6 Single Sensor Measurement To Track Data Association

9.6.1 Introduction

Depending on the sensor type, if the sensor generates random clutter frequently and prone to miss detections PDAF is used.

The PDAF assumptions are as follows

1. Each sensor generates at most one measurement from the object, i.e. the objects are assumed to be point objects and they produce zero or one measurement per scan randomly. If multiple measurements are gated with a track it is assumed that one measurement is track originated and the remaining are from clutter, accordingly association probabilities are computed based on the above disjoint events.
2. Only measurements that fall within a proximity of the expected measurement (i.e., that fall within the validation gate) are considered for processing
3. Measurements are produced by the sensor with infinite resolution. In other words, each measurement can have only one source which excludes the events where two objects generate one measurement due to finite sensor resolution.
4. The object may or may not be detected all the time and the detection probability is denoted by P_D
5. All non-object originated measurements are assumed to be originated from clutter that is uniformly distributed in space
6. The object motion obeys linear Gaussian statistics, possible object trajectory models are assumed known, and the models are assumed to propagate as Markov chains
7. The measurement noise is white Gaussian.
8. The object track has been initialized by Track Management.

9.6.2 Association Probability Computation

INPUT

$m_k^i \rightarrow$ Total number of measurements generated by the installed sensor at time k which are within the gate

$\hat{x}_{k|k-1} \rightarrow$ Predicted state of a track

$P_D^i \rightarrow$ Probability of detection corresponding to sensor 'i'

$P_G \rightarrow$ Probability of Gating

$Z_k^i = \{z^1, z^2, \dots, z^{m_k^i}\} \rightarrow$ Gated Set of measurements receive from the sensor 'i' at time k

The data association is described as a set of disjoint events: $(\theta_0, \theta_1, \dots, \theta_{m_k^i})$, where

$\theta = \theta_0$: if the object is not detected and the gated measurements are clutter detections

$\theta = \theta_1$: if the z^1 measurement is an object detection and the remaining $m_k - 1$ number of measurements are clutter detections

$\theta = \theta_2$: if the z^2 measurement is an object detection and the remaining $m_k - 1$ number of measurements are clutter detections

$\theta = \theta_1$: if the z^1 measurement is an object detection and the remaining $m_k - 1$ number of measurements are clutter detections

Case 1: $\theta = \theta_0$

An object is detected with probability P_D^i ,

if detected probability that the detection is within the gate is $P_D^i P_G$

So the probability of miss-detection is: $(1 - P_D^i P_G)$

Association weight for miss-detection: $w_0 = (1 - P_D^i P_G)$

Case 2: $\theta = \theta_1, \theta_2, \dots, \theta_{m_k}$

Assuming the measurement corresponding to θ : (z^θ) is originated from the track, the corresponding association weight is computed as follows:

Assuming the object is detected from sensor 'i' with probability : P_D^i

Measurement likelihood: $\mathcal{N}(z^\theta; H_k \hat{x}_{k|k-1}, S_k)$

$$\mathcal{N}(z^\theta; H_k \hat{x}_{k|k-1}, S_k) = \frac{1}{\sqrt{|2\pi S_k|}} e^{-0.5(z^\theta - H_k \hat{x}_{k|k-1})^T S_k^{-1} (z^\theta - H_k \hat{x}_{k|k-1})}$$

$$\text{Gate Volume} : V_k = \frac{\pi^2}{2} \gamma \sqrt{S_k}$$

$$\text{Uniform Clutter density} : \lambda = \frac{m_k}{V_k}$$

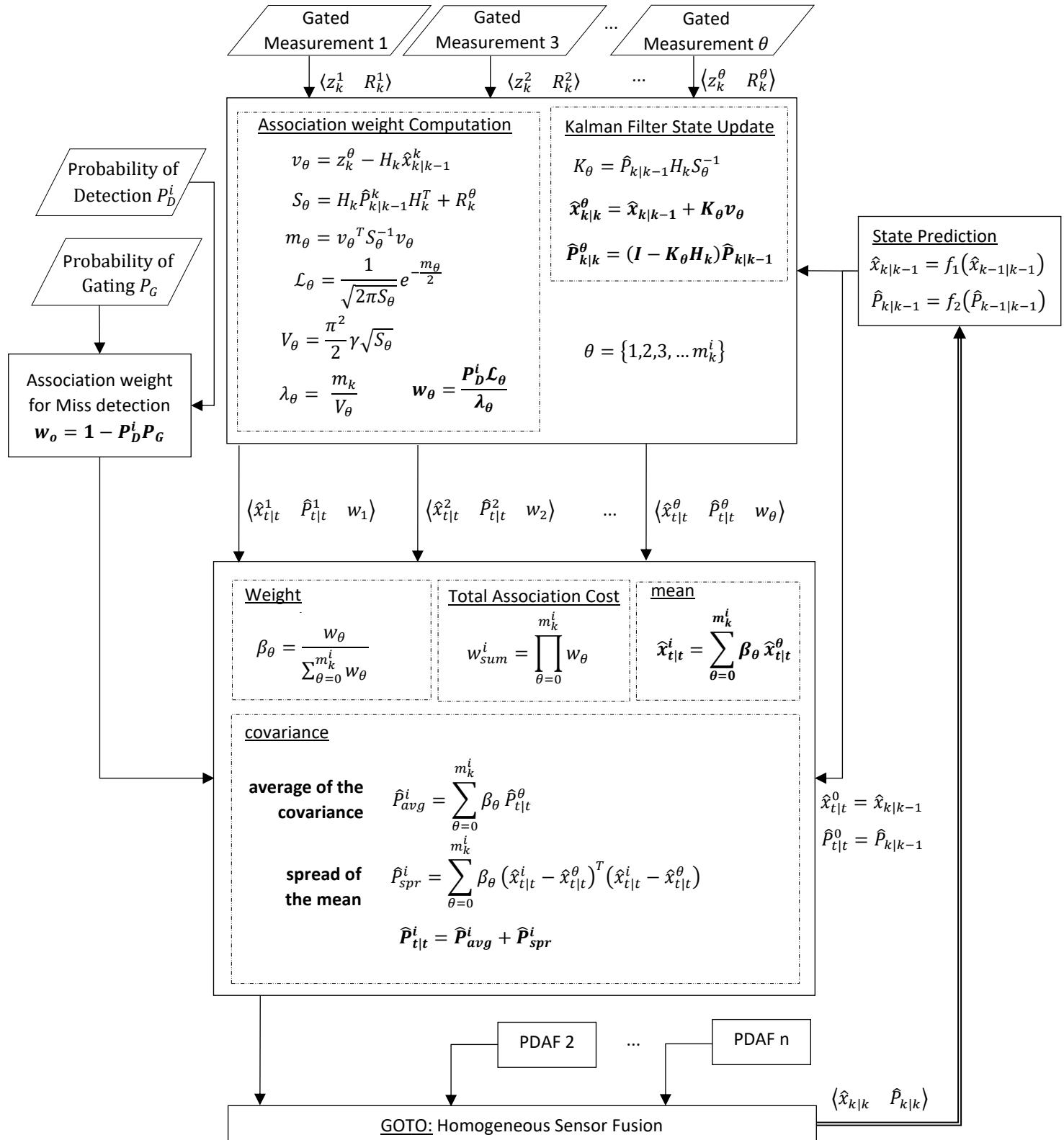
Association weight is computed as follows

$$w_\theta = \frac{P_D^i \mathcal{N}(z^\theta; H_k \hat{x}_{k|k-1}, S_k)}{\lambda}$$

$$w_\theta = \begin{cases} (1 - P_D^i P_G), & \theta = \theta_0 \\ \frac{P_D^i \mathcal{N}(z^\theta; H_k \hat{x}_{k|k-1}, S_k)}{\lambda}, & \theta = \theta_1 \text{ to } \theta_{m_k} \end{cases}$$

$$\beta_\theta = \frac{w_0}{\sum_{\theta=\theta_0}^{\theta_{m_k}} w_\theta}$$

9.6.3 Single Sensor Data Association Flowchart



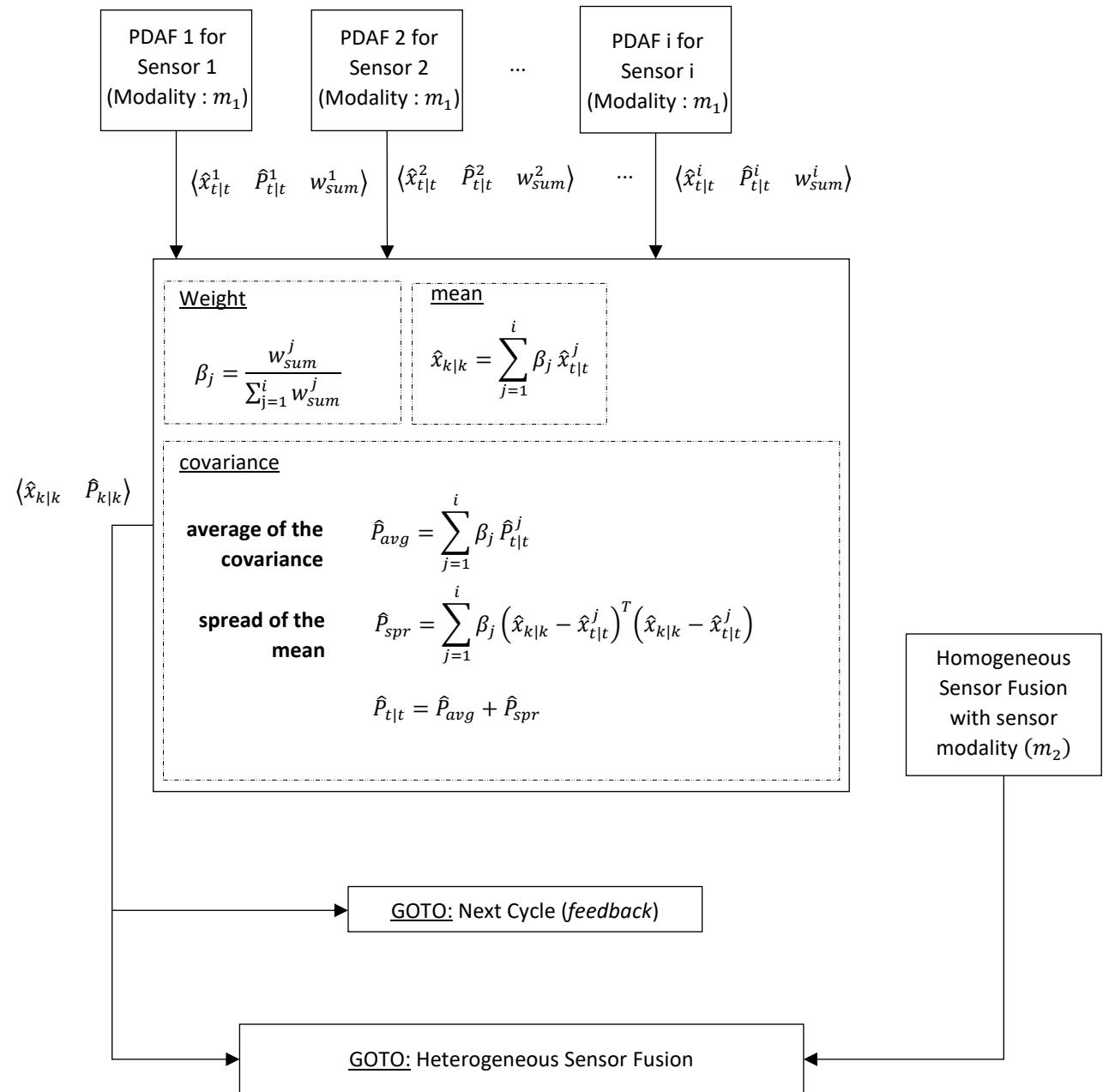
9.7 Multiple Sensor Fusion

9.7.1 Homogeneous Sensor Fusion Flowchart

Note :

The module architecture/execution flow for Radar Fusion is depicted in the architecture diagram in [Section 5.3.2](#)

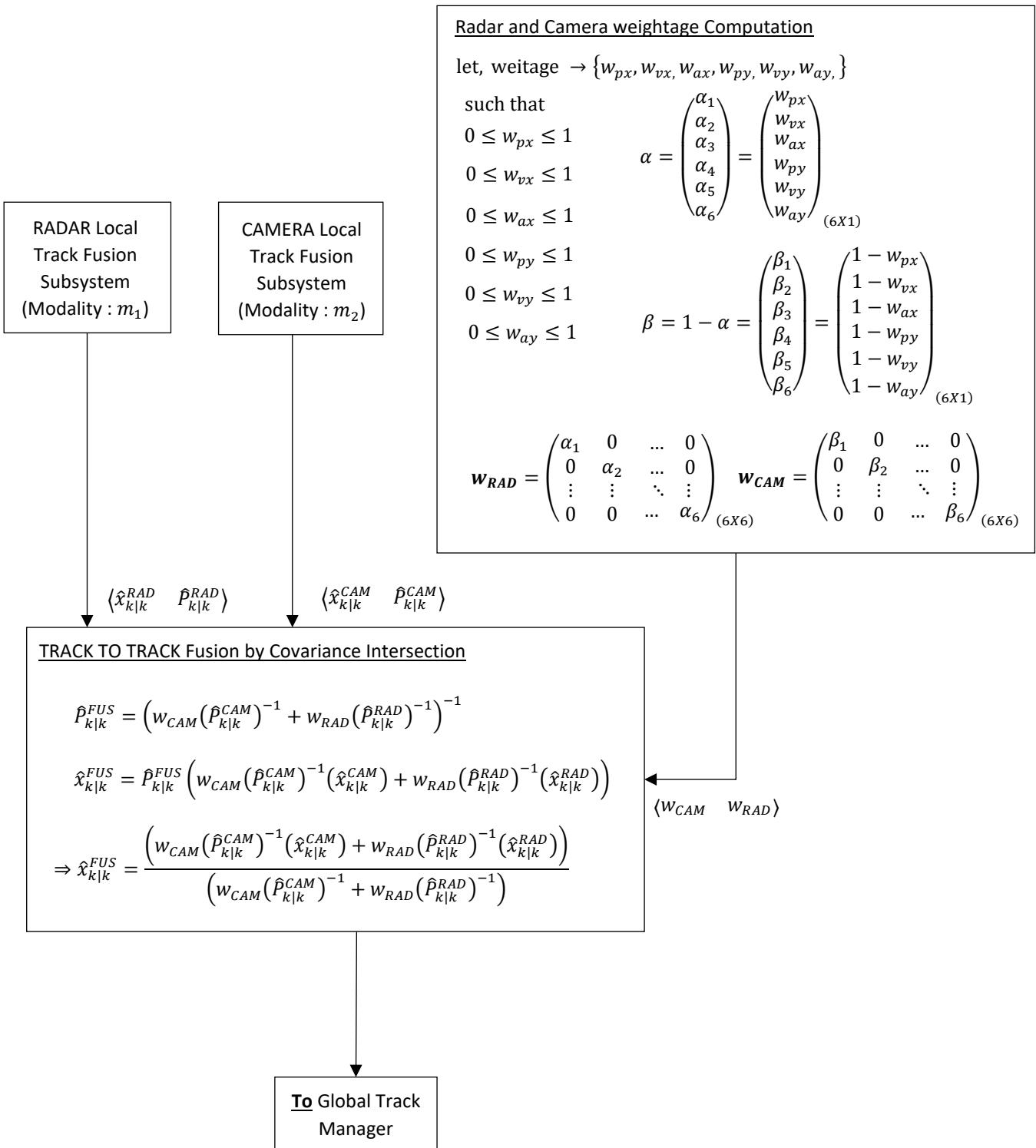
The module architecture/execution flow for Camera Fusion is depicted in the architecture diagram in [Section 5.3.3](#)



9.7.2 Heterogeneous Sensor Fusion Flowchart

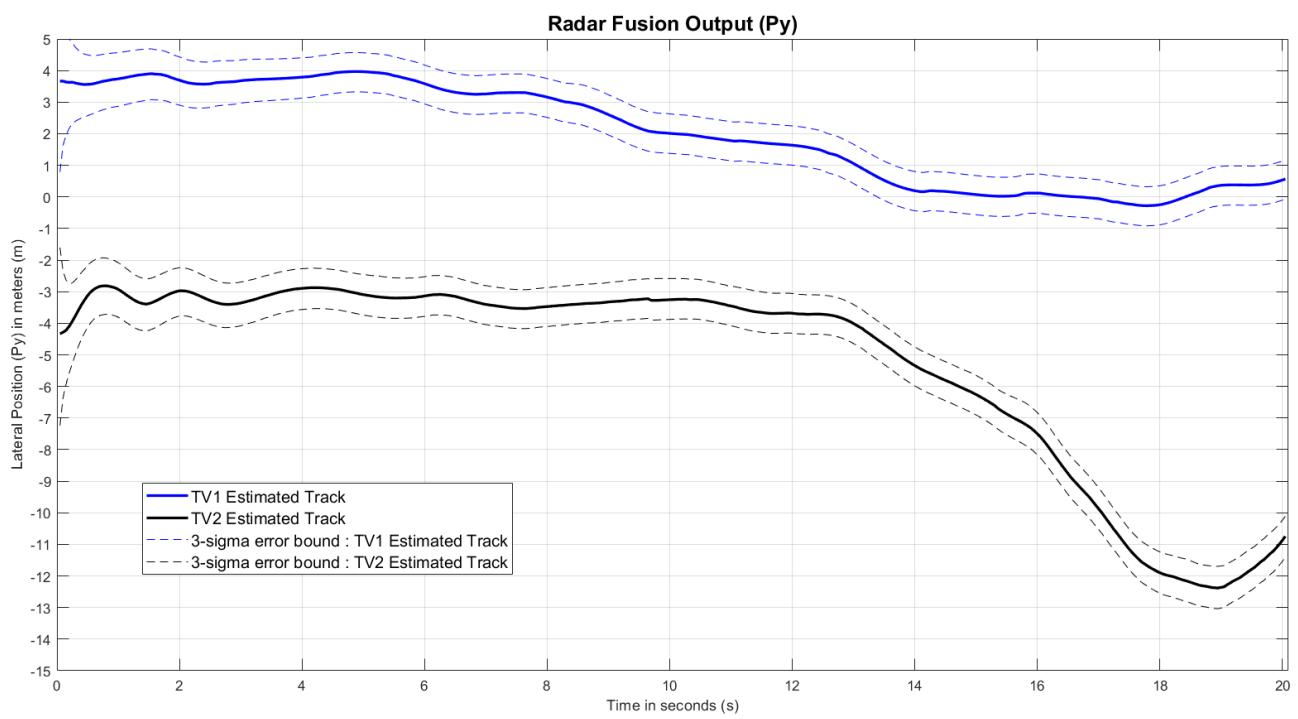
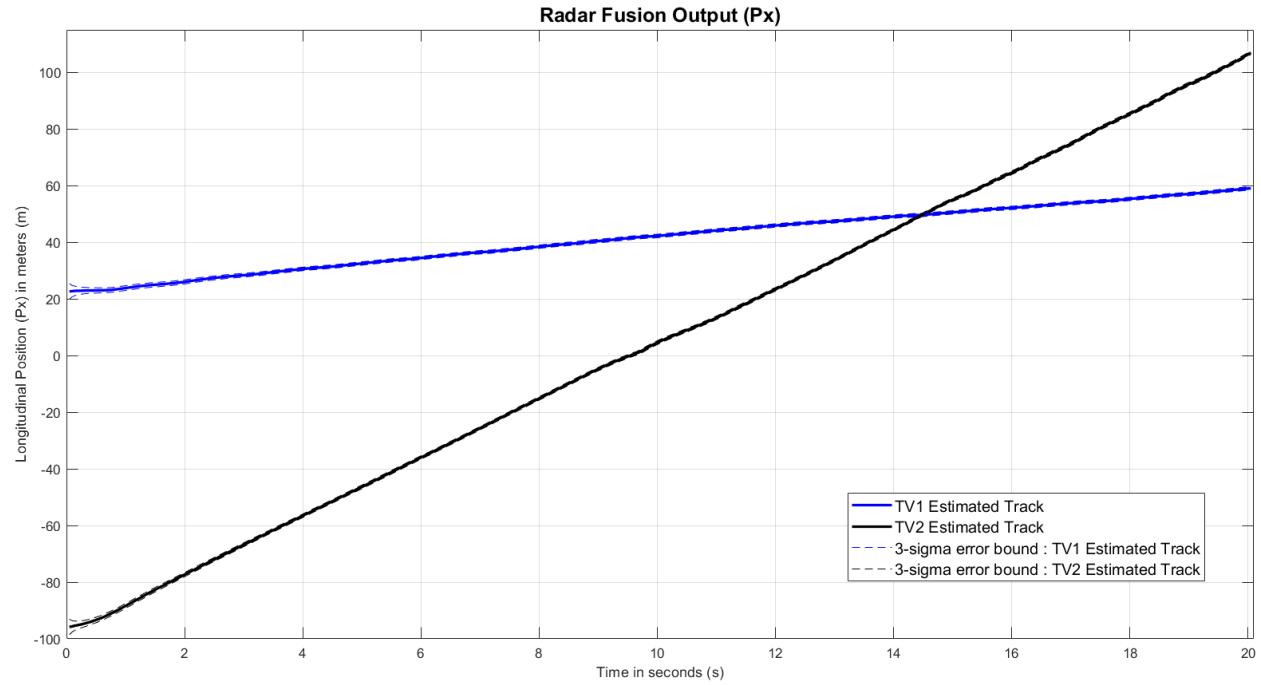
Note :

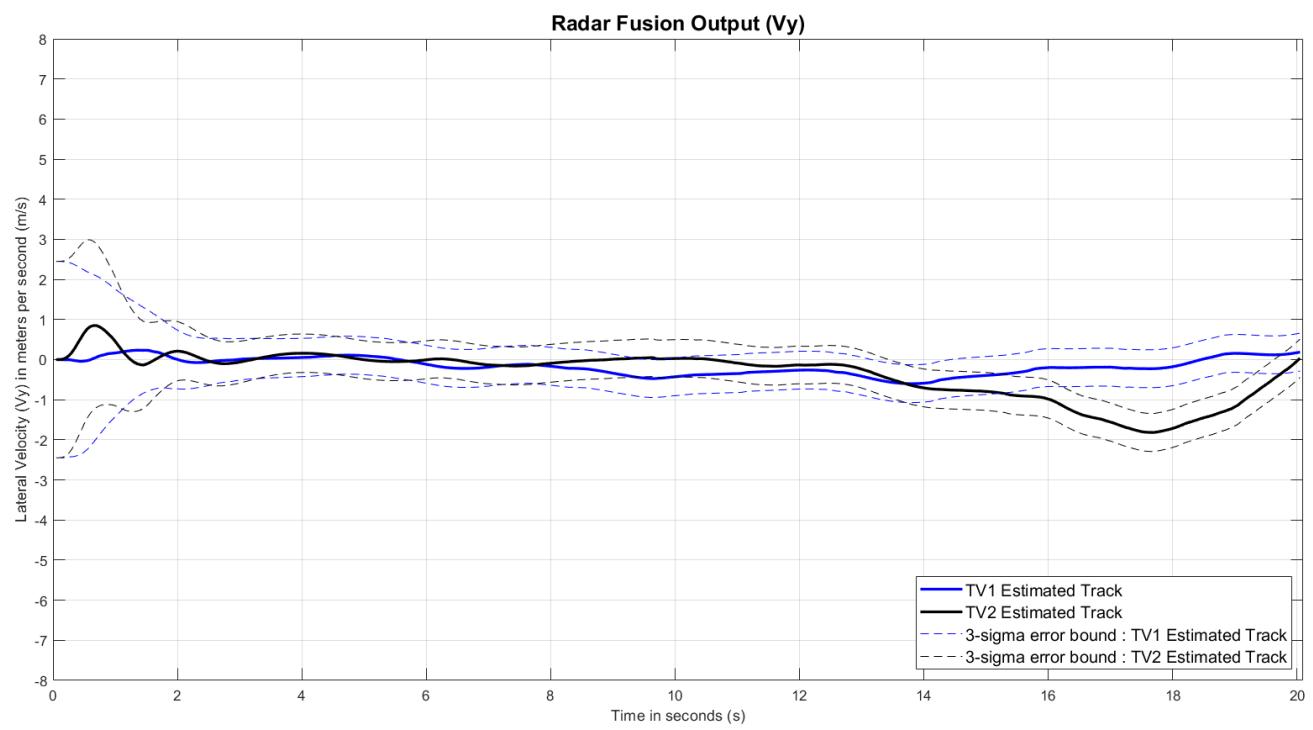
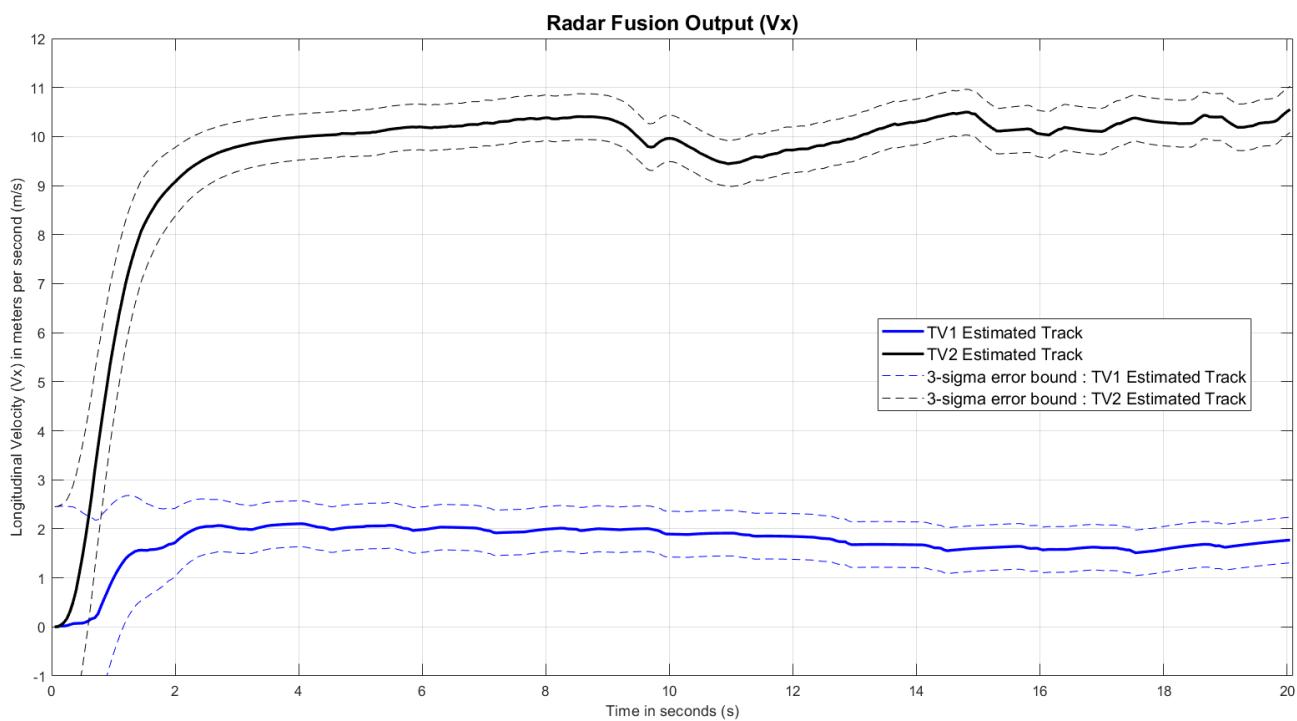
The architecture for Radar Track To Camera Track fusion subsystem is depicted in [Section 5.3.4](#)



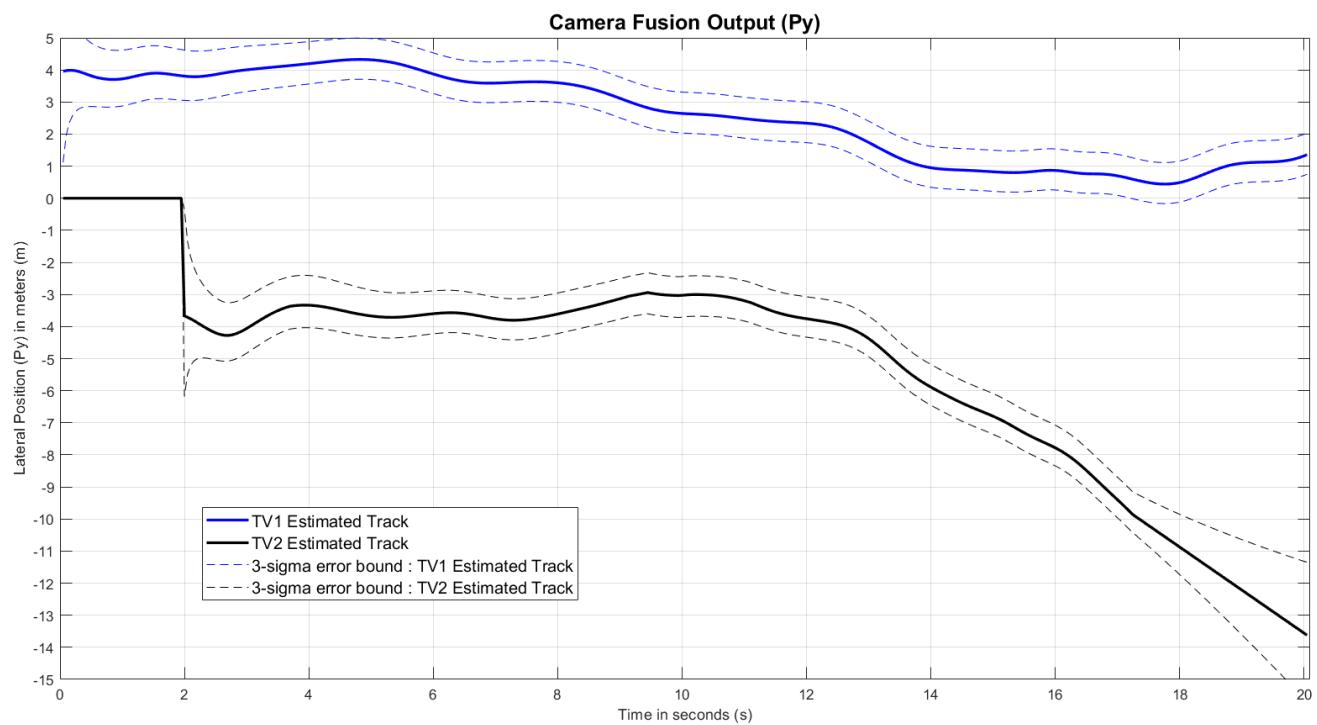
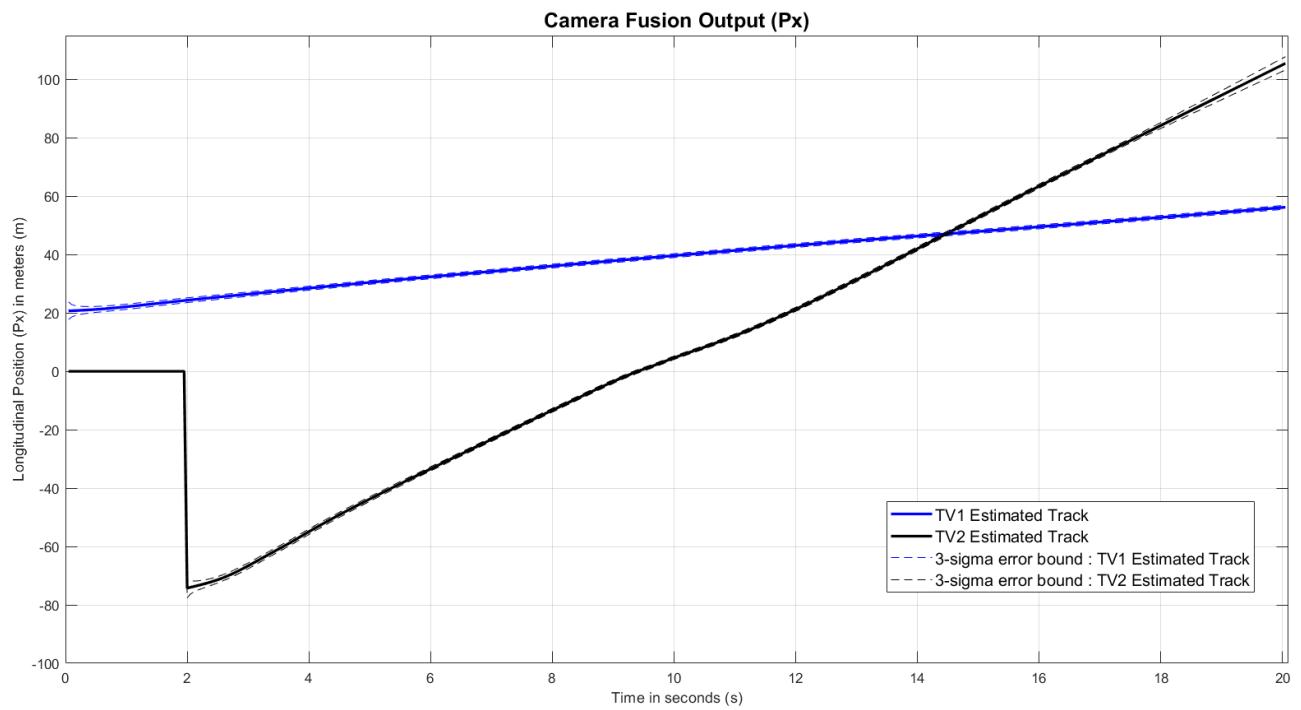
9.8 Plots

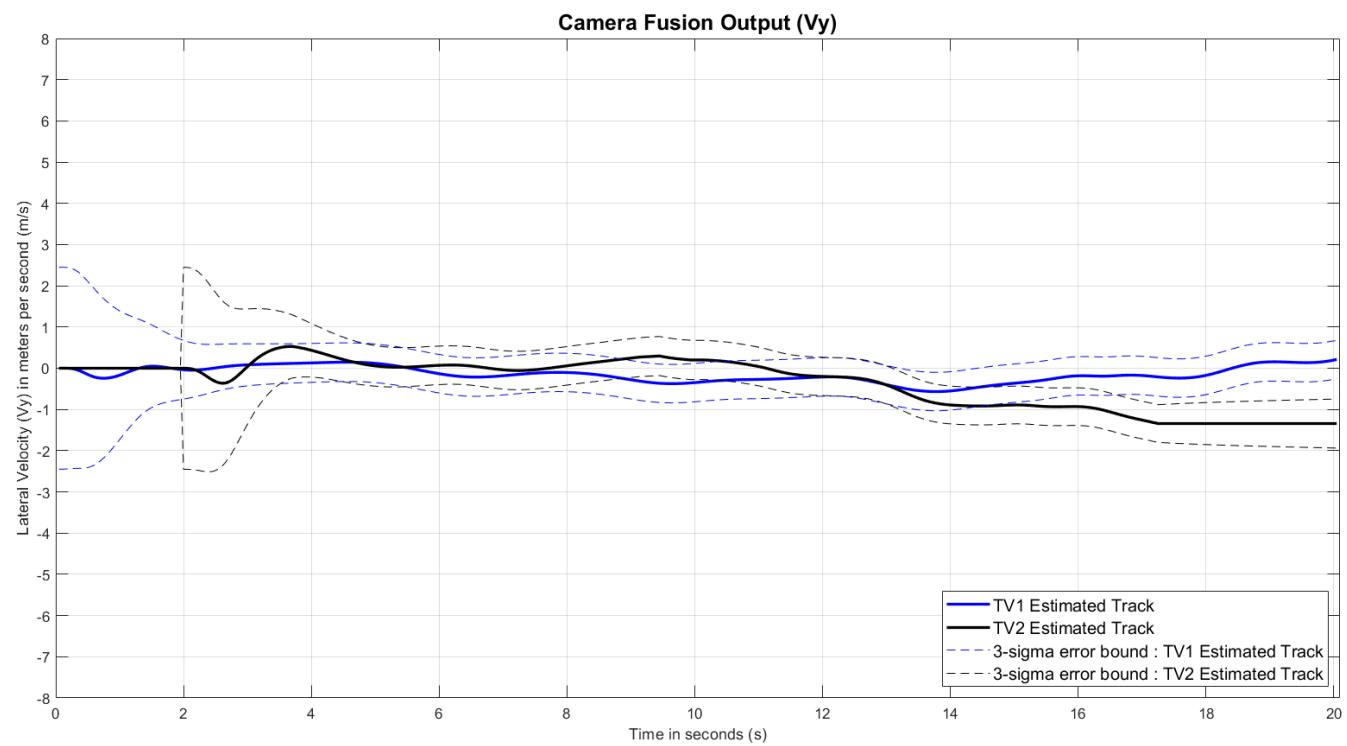
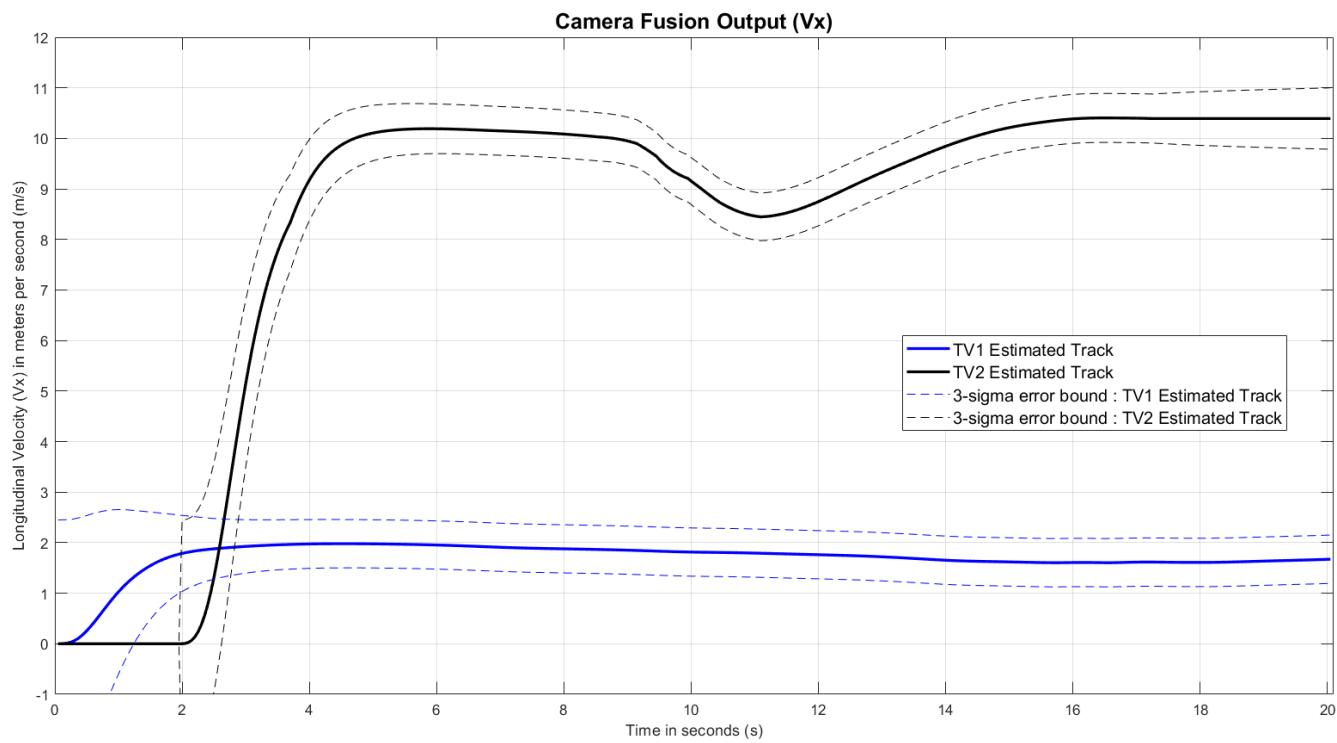
9.8.1 Radar Track Fusion Estimates



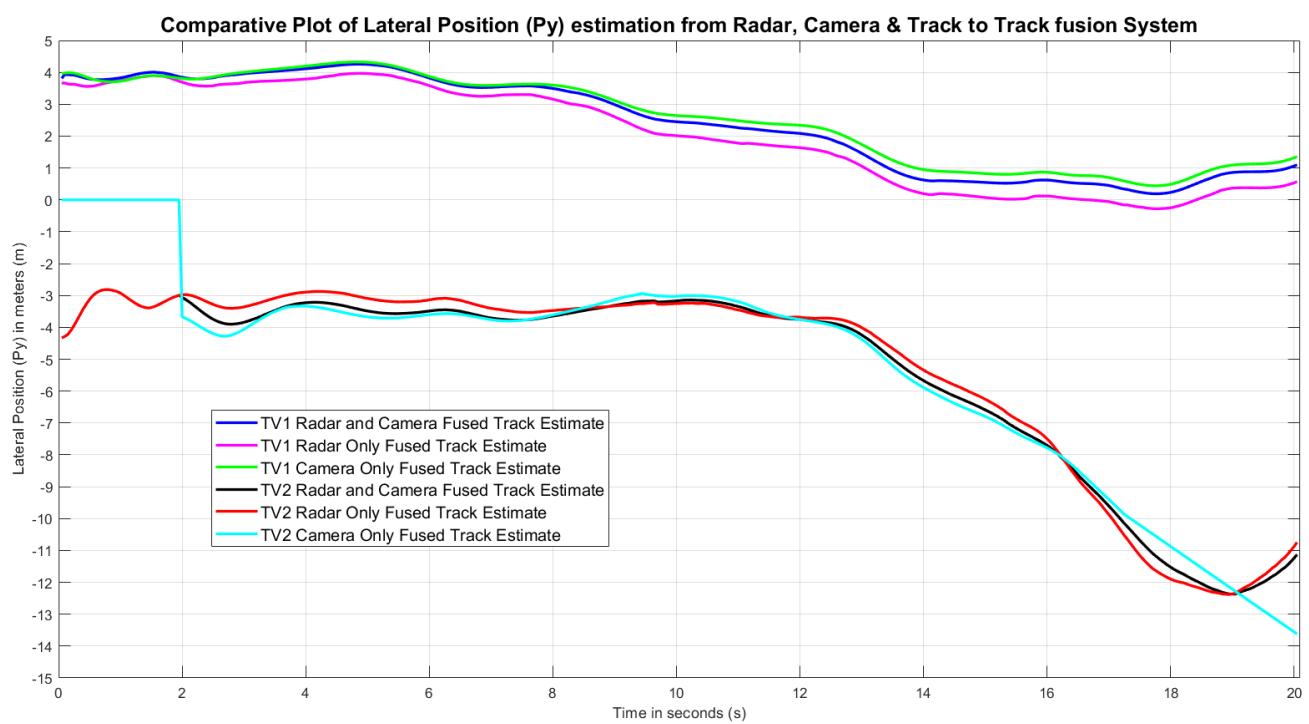
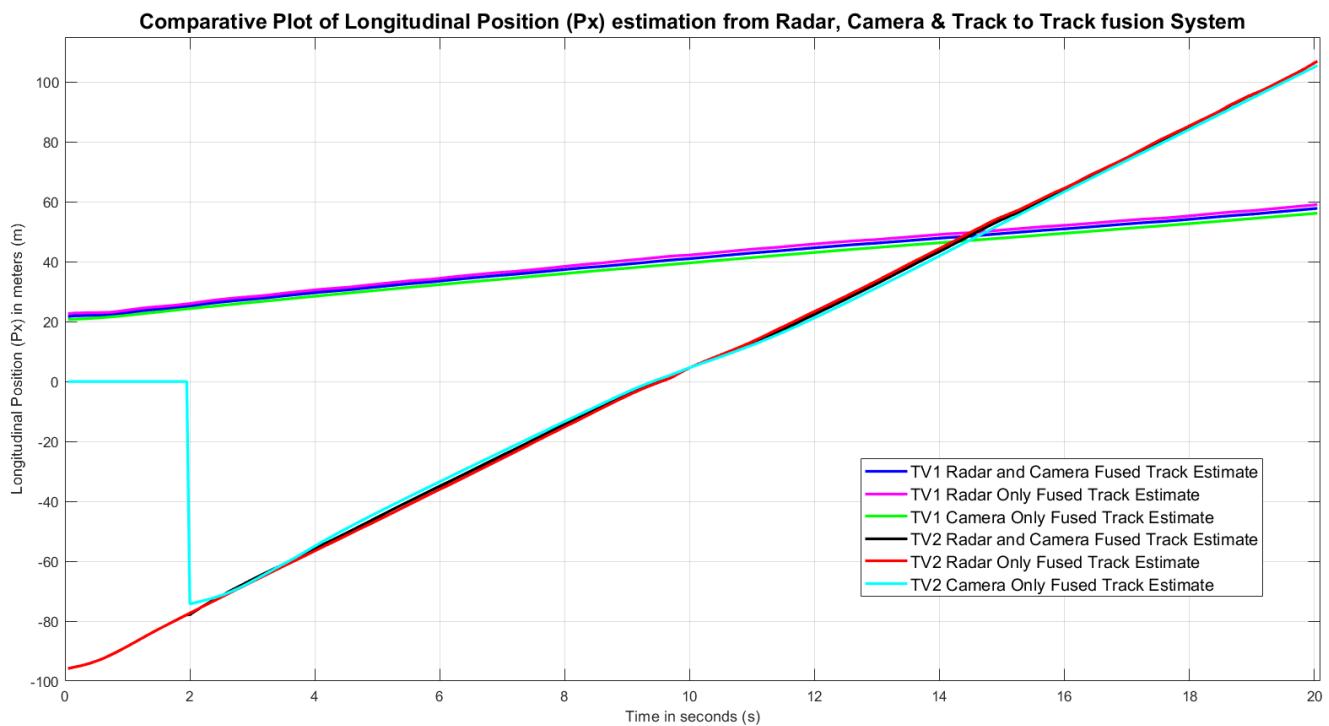


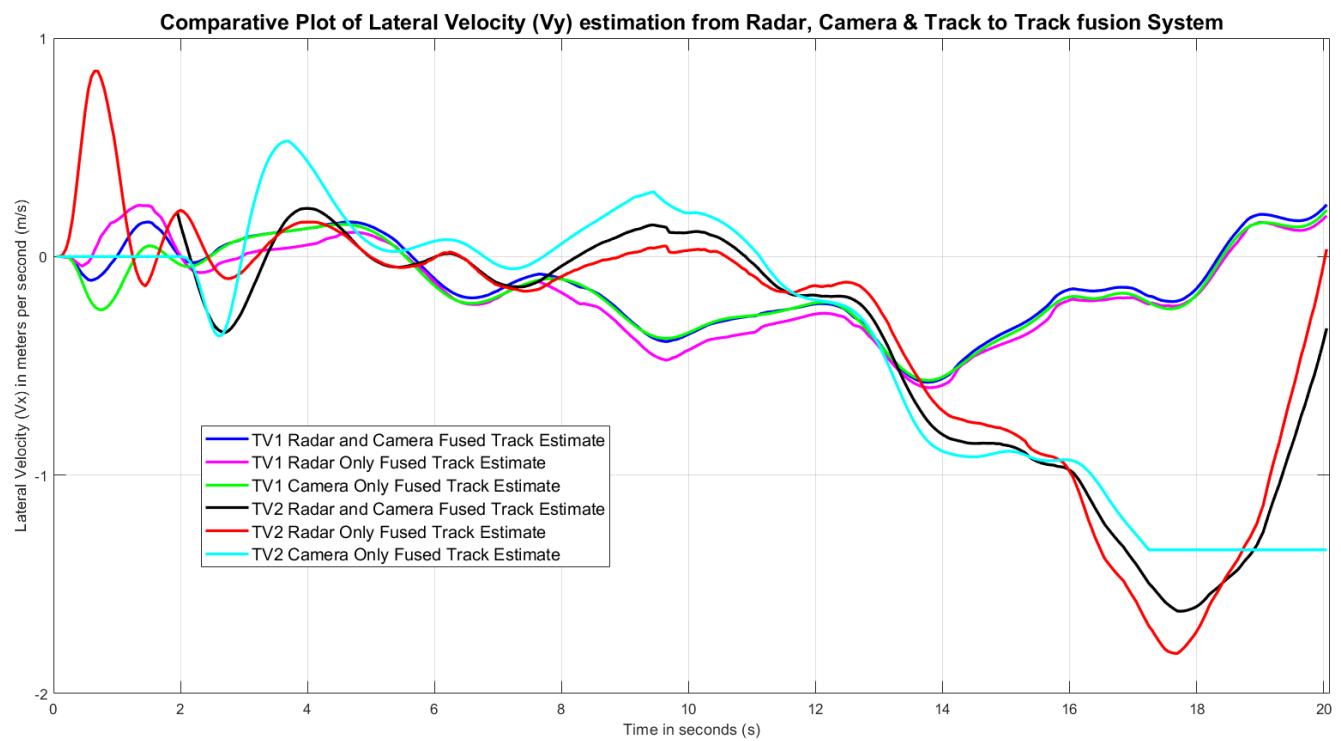
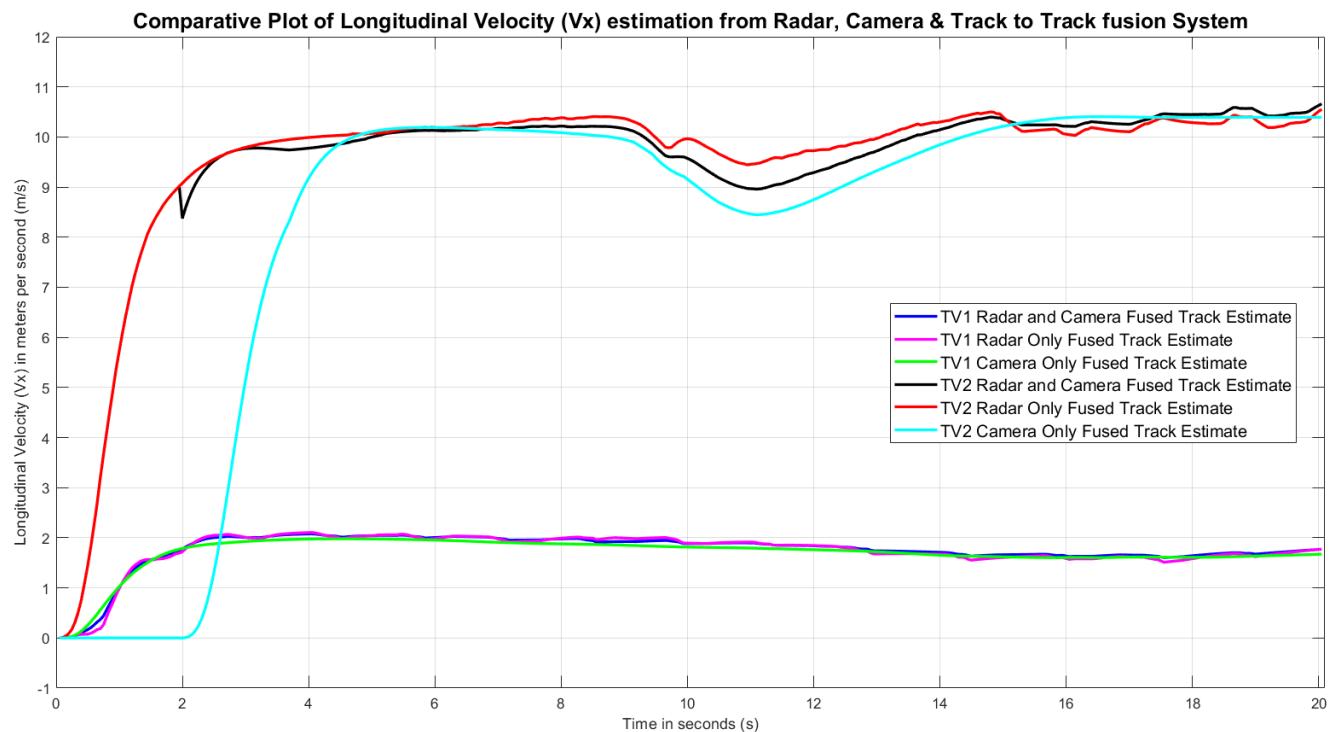
9.8.2 Camera Track Fusion Estimates





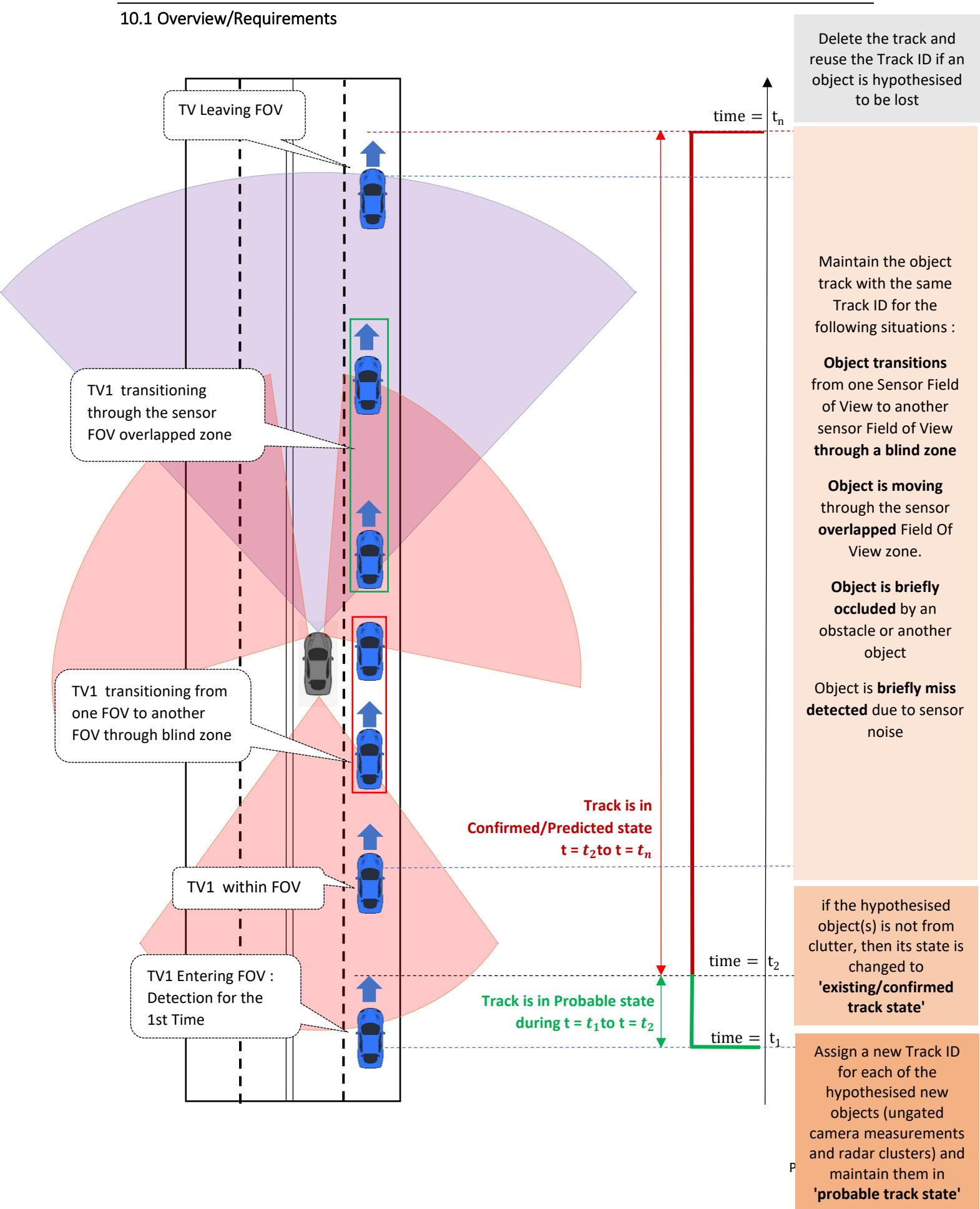
9.8.3 Radar and Camera Track To Track Fusion Estimates





10. TRACK MANAGEMENT

10.1 Overview/Requirements

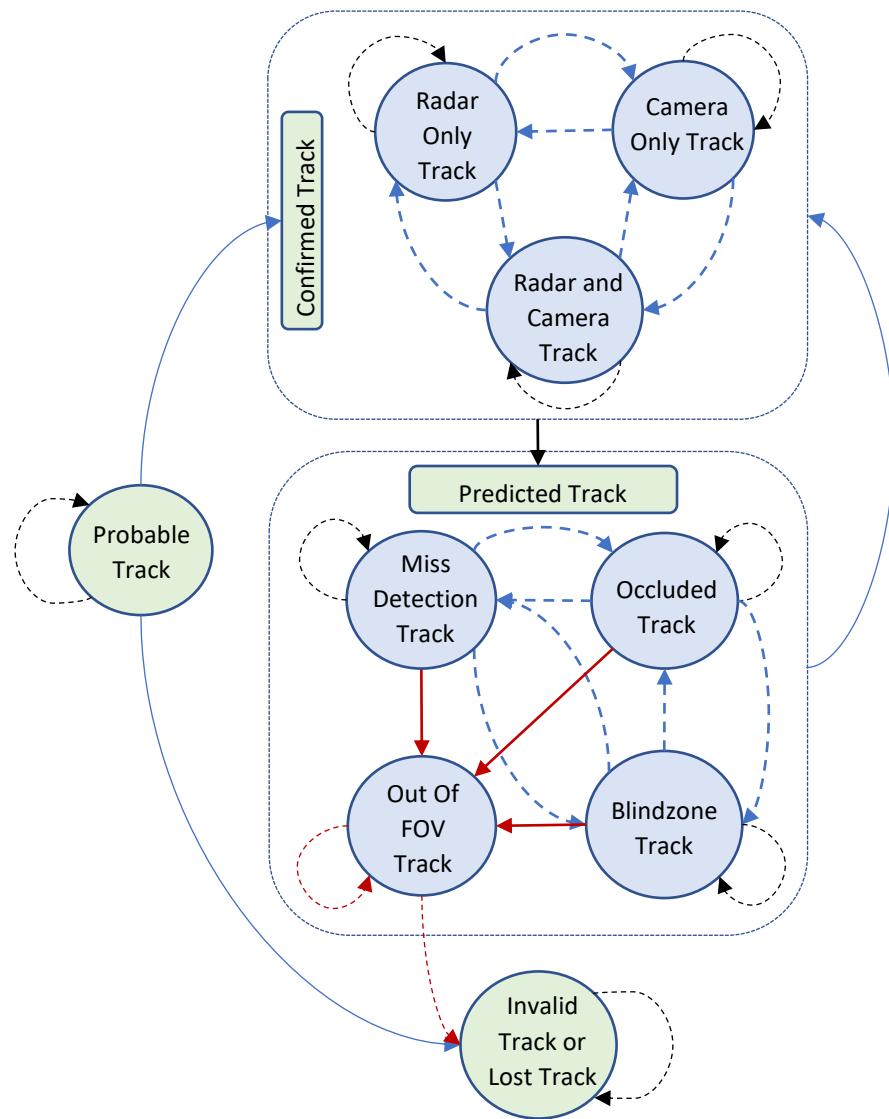


10.2 Deterministic Finite State Automata as Track State Machine

10.2.1 Track States

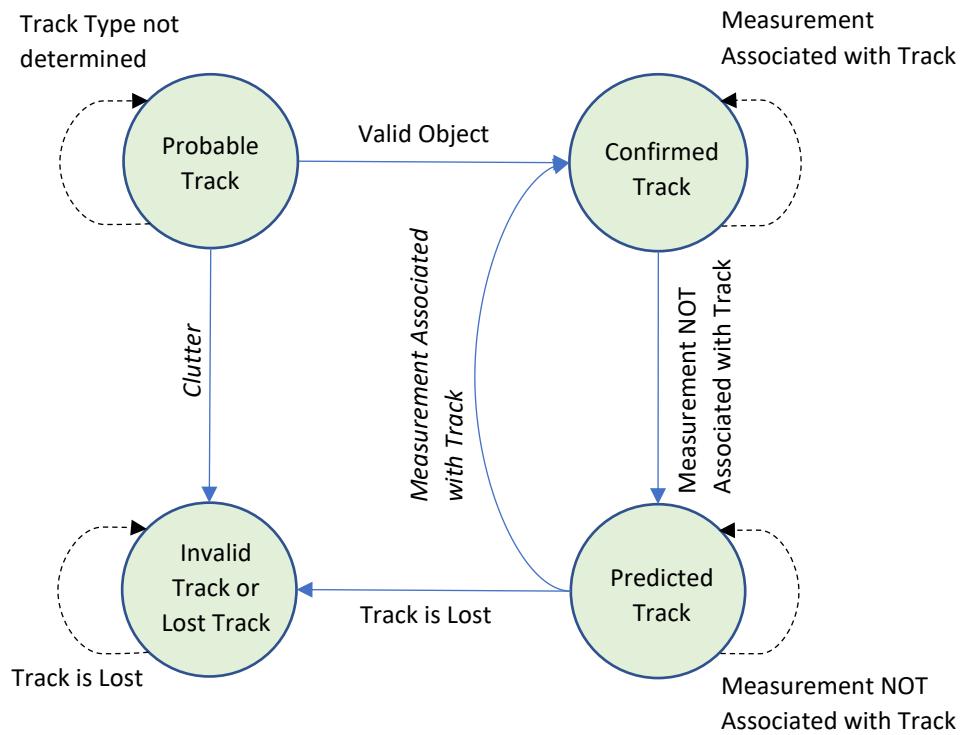
A TRACK can have the following Status (States) :

1. Probable Track : Tentative Track (either Clutter or Object)
2. Confirmed Track : Valid Object Track
 1. **Radar Only Track** : Track update by Radar Measurements
 2. **Camera Only Track** : Track update by Camera Measurements
 3. **Radar and Camera Track** : Track update by Radar and Camera Measurements
3. Predicted Track : Object Tracks NOT updated by any measurements
 1. **Miss-Detection Track** : Tracks not detected (could be due to low snr or sensor noise)
 2. **Occluded Track** : Tracks not detected due to Occlusion
 3. **Blind Zone Track** : Track located in the blind zone, hence not detected
 4. **Out of FoV Track** : Track has moved out of sensor FoV (all sensors)
4. Invalid/Lost Track : Track from Clutter/Object Track not of interest

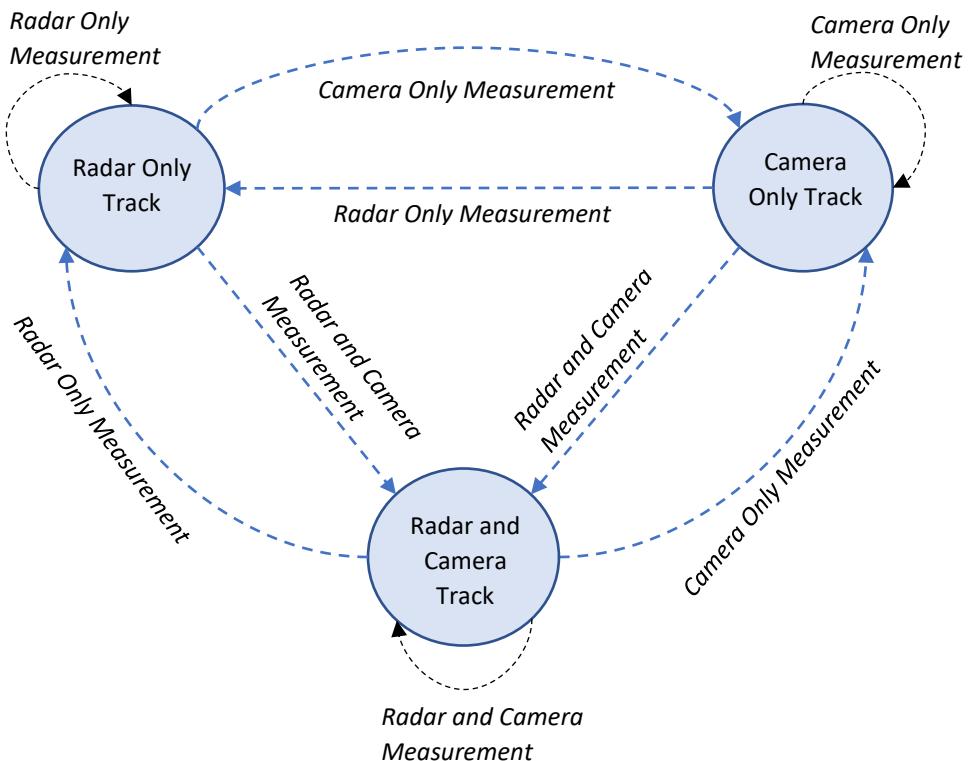


10.2.2 Track State Transitions

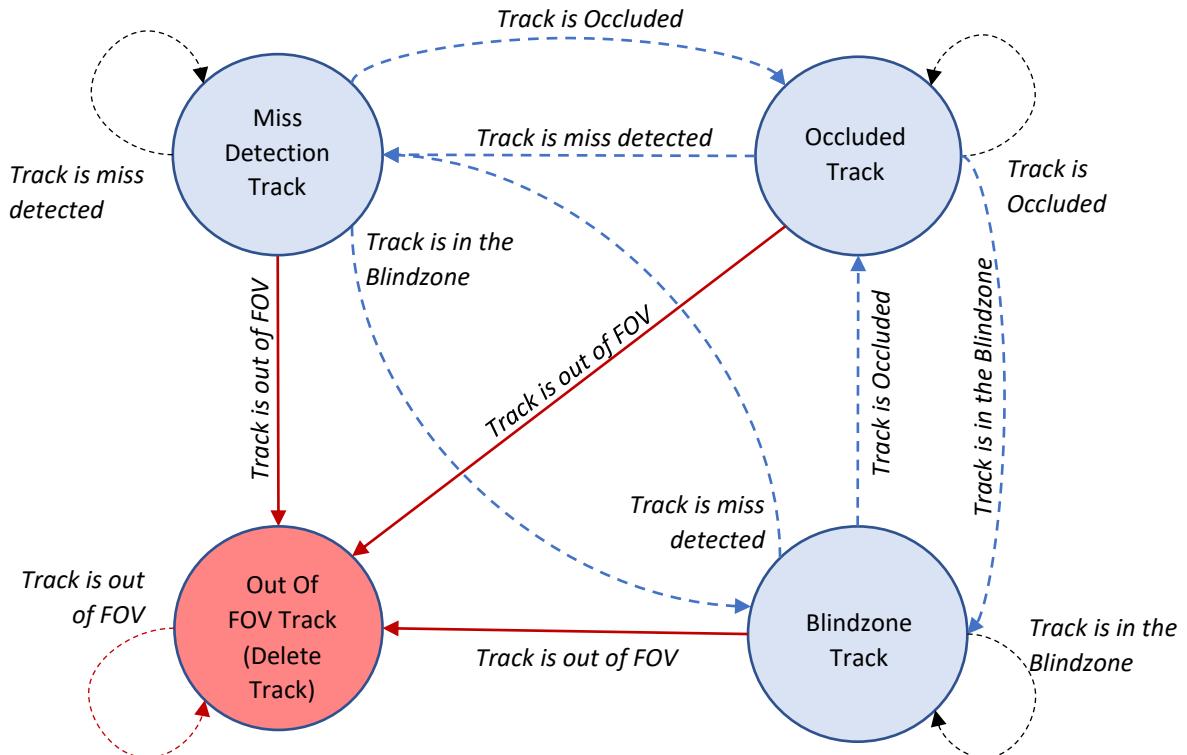
10.2.2.1 Track Manager Main



10.2.2.2 Confirmed Track Manager



10.2.2.3 Predicted Track Manager



10.2.3 Current Implementation

A Simple counter based logic is implemented for determination of confirmed track and lost track. Once adequate simulation data is generated as sensor data, the implementation shall be made more sophisticated as per the proposed design.

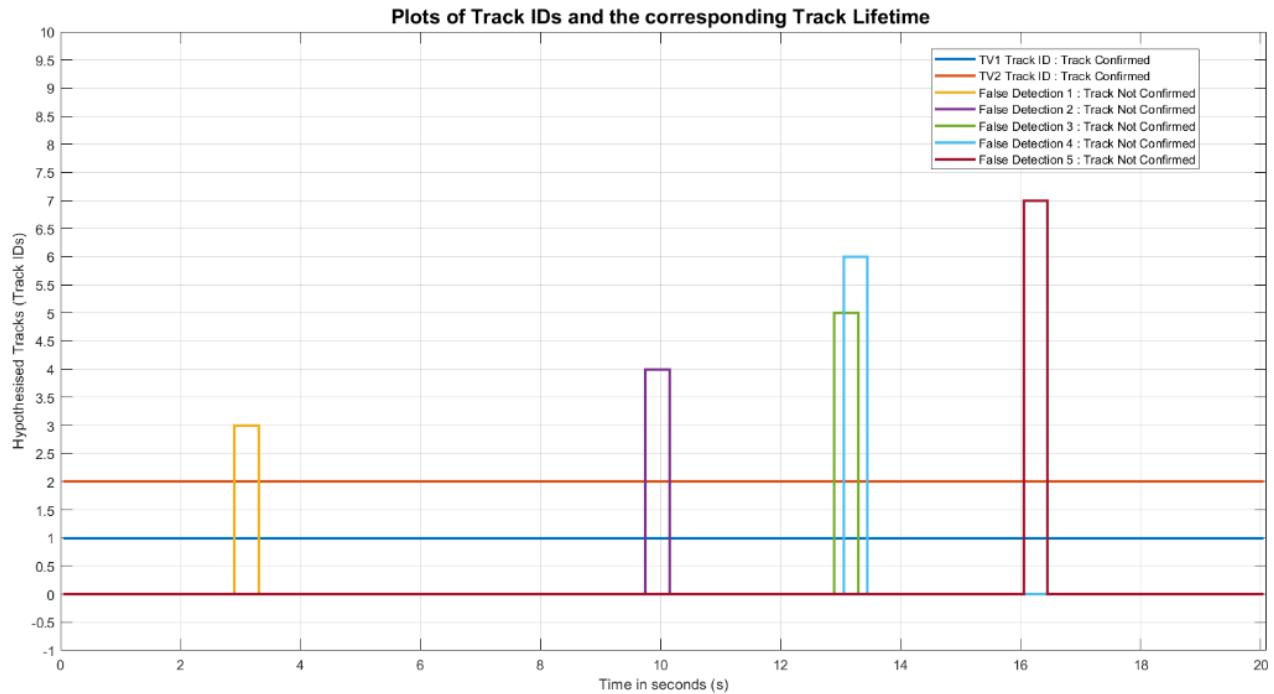
The simplified predicted and gated counter based implementation can be found in the project repository.

Note :

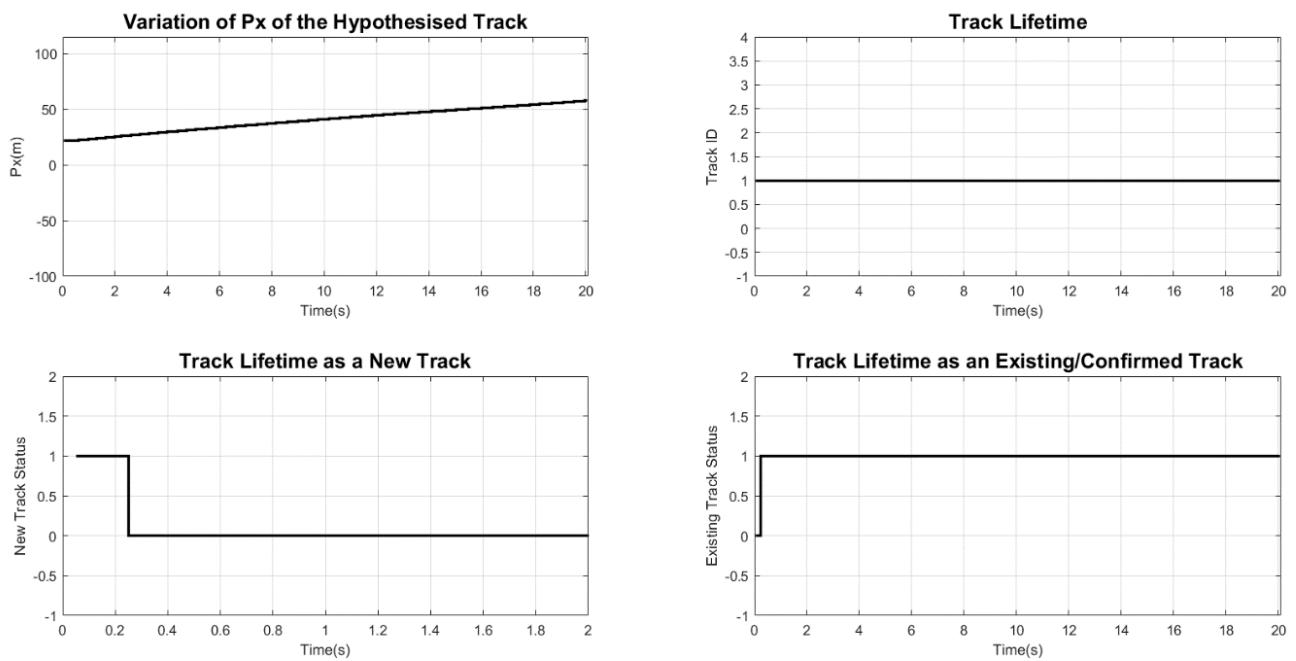
- Radar Track Fusion subsystem and Camera Track Fusion subsystem shall have individual Track Managers which are slightly more simplified than the one which is depicted in this document.
- Due to the complementary nature of Radar and Camera the two individual Track Managers can be merged into a Global Track Manager which is depicted in this document as state machines.

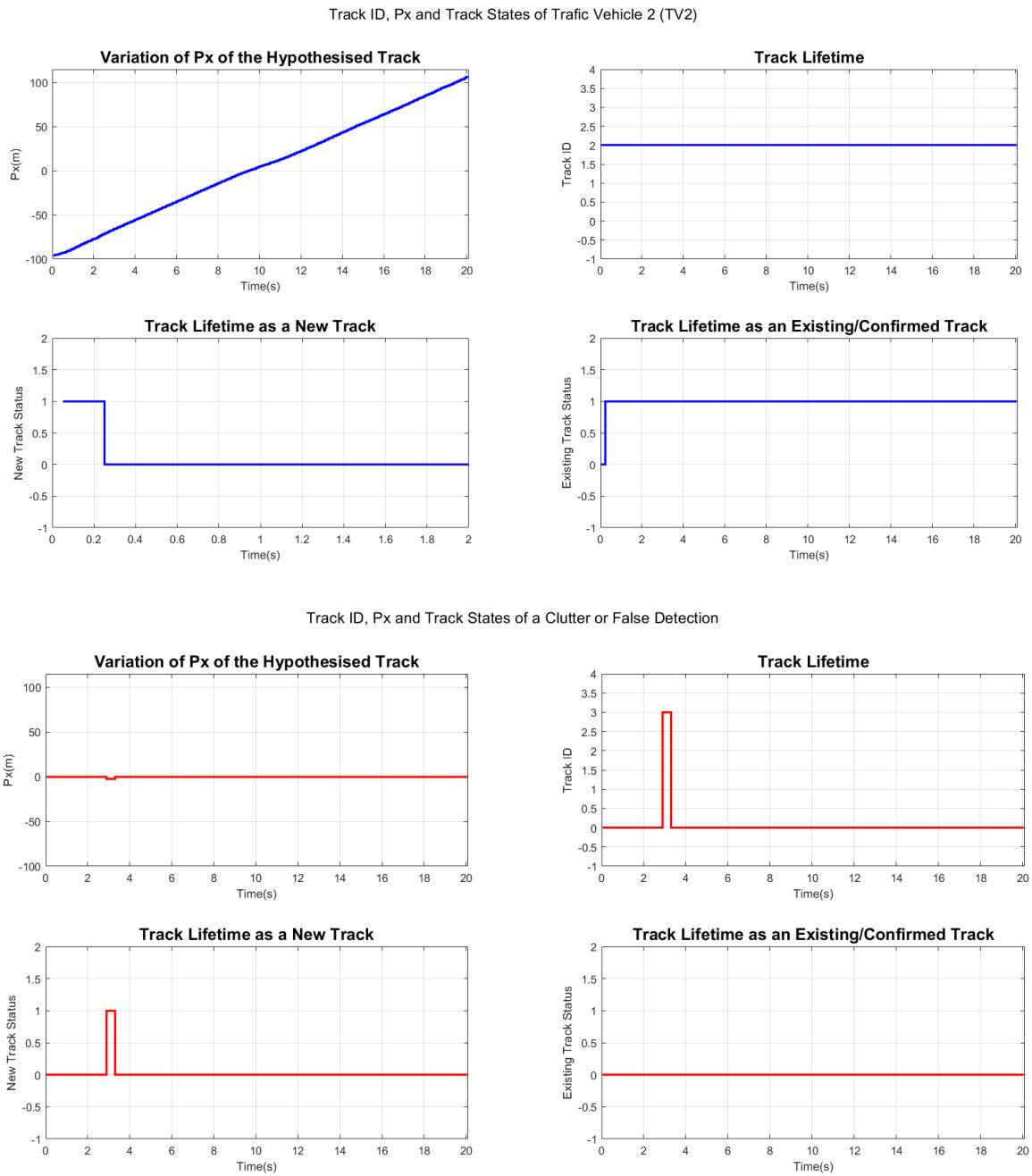
10.3 Plots :

10.3.1 Track ID & Track Status



Track ID, Px and Track States of Trafic Vehicle 1 (TV1)





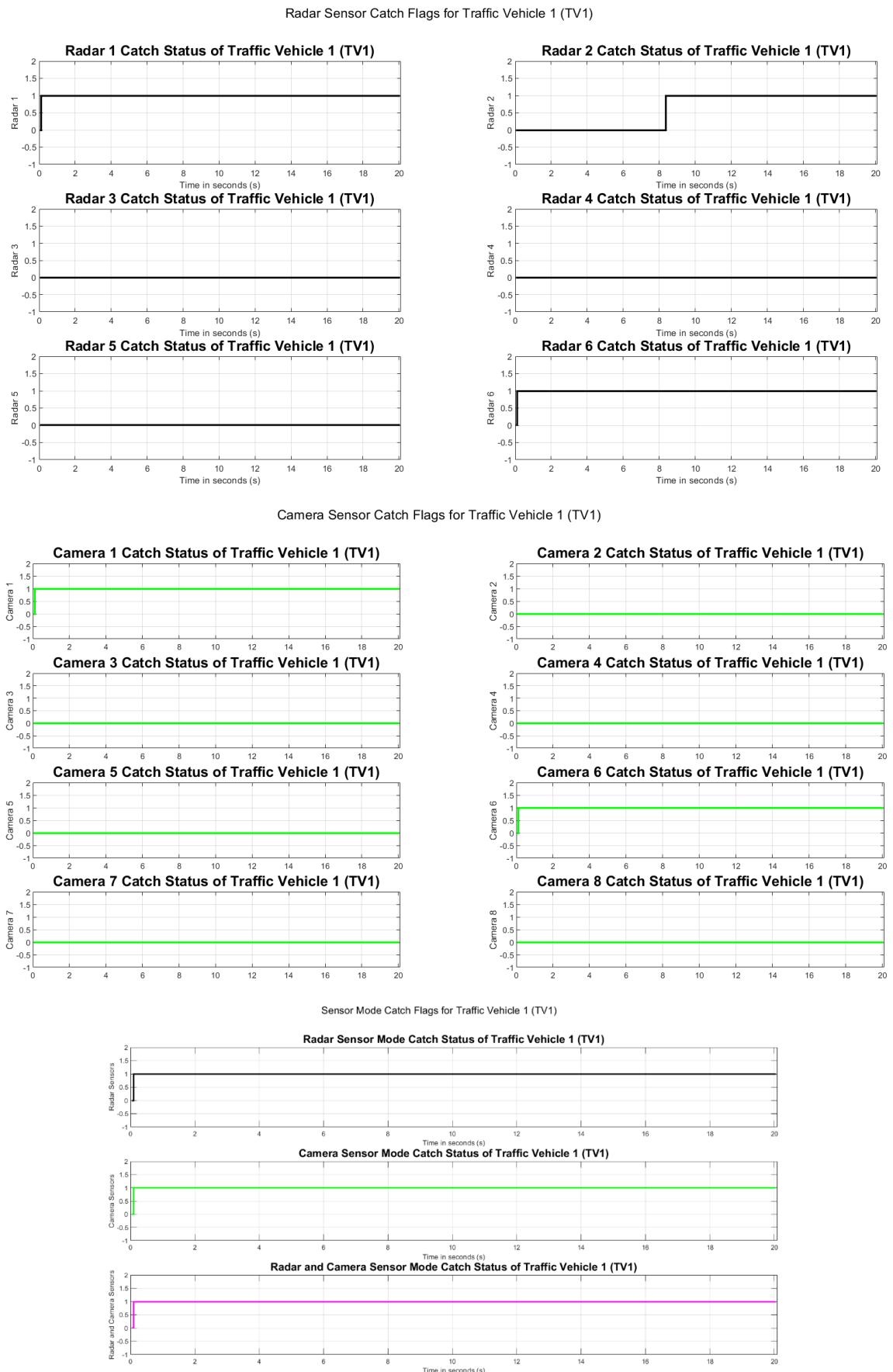
Explanations :

- Total 7 Track IDs are generated
- ID 1 and ID 2 are from valid objects. For both of these tracks, the track status initially is in ‘new’ state and later after confirmation, the status changes to ‘confirmed’ state
- ID 3 to ID 7 are unconfirmed tracks, the tracks are initially is in ‘new’ state, but the tracks are NOT confirmed and eventually are deleted. These tracks are false detections.

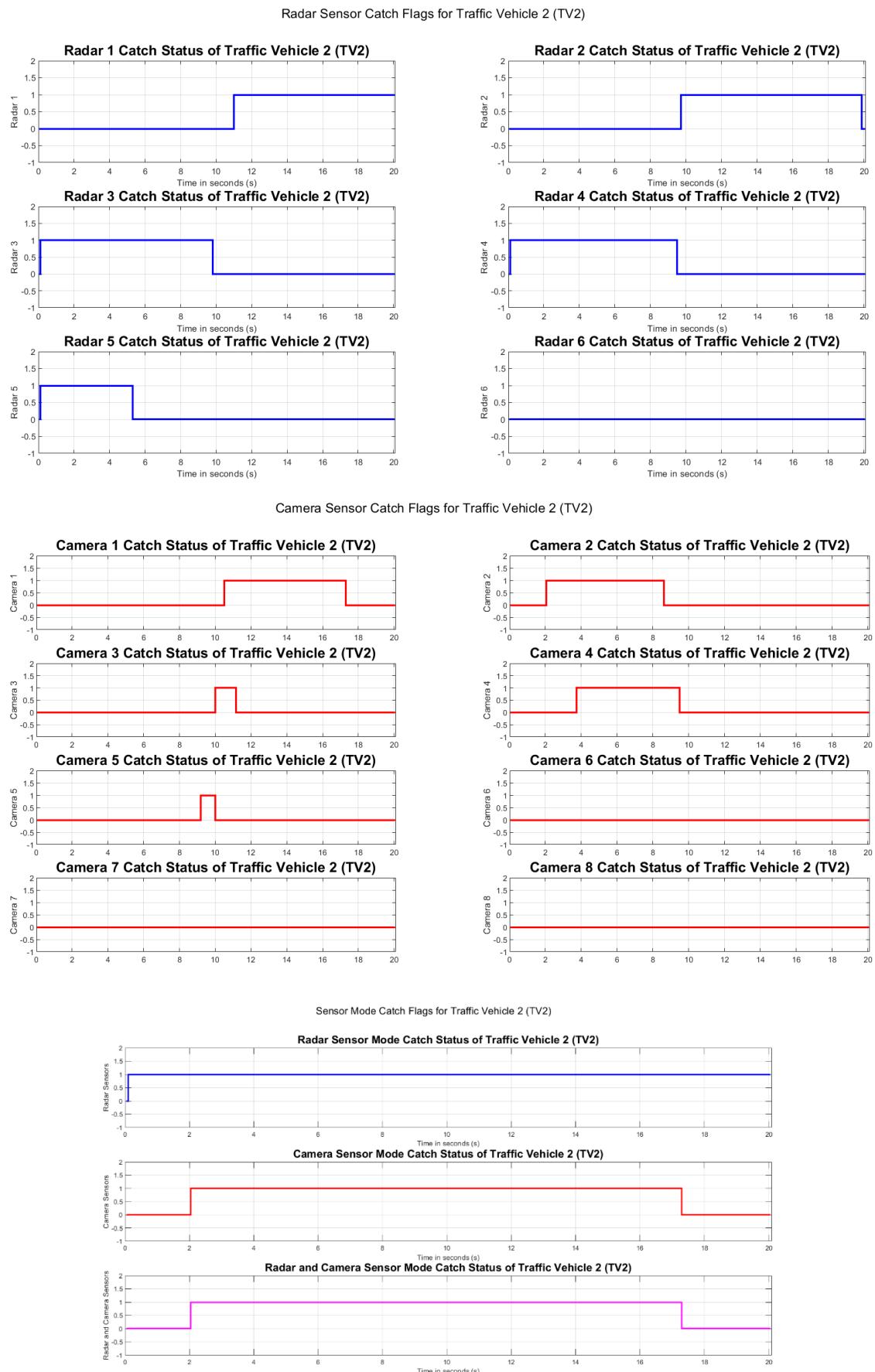
Note :

Track IDs are always positive.

10.3.2 Sensor Catch Status for TV1 : time plot of sensor contributions to the estimation



10.3.3 Sensor Catch Status for TV2 : time plot of sensor contributions to the estimation



11. CONCLUSION

This project in its current version is aimed at designing a robust and scalable fusion architecture. Hence complex fusion techniques like Multi Hypothesis Tracking (MHT), Random Finite Set (RFS) based methods and particle filter based estimations are not considered. In the subsequent versions more state of the art techniques shall be explored and integrated as a fusion library into the proposed fusion framework. Hope the reader finds this document useful, informative and engaging.

Project repository :

- https://github.com/UditBhaskar19/OBJECT_TRACKING_MULTI_SENSOR_FUSION.git

12. References

Textbooks :

- “Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software”,
Authors : Yaakov Bar-Shalom, X.-Rong Li, Thiagalingam Kirubarajan
- “Fundamentals of Object Tracking”, Authors : Subhash Challa, Mark R. Morelande, Darko Mušicki,
Robin J. Evans
- “Statistical Sensor Fusion”, Author : Fredrik Gustafsson
- “Statistical Multi-source multi target information fusion”, Author : Ronald Mahler
- “Advances in Statistical Multisource-Multitarget Information Fusion”, Author : Ronald Mahler

Journals/Papers :

- Thia Kirubarajan and Yaakov bar-shalom “Probabilistic Data Association Techniques for Target Tracking in Clutter”, IEEE · April 2004
- Yaakov bar-shalom and Fred Daum, “The probabilistic data association filter”, IEEE control systems, January 2010.
- Klaudius Werber, Matthias Rapp, Jens Klappstein, Markus Hahn. “Automotive radar gridmap representations”, Conference: Microwaves for Intelligent Mobility (ICMIM), 2015
- Stephan Matzka and Richard Altendorfer, “A Comparison of Track-to-Track Fusion Algorithms for Automotive Sensor Fusion”, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008
- Yaakov bar-shalom, Kuochu C Chang, Rajat K. Saha, “On optimal track-to-track fusion”, IEEE Transactions on Aerospace and Electronic Systems · November 1997
- Martin Ester, Hans-Peter Kriegel, Jiří Sander, Xiaowei Xu “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, KDD-96 Proceedings 1996
- Dominik Kellner, Jens Klappstein, Klaus Dietmayer “Grid-based DBSCAN for clustering extended objects in radar data”, Conference: Intelligent Vehicles Symposium (IV), June 2012 IEEE