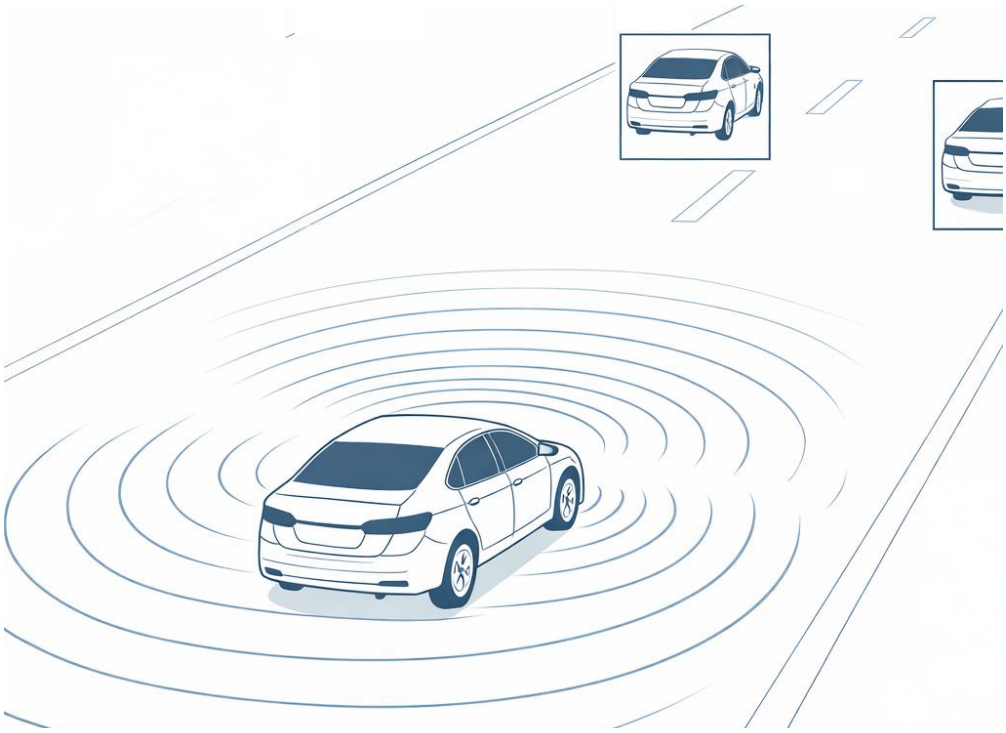
CHALMERS
UNIVERSITY OF TECHNOLOGY



Evaluating Motion Model Hypotheses for Automotive Radar Tracking

A comparative study of motion model hypotheses and automated parameter tuning in EKF-based multi-object tracking using Interacting Multiple Model filtering

Master's thesis in Systems, Control & Mechatronics

Noel Andersson, Hannes Lundberg

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Evaluating Motion Model Hypotheses for Automotive Radar Tracking

A comparative study of motion model hypotheses and automated parameter tuning in EKF-based multi-object tracking using Interacting Multiple Model filtering

NOEL ANDERSSON
HANNES LUNDBERG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Signal Processing
Chalmers University of Technology
Gothenburg, Sweden 2026

Evaluating Motion Model Hypotheses for Automotive Radar Tracking
A comparative study of motion model hypotheses and automated parameter tuning in EKF-based multi-object tracking using Interacting Multiple Model filtering
Noel Andersson
Hannes Lundberg

Supervisor: Niclas Carlström, Aptiv
Examiner: Tomas McKelvey

Master's Thesis 2026
Department of Electrical Engineering
Division of Signal Processing
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Generic vehicle tracking multiple targets generated by Microsoft 365 Copilot.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Evaluating Motion Model Hypotheses for Automotive Radar Tracking
A comparative study of motion model hypotheses and automated parameter tuning in EKF-based multi-object tracking using Interacting Multiple Model filtering
Noel Andersson
Hannes Lundberg
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Automotive radar is a core ADAS sensor due to weather robustness and direct Doppler velocity measurements, but radar multi-object tracking is challenged by heterogeneous and maneuvering target dynamics. This thesis evaluates a white-noise jerk model (CCA) and a curvilinear motion model (CTCA), each implemented in an EKF, and assesses whether combining them in an Interacting Multiple Model (IMM) filter improves robustness across scenarios. To enable an unbiased comparison between CCA, CTCA, and IMM, process-noise parameters and IMM transition/interactions are tuned automatically by formulating tracking as a black-box optimization problem. Performance is optimized using the probabilistic GOSPA (P-GOSPA) metric, which penalizes localization error as well as missed and false tracks under multi-Bernoulli set representations. CMA-ES is used to search the nonconvex parameter space without gradients. Evaluation is performed in a controlled MATLAB simulation with a four-corner radar configuration and known ground truth, fusing radar range-rate detections with object-level pseudo-measurements of position and orientation. Results show strong scenario dependence for single-model tracking and indicate that automated tuning is necessary to avoid biased motion-model conclusions, the IMM provides more consistent performance across diverse driving scenarios than either single model.

Keywords: EKF, IMM, MOT, Radar, Automated Parameter Tuning, P-GOSPA, CMA-ES

Acknowledgements

We would like to thank Aptiv for the opportunity and support throughout this work. Also a special thanks to our supervisor Niclas Carlström for the ongoing guidance and support, Erik Kindlund for helping with a robust simulation environment and last but not least our examiner Tomas McKelvey for providing helpful feedback.

Noel Andersson, Hannes Lundberg, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADAS	Advanced Driver Assistance Systems
BB	Bounding Box
CCA	Constant Cartesian Acceleration
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CTCA	Constant Turn Constant Acceleration
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
FMCW	Frequency Modulated Continuous Wave
GT	Ground Truth
IMM	Interacting Multiple Model
MOT	Multi Object Tracking
P-GOSPA	Probabilistic Generalized Optimal Sub-Pattern Assignment
RADAR	Radio Detection and Ranging
SMM-IMM	Single Motion Model Interacting Multiple Model
NIS	Normalized Innovation Squared
NEES	Normalized Estimation Error Squared

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

j, i	Index for motion models
t	Index for continuous time
k	Index for discrete time step
N	Number of radar detections
r	Number of motion models

Sets

f_X	Tracker output Multi Bernoulli set density
f_Y	Ground truth Multi Bernoulli set density

Parameters

T	Tracker sample time
q_1	Tuning parameter 1 used in automated tuning
q_2	Tuning parameter 2 used in automated tuning
π_{ij}	Markov transition probability from model i to model j

Variables

$\mathbf{x}_k$	State vector

$\mathbf{u}_k$	Input vector
$\mathbf{z}_k$	Measurement vector
$\mathbf{F}_k$	State transition matrix
$\mathbf{B}_k$	Input matrix
$\mathbf{H}_k$	Observation matrix
$\mathbf{w}_k$	Process noise
$\mathbf{v}_k$	Measurement noise
$\mathbf{Q}_k$	Process noise covariance matrix
$\mathbf{R}_k$	Measurement noise covariance matrix
$\mathbf{P_k}$	State covariance matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{\Pi}$	Markov transition matrix
x	Object cartesian x position [m]
y	Object cartesian y position [m]
v	Object velocity [m/s]
a	Object acceleration [m/s^2]
ϕ	Object heading [rad]
ω	Object turn-/yaw-rate [rad/s]
$\dot{r}$	Radar detection range-rate measurement [m/s]
σ^2	Variance
μ	Probability of motion model
Λ	Likelihood
c	P-GOSPA cutoff distance
p	P-GOSPA outlier sensitivity
α	P-GOSPA existance mismatch penalty
g	CMA-ES generation/epoch
$\mathbf{m}^{(g)}$	CMA-ES parameter mean vector at generation g
$\mathbf{C}^{(g)}$	CMA-ES parameter covariance matrix at generation g
$\sigma^{(g)}$	CMA-ES step-size at generation g

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

The background presented in this chapter provides the necessary context for the problem addressed in this thesis.

1.1.1 Advanced Driver Assistance Systems

Advanced Driver Assistance Systems (ADAS) include a broad class of safety and comfort enhancing functions designed to support the driver by perceiving the surrounding environment and reacting according to it. Typical ADAS applications include adaptive cruise control, automatic emergency braking, blind spot detection, lane keeping assistance and collision avoidance systems. The reliability of such functions is tightly coupled to the quality of the underlying perception stack which must operate robustly under varying conditions.
Radar sensors play a central role in modern ADAS due to their robustness to adverse weather conditions, long detection range, direct measurement of radial velocity via the Doppler effect and cost of alternative sensors [15]. Automotive radars typically provide range, azimuth angle, and radial velocity measurements enabling the detection and tracking of surrounding objects. Compared to optical sensors such as cameras and LiDARs, radar measurements are generally noisier and less spatially resolved which places greater demands on the downstream signal processing and tracking algorithms.

1.1.2 Multi Object Tracking

Multi Object tracking (MOT) aims to estimate the number of objects present in the environment as well as their individual states over time [4]. Tracked object states typically include position, velocity, acceleration, heading and yaw rate depending on the tracker structure [18]. A complete MOT pipeline usually consists of several interacting components, including data association, track initiation and deletion logic, motion modeling and state estimation.
The Extended Kalman Filter (EKF) is a widely used state estimation technique in MOT due to its computational efficiency and ability to handle nonlinear measurement models [20]. By recursively predicting object states using a motion model and correcting them with incoming sensor measurements, the EKF provides a principled framework for tracking dynamic objects in noisy environments.

However real-world traffic participants rarely adhere to a single, simple motion model. To address this, multiple-model approaches such as the Interacting Multiple Model (IMM) are deployed [6]. IMM filtering allows simultaneous tracking with multiple motion hypotheses and probabilistically blends their estimates based on the observed measurements.

1.1.3 Pursuit of the Perfect Tuning

One of the most persistent challenges in EKF applications or general estimation problems is tuning or hyper parameter selection. In practice tuning is often performed manually based on engineering intuition, incremental adjustment, and visual inspection of results [3]. While such approaches can yield acceptable results they rarely lead to truly optimal parameter settings and are difficult to reproduce.

A common alternative is brute-force tuning, where large numbers of parameter combinations are evaluated to identify configurations that perform well according to a chosen metric [3]. However, even for moderately complex EKF-based MOT systems the dimensionality of the tuning space quickly becomes prohibitive. As a result, exhaustive search strategies become computationally infeasible for realistic systems and scenario sets.

Automating the tuning process using optimization techniques is therefore highly appealing, but it introduces several nontrivial challenges:

- Defining a well behaved KPI metric:
  The objective function must meaningfully reflect overall tracking performance.
- Availability of representative datasets:
  Effective tuning requires scenarios that reflect the diversity of real world driving while providing access to reliable ground truth.
- High-dimensional and nonlinear parameter spaces:
  The relationship between EKF parameters and tracking performance is highly nonlinear and often non-intuitive.

These challenges collectively motivate treating the MOT pipeline as a black-box optimization problem, where parameter configurations are evaluated solely based on their resulting tracking performance.

1.2 Related Work

Radar-based perception and tracking have been extensively studied in the context of ADAS. Due to its robustness under adverse environmental conditions and its ability to directly measure relative velocity, radar has become a key sensing modality for automotive tracking applications [16]. However, radar measurements are inherently affected by noise and uncertainty, arising from factors such as limited angular resolution, multipath propagation, and target-dependent reflection characteristics. As a result, reliable radar-based tracking relies heavily on accurate motion prediction and appropriate filter tuning to compensate for measurement uncertainty.

1.2.1 Evaluating Motion Models

The choice of motion model is a central aspect of object tracking systems, as it directly determines a filter's ability to predict target behavior between measurements. In the literature, a wide range of motion models has been proposed, each based on different assumptions regarding target dynamics.
Li and Jilkov provide a comprehensive survey of dynamic models for target tracking and emphasize that no single motion model can adequately describe all target behaviors [18]. They further study multi-model tracking approaches and conclude that multi-model based techniques provide a robust solution for many maneuvering target tracking problems [19]. A particularly well-established multi-model approach in this context is the IMM framework, which enables probabilistic switching between motion hypotheses based on estimated model likelihoods.
Several studies have performed comparative evaluations of motion models in vehicular tracking contexts. Schubert et al. compare multiple motion models using real vehicle trajectories and demonstrate that tracking performance is highly scenario-dependent, with different models exhibiting varying strengths depending on the underlying motion characteristics [21]. These findings underscore the importance of motion model selection in practical object tracking systems.
While such works highlight the impact of motion model choice, comparative evaluations are often limited by practical considerations. In many cases, studies rely on theoretical analyses or single simulation-based scenarios that may not fully capture the diversity of motion encountered in automotive environments. Additionally, the influence of filter parameter tuning, particularly process noise selection, is frequently treated heuristically and often not evaluated at all, which can complicate direct comparison between motion models. These factors motivate further investigation into motion model behavior under consistent modeling assumptions and representative automotive tracking scenarios. Since the assumed process noise directly governs how strongly a motion model can deviate from its nominal dynamics, motion model selection and filter tuning are inherently coupled in practical tracking systems.

1.2.2 Filter Parameter Tuning

Several approaches have been proposed to automate the selection of filter parameters using data-driven or optimization-based techniques. In particular, black-box optimization methods have been applied to tune parameters that strongly influence estimation performance, such as process noise covariances. Covariance Matrix Adaptation Evolution Strategies (CMA-ES) provide a principled framework for such optimization tasks by iteratively adapting a search distribution based on observed performance feedback [9].
More recently, optimization-based approaches have been proposed to automate Kalman filter parameter tuning. Chen et al. formulate the tuning problem as a stochastic black-box optimization task and apply Bayesian optimization to tune process noise parameters using consistency-based cost functions derived from Normalized Innovation Squared (NIS) and Normalized Estimation Error Squared (NEES) statistics [7]. Their approach avoids reliance on gradient information and demonstrates robustness to local minima and stochastic variations in the objective function.

In the literature, tuning objectives are commonly defined using statistical consistency metrics such as NIS or NEES, which assess the agreement between predicted uncertainty and observed estimation residuals. While such methods provide a principled framework for filter tuning, they may incur significant computational cost due to repeated filter evaluations or Monte Carlo simulations, particularly as the dimensionality of the tuning problem increases. Consequently, existing work highlights important trade-offs between robustness, computational complexity, and practical applicability in automated Kalman filter tuning.

1.3 Purpose

The purpose of this thesis is to investigate and compare different motion models commonly used within EKF based MOT systems, with a particular focus on radar-based automotive applications. By analyzing the behavior of these motion models across a diverse set of representative driving scenarios, the work aims to identify their respective strengths and limitations with respect to different types of target motion.
Further focus is put on on how these complementary motion characteristics can be leveraged through an IMM filter. The IMM framework enables the simultaneous use of multiple motion hypotheses and adaptively blends their estimates based on measurement consistency. The objective is to assess whether combining motion models in an IMM structure leads to more robust and consistent tracking performance across scenarios compared to relying on a single motion model.
This can be summarized by the main research question: How do different motion models used in EKF-based multi-object tracking for automotive radar perform across varying driving scenarios, and does combining these models within an IMM framework yield more robust and consistent tracking performance than relying on a single motion model?

1.4 Scope

The scope is limited to the evaluation of motion models within an Extended Kalman Filter–based multi-object tracking framework using artificially generated simulation data with known ground truth of both tracked objects and host motion. All experiments are conducted in a controlled environment to enable objective and reproducible performance comparison between models.
In addition to traditional automotive radar detections, this work utilizes radar derived pseudo-measurements produced by an upstream perception module, providing object-level position, orientation and size estimates that are fused within the tracking framework.
Only front-steering vehicles of a single vehicle type are considered in the tracking scenarios. Further, the thesis focuses on high level tracking performance and filter behavior and thus low level radar signal processing, detection generation, and data association mechanisms are not subject to analysis or optimization.

2
Theory

2.1 Automotive Radar Fundamentals

Automotive radar systems are an essential part of today's ADAS systems. Being both robust and cheap it creates a good balance between cost effectiveness and performance [17]. These systems leverage radio waves to measure range, range-rate, azimuth and elevation angle in the vehicles surroundings. This section will introduce the basic fundamentals of radar systems used in the automotive industry will be introduced.

2.1.1 Frequency Modulated Continuous Wave Radar

Modern automotive radar systems predominantly use Frequency Modulated Continuous Wave (FMCW) radar due to its ability to simultaneously measure target range and radial velocity with high accuracy while maintaining low peak transmit power. In FMCW radar, the transmitted signal is a continuous electromagnetic wave whose carrier frequency is modulated over time, typically using linear frequency sweeps referred to as chirps [16].
During operation, the FMCW radar continuously transmits the modulated waveform while simultaneously receiving reflections from objects in the environment. The received signal experiences a propagation delay depending on target distance and, for moving targets, an additional Doppler-induced frequency shift. By mixing the received signal with a replica of the transmitted signal, a low-frequency beat signal is generated. The spectral content of this beat signal encodes information related to the time delay and relative motion between the radar and the target [5].
A key property of FMCW radar is that range-related and motion-related information are embedded in the frequency domain of the received signal rather than in explicit time-of-flight measurements. By transmitting and processing multiple chirps, these contributions can be separated in subsequent signal processing stages. This efficient encoding of information, together with reduced hardware complexity compared to pulsed radar systems, makes FMCW radar particularly well suited for automotive applications operating in the millimeter-wave frequency bands [16].

2.1.2 Radar Measurements and Uncertainty

Automotive radar systems typically provide measurements of target range, radial velocity, and angular direction. These quantities are obtained through a combination

of radio-frequency hardware, antenna geometry, and digital signal processing, and each measurement type exhibits distinct accuracy and uncertainty characteristics.
The target range R is determined from the round-trip propagation time of the transmitted electromagnetic wave and is given by

$$R = \frac{c\Delta T}{2}, \tag{2.1}$$

where c is the speed of light and ΔT denotes the measured time delay between transmission and reception [5]. In practice, the time delay is not measured directly but inferred from frequency differences introduced by the modulation waveform, which leads to range uncertainty that depends primarily on the signal bandwidth.
Radial velocity is estimated using the Doppler effect, which introduces a frequency shift proportional to the relative motion between the radar and the target. For a monostatic radar, the Doppler frequency f_D is related to the target radial velocity v_r by

$$f_D = \frac{2v_r}{\lambda}, \tag{2.2}$$

where λ is the wavelength of the transmitted signal. Due to coherent signal processing and long integration times, automotive radar systems typically achieve high accuracy in radial velocity estimation compared to position measurements [16].
Angular information is obtained by exploiting phase differences between signals received at spatially separated antenna elements. By employing an antenna array in a multiple-input multiple-output (MIMO) configuration, the angle of arrival of reflected signals can be estimated using beam forming or spectral estimation techniques [5]. The angular resolution and accuracy are fundamentally limited by the effective antenna aperture and the operating wavelength, and angle measurements generally exhibit higher uncertainty than range and velocity measurements.
Overall, radar measurement uncertainty arises from thermal noise, finite bandwidth, limited antenna aperture, and environmental effects such as multipath propagation. These uncertainties are commonly modeled as zero-mean random errors in subsequent tracking and state-estimation algorithms.

2.2 Kalman Filters

Kalman filtering is a robust recursive filtering method originally introduced by Kalman in 1960 [10]. The filter provides an optimal linear estimator, in the minimum mean square error sense, for systems described by linear stochastic difference equations with Gaussian noise. The Kalman filter operates by alternating between a prediction and a correction step, producing a sequence of state estimates and corresponding error covariance matrices. While the original work by Kalman was presented using continuous-time formulations, the equivalent discrete-time matrix form used in this thesis follows the standard presentation found in modern state estimation literature [20].
The discrete-time linear system is described by the state and measurement equations

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, \tag{2.3}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{2.4}$$

where $\mathbf{x}_k$ is the system state, $\mathbf{u}_k$ the input vector, $\mathbf{z}_k$ the measurement vector, $\mathbf{F}_k$ the state transition matrix, $\mathbf{B}_k$ the input matrix and $\mathbf{H}_k$ is the observation matrix. The process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$ are assumed to be zero-mean Gaussian with covariances

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{2.5}$$
$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{2.6}$$

and are mutually uncorrelated [10].

2.2.1 Prediction

In the prediction step, the state estimate and its covariance are propagated forward in time using the system model. The state estimate is given by

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k, \tag{2.7}$$

while the corresponding error covariance matrix is

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k. \tag{2.8}$$

These equations describe the uncertainty growth due to model dynamics and process noise and constitute the time update of the Kalman filter.

2.2.2 Correction

When a new measurement becomes available, the predicted estimate is updated using the innovation, defined as

$$\widetilde{y}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}. \tag{2.9}$$

The innovation covariance is

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k. \tag{2.10}$$

The Kalman gain is computed as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}. \tag{2.11}$$

The Kalman gain balances the confidence between prediction and measurement. Larger measurement uncertainty results in a smaller gain, giving more weight to the prediction, while smaller measurement noise causes the estimate to closely follow the measurement. Using the Kalman gain, the updated state estimate and error covariance are given by

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \widetilde{y}_k, \tag{2.12}$$
$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \tag{2.13}$$

These equations complete the recursive filtering cycle.

2.2.3 Extended Kalman Filter

The standard Kalman filter formulation assumes linear system and measurement models. For nonlinear systems,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \tag{2.14}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \tag{2.15}$$

the EKF may be applied. The EKF approximates the nonlinear functions by first-order Taylor expansions around the current state estimate. However, since it relies on first-order Taylor expansions its performance may degrade for strongly nonlinear systems.
The state transition and measurement Jacobians are defined as

$$\mathbf{F}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}}, \tag{2.16}$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}}. \tag{2.17}$$

These Jacobians are used to linearize the system locally around the current estimate, enabling the propagation of the state mean and covariance using the standard Kalman filter prediction and update equations. In particular, the nonlinear state transition function is applied directly to propagate the state estimate, while the covariance is propagated using the linearized model. Similarly, the innovation is computed using the nonlinear measurement function, but the Kalman gain is obtained using the linearized measurement model. This results in a recursive estimation algorithm applicable to moderately nonlinear systems [22].

2.3 Motion Models

In object tracking the motion model describes the dynamics of the tracked object where different models are suitable for objects depending on the typical dynamics of the object and what information is of interest.
Some common motion models are Constant Velocity, Constant Acceleration, Constant Turn and Constant Turn Constant Velocity that each are more or less suitable for different objects. For instance a front steering vehicle can be considered to have a curved trajectory based on speed and heading whilst a pedestrian which can make sudden direction changes and have no clear heading could be considered to be modelled by a constant acceleration model.
The general model can be described as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{\Gamma}\mathbf{w}(t), \tag{2.18}$$

where $\mathbf{f}(\mathbf{x}(t))$ describes the model dynamics, $\mathbf{w}(t)$ is a continuous white noise process and $\mathbf{\Gamma}$ is a matrix that maps the noise process dimension to the state dimension.

The choice of motion model is not only determined by the target dynamics but also by the characteristics of the sensing system. In particular, the model complexity must be balanced against the measurement rate and sensor accuracy. For high update rates and low measurement noise, the motion between consecutive measurements is limited, and simpler models may be sufficient. In contrast, for lower update rates or more nonlinear motion, a more expressive model is typically required to maintain accurate state estimates [2].

2.3.1 White Noise Jerk Model

The White Noise Jerk Model [18] referred to as Constant Cartesian Acceleration model (CCA) in this report describes the objects dynamics by position, velocity and acceleration in x and y direction. The state vector is defined as

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ v_x(t) \\ a_x(t) \\ y(t) \\ v_y(t) \\ a_y(t) \end{bmatrix}, \tag{2.19}$$

where the system dynamics are described as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{\Gamma}\mathbf{w}(t), \quad \mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} v_x(t) \\ a_x(t) \\ 0 \\ v_y(t) \\ a_y(t) \\ 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.20}$$

where Γ maps the process noise dimension to the state dimension. Changes in lateral and longitudinal acceleration approximately zero, with disturbances captured by the process noise $\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$ where

$$\mathbf{Q}(t) = \begin{bmatrix} \sigma_{\dot{a}_x}^2 & 0 \\ 0 & \sigma_{\dot{a}_y}^2 \end{bmatrix}. \tag{2.21}$$

2.3.2 Curvilinear Motion Model

The Curvilinear Motion model is a family of motion models that describes a curved trajectory using kinematic relations between position, speed, heading, accelerations and turn-/yaw-rate [18]. The Constant Turn Constant Acceleration (CTCA) model (also refered to as CTRA [23]) assumes tangential acceleration $a \neq 0$ where the state vector is defined as

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ a(t) \\ \phi(t) \\ \omega(t) \end{bmatrix}, \tag{2.22}$$

and system dynamics are described as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{\Gamma}\mathbf{w}(t), \quad \mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} v(t)\cos\phi(t) \\ v(t)\sin\phi(t) \\ a(t) \\ 0 \\ \omega(t) \\ 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.23}$$

where Γ maps the process noise dimension to the state dimension. Changes in acceleration and yaw rate are approximately zero, with disturbances captured by the process noise $\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$ where

$$\mathbf{Q}(t) = \begin{bmatrix} \sigma_{\dot{a}}^2 & 0 \\ 0 & \sigma_{\dot{\omega}}^2 \end{bmatrix}. \tag{2.24}$$

2.4 Interacting Multiple Model

A more advanced approach to predicting target behaviour is the usage of adaptive motion models. A normal single motion model is reliable for specific cases and often break down when more advanced dynamics and movements are introduced. An adaptive motion model often depend on multiple models to include a larger amount of dynamics [1]. One adaptive motion model structure in particular is the IMM.

The IMM is a multi-modal Bayesian estimator that runs several motion models and their corresponding filters in parallel [6]. Typically, each model is associated with its own variant of a Kalman-based estimator, such as a standard Kalman Filter, EKF, or Unscented Kalman Filter (UKF) among others [11]. By operating several models simultaneously, the IMM can represent different hypotheses about the target's motion, such as constant-velocity, acceleration, turn and evaluate them probabilistically as new measurements arrive.

One key aspect of the IMM is that it does not simply choose the "best" model at each time step, it interacts the different models by a structured mixing process. The state estimates and covariances from each model are probabilistically mixed based on likelihood of model transitions. These transitions are decided by a Markov chain that defines how likely the target is to switch from one motion to another [6]. After this

mixing, each model performs its own prediction update step independently. Finally the IMM combines the updated model estimates into a single overall estimate by weighting the separate updated models according to their mode probabilities. Figure 2.1 show a description of the workflow. Where $\mathbf{x}$ denotes the state vector, $\mathbf{P}$ denotes the Covariance matrix, z denotes the new measurement information, k denotes the current time step, μ denotes the model probability and Λ denotes the likelihood from each filter update.

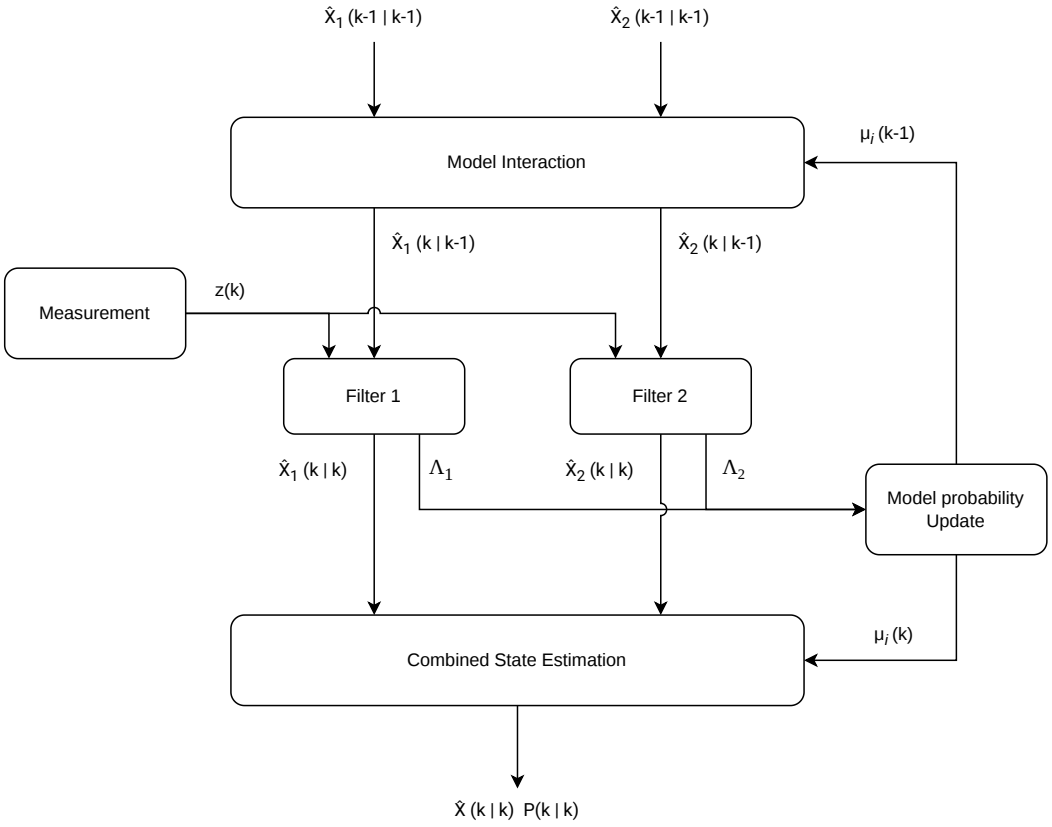


Figure 2.1: A workflow picturing the process of a two model IMM filter.

2.4.1 Model Interaction

The model interaction or mixing step initializes each model at time step k with prior knowledge of a probabilistic blend of the previous model estimates at time $k-1$. This realizes the possibility that the motion has switched between samples.
At time $k-1$ the IMM maintains for each motion model a state estimate $\hat{\mathbf{x}}_i(k-1)$, a covariance $\mathbf{P}_i(k-1)$ and a model probability $\mu_i(k-1)$ for $i = 1, 2, ..., r$ where r equals the number of models used. The model transitions are represented by a Markov transition matrix $\mathbf{\Pi} = [\pi_{ij}]$, where π represent the possibility of switching from model i to j. In this matrix a larger diagonal equals a slower switching process. To initialize each model the IMM computes mixing probabilities that weigh how

much each of the filters contributes to the initial estimate of the models. These probabilities are defined as

$$c_j(k-1) = \sum_{\ell=1}^{r} \pi_{\ell j}\, \mu_\ell(k-1), \qquad j = 1, \ldots, r, \tag{2.25}$$

and

$$\mu_{i\to j}(k-1) = \frac{\pi_{ij}\, \mu_i(k-1)}{c_j(k-1)}, \qquad \sum_{i=1}^{r} \mu_{i\to j}(k-1) = 1. \tag{2.26}$$

Using these weights, the initial mixed state for model j is formed as

$$\hat{\mathbf{x}}_j^0(k-1) = \sum_{i=1}^{r} \mu_{i\to j}(k-1)\, \hat{\mathbf{x}}_i(k-1), \tag{2.27}$$

and the corresponding mixed initial covariance becomes,

$$\mathbf{P}_j^0(k-1) = \sum_{i=1}^{r} \mu_{i\to j}(k-1) \left[\mathbf{P}_i(k-1) + \left(\hat{\mathbf{x}}_i(k-1) - \hat{\mathbf{x}}_j^0(k-1)\right)\left(\hat{\mathbf{x}}_i(k-1) - \hat{\mathbf{x}}_j^0(k-1)\right)^\top\right]. \tag{2.28}$$

If the motion models use different state parameterizations, the state estimate and covariance from model i must first be mapped into the state space of model j before mixing. This transformation is performed through a model-to-model state mapping $T_{i\to j}$ and its associated Jacobian $\mathbf{J}_{i\to j}$. The transformed state and covariance are computed as

$$\hat{\mathbf{x}}_{i\to j}(k-1) = \mathbf{T}_{i\to j}(\hat{\mathbf{x}}_i(k-1))\,, \qquad \mathbf{P}_{i\to j}(k-1) \approx \mathbf{J}_{i\to j}\, \mathbf{P}_i(k-1)\, \mathbf{J}_{i\to j}^\top. \tag{2.29}$$

After this transition into the common state space and the mixing of previous information, the combined state space is split into each corresponding model before continuing the filtering process.

2.4.2 Model Conditioned Filtering

For this part the different models each run an independent Bayesian filter. As described earlier this will most often be a Kalman variant [11]. Here a traditional Kalman cycle is performed as described in Section 2.2. The IMM specific part of this cycle can be found in the measurement likelihood, which gives information on how well each model explains incoming measurement data. This likelihood forms the basis of which the mode probabilities are updated. The likelihood of the measurement $\mathbf{z}(k)$ under model j is computed as

$$\Lambda_j(k) = \frac{1}{\sqrt{(2\pi)^m\, |\mathbf{S}_j(k)|}} \exp\!\left(-\frac{1}{2}\widetilde{y}_k^\top \mathbf{S}_j^{-1}(k)\widetilde{y}_k]\right), \tag{2.30}$$

where $\widetilde{y}_k$ is the innovation for model j, $\mathbf{S}_j(k)$ is the corresponding innovation covariance and m denotes the measurement dimension [11]. A higher likelihood indicates that the measurement is more consistent with the motion hypothesis predicted by the motion model.

2.4.3 Mode Probability Update

Following the model conditioned filtering the mode probability is calculated using the likelihoods gotten from the previous step. The mode probability reflect how well each motion model and their respective filter process explains the new measurement [6]. The update begins by computing the prior mode probability $\bar{\mu}$ for each model M_j by propagating the previous probability $\mu(k-1)$ through the Markov transition matrix mentioned earlier,

$$\bar{\mu}_j(k) = \sum_{i=1}^{r} \pi_{ij}\, \mu_i(k-1), \qquad j = 1, \ldots, r, \qquad \text{with } \sum_{j=1}^{r} \bar{\mu}_j(k) = 1. \tag{2.31}$$

The prior explains how likely it is before a new measurement is introduced that the target follows model j.

Then, given the measurement likelihoods $\Lambda_j(k)$ from Equation (2.30), the posterior mode probabilities are updated via Bayes' rule,

$$\mu_j(k) = \frac{\Lambda_j(k)\, \bar{\mu}_j(k)}{\sum\limits_{\ell=1}^{r} \Lambda_\ell(k)\, \bar{\mu}_\ell(k)}, \qquad \sum_{j=1}^{r} \mu_j(k) = 1. \tag{2.32}$$

The result of this is a normalized percentage of which model is more likely to fit with the targets movement. So if a model j has a large prior and a high likelihood then it will get a higher percentage.

2.4.4 Estimation Combination

In the final step of the IMM filter a single, combined state estimate and covariance is produced by mixing the updated model conditioned estimates with a probability weighting. Given the posterior model-conditioned estimates $\{\hat{\mathbf{x}}_j(k), \mathbf{P}_j(k)\}_{j=1}^{r}$ and the posterior mode probabilities $\{\mu_j(k)\}_{j=1}^{r}$, the IMM combined estimate is

$$\hat{\mathbf{x}}(k) = \sum_{j=1}^{r} \mu_j(k)\, \hat{\mathbf{x}}_j(k), \tag{2.33}$$

$$\mathbf{P}(k) = \sum_{j=1}^{r} \mu_j(k) \Big[\mathbf{P}_j(k) + \big(\hat{\mathbf{x}}_j(k) - \hat{\mathbf{x}}(k)\big)\big(\hat{\mathbf{x}}_j(k) - \hat{\mathbf{x}}(k)\big)^{\top} \Big]. \tag{2.34}$$

These expressions yield the final IMM output at time k. The mean is a convex combination of the model-conditioned estimates, while the covariance includes both each model's posterior uncertainty and a between-means term that preserves uncertainty due to disagreement across models.

2.5 P-GOSPA

Probabilistic Generalized Optimal Sub-Pattern Assignment (P-GOSPA) is a Key Performance Indicator (KPI) used in MOT where both the ground truth and the

filter output can be represented as multi-Bernoulli (MB) set densities, each parameterized by an existence probability $r \in (0, 1]$ and a single-object probability density $p(\cdot)$ [24].

Classical GOSPA is a metric defined on deterministic finite sets of object states and penalizes localization error, missed detections, and false detections but ignores posterior uncertainty. P-GOSPA generalizes GOSPA from deterministic sets to MB densities, so evaluation directly accounts for both existence uncertainty and state-estimate uncertainty.

Let f_X and f_Y be MB densities with n_X and n_Y Bernoulli components $\{(r_i^x, p_i^x)\}$ and $\{(r_j^y, p_j^y)\}$. For a chosen base metric $d(p_x, p_y)$ between single-object densities and its cut-off $d^{(c)} = \min(d, c)$, the P-GOSPA distance $d_p^{(c,\alpha)}(f_X, f_Y)$ is

$$[\min_{\pi \in \Pi_{n_Y}} (\sum_{i=1}^{n_X} \left[\min(r_x^i, r_y^{\pi(i)}) d^{(c)}(p_x^i, p_y^{\pi(i)})^p + |r_x^i - r_y^{\pi(i)}| \frac{c^p}{\alpha} \right] + \frac{c^p}{\alpha} \sum_{i=n_X+1}^{n_Y} r_y^{\pi(i)})]^{\frac{1}{p}}. \quad (2.35)$$

The cut-off $c > 0$ saturates large localization errors, the exponent $p \in [1, \infty)$ controls outlier sensitivity, and $\alpha \in (0, 2]$ weighs localization errors against association errors. Setting $\alpha = 2$ is recommended for MOT because it makes "one miss + one false" cost the same (before the $1/p$ root) [24].

2.6 Covariance Matrix Adaptation Evolution Strategy

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [9] is a stochastic, derivative-free optimization algorithm designed for continuous, non-linear, and non-convex black-box optimization problems.
CMA-ES models the search process as the iterative adaptation of a multivariate normal distribution. At generation g, candidate solutions $\mathbf{x}_k^{(g)}$ are sampled as

$$\mathbf{x}_k^{(g)} = \mathbf{m}^{(g)} + \sigma^{(g)} \mathbf{C}^{(g)^{1/2}} \mathbf{z}_k, \quad \mathbf{z}_k \sim \mathcal{N}(0, \mathbf{I}) \quad (2.36)$$

where $\mathbf{m}^{(g)}$ is the mean, $\sigma^{(g)}$ the global step size and $\mathbf{C}^{(g)}$ the covariance matrix encoding variable dependencies.
After evaluating the objective function, solutions are ranked, and the best μ individuals are used to update the sampling distribution. CMA-ES relies exclusively on ranking rather than absolute objective values, which makes it invariant to strictly monotonic transformations of the objective function.

Covariance Matrix Adaptation The core idea of CMA-ES is to learn the second order structure of the objective function by adapting the covariance matrix. The update combines two mechanisms:

- Rank-μ update, which incorporates information from several selected steps

- Rank-one update, based on an evolution path that accumulates successful search directions across generations.

Together, these updates allow the algorithm to approximate the inverse Hessian on convex quadratic problems, enabling efficient search on ill-conditioned and non-separable landscapes.

Step-Size Control In addition to shape adaptation, CMA-ES uses cumulative step-size adaptation to adjust the global search scale. This mechanism compares the length of the evolution path to its expected length under random selection and increases or decreases the step-size accordingly. Cumulative step-size adaptation allows the algorithm to automatically transition between exploration and exploitation without external parameter tuning.

Properties and Applicability CMA-ES exhibits several desirable invariance properties, including invariance to rotations, translations, and scaling of the search space. These properties contribute to its robust performance across a wide class of optimization problems. The algorithm is particularly effective for dimensions ranging roughly from 5 to 100, though computational complexity grows quadratically with dimension due to covariance matrix operations.
Because CMA-ES does not require gradients and handles noise, discontinuities, and multimodality well, it has been successfully applied to engineering design, control, hyperparameter optimization, and simulation-based optimization tasks.

3
Methods

3.1 MATLAB Simulations

To verify the fundamental behaviour of the developed algorithms and motion models, high-quality input data is essential for both functional validation and performance evaluation. When using real automotive radar data, numerous sources of uncertainty, such as multipath effects, false alarms, and inaccuracies in ego-motion estimation, can introduce large discrepancies in the evaluation of tracking algorithms.

To address this, well-controlled simulations with known target trajectories and ego-vehicle motion are used as an initial evaluation step. Such simulations enable repeatable experiments and allow systematic testing of edge cases, which is crucial for assessing the theoretical limits of the motion models.

3.1.1 Simulation Assumptions and Limitations

To clearly define the scope and limitations of the MATLAB simulation environment, the following assumptions and simplifications are imposed:

- Detection noise is strictly Gaussian
- No false alarms, missed detections, or multipath effects
- Ideal ego-motion information without noise or latency
- Clutter-free environment

These simplifications ensure that any observed performance limitations primarily originate from the tracking algorithms and motion models themselves, rather than from sensor or perception uncertainties.

3.1.2 Simulation Setup

The simulation will only be used to gather data and transform it into binaries that can later be used by the algorithm written in C++. The simulation is built upon several different MATLAB add ons with the main ones being Driving Scenario Designer [12] and Radar Data Generator [13]. This system consist of a four corner-radar setup, this gives a 360 degree radar perception coverage.

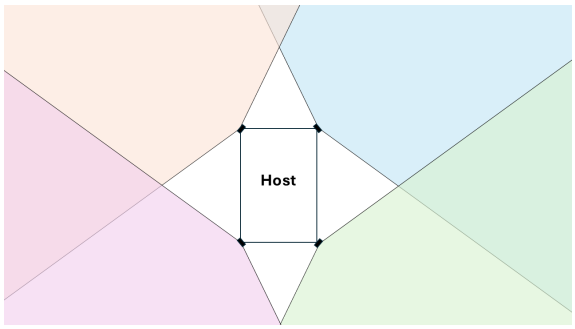


Figure 3.1: Four corner radar setup.

Figure 3.1 shows an illustration of the radar configuration used in the simulated scenarios. Each individual radar is configured and parameterized according to a standard automotive radar configuration. The common attributes of the radars are presented in Table 3.1, while their individual mounting positions and angels are presented in Table 3.2.

These configuration parameters define the sensing capabilities of the simulated radar system, including detection coverage, angular resolution, maximum measurable range, and Doppler performance. Accurately modelling these attributes ensures that the simulated detections resemble those produced by a real automotive radar system, both in terms of geometry and detection behaviour. This is crucial when evaluating motion models, as the measurement characteristics directly influence the filter performance.

Table 3.1: Common radar attributes used in the four-radar simulation setup.

Attribute	**Value**
Center frequency	79 GHz
Update rate	20 Hz
Field of view (azimuth × elevation)	150° × 24°
Range limits	[0, 250] m
Range-rate limits	[−150, 150] m/s
Max unambiguous range	250 m
Max unambiguous radial speed	150 m/s
Azimuth resolution	0.5°
Elevation resolution	4°
Range resolution	0.5 m
Range-rate (Doppler) resolution	0.1 m/s
Detections include range-rate	Yes
Detections include elevation	Yes
Noise	Enabled
Ghosts	Disabled
False alarms	Disabled
Occlusion	Enabled
Range ambiguities	Enabled
Range-rate ambiguities	Enabled
Reference range	5 m
Reference RCS	−30 dBsm
Detection probability at reference	0.9
Target report format	Detections
Detection coordinates	Sensor spherical
Actor profiles source	`actorProfiles(scenario)`
Max number of reports (per update)	2000

Table 3.2: Per-sensor mounting locations and orientations relative center of rear axle.

Sensor	**Mounting location (x, y, z) [m]**	**Mounting angles (yaw, pitch, roll) [°]**
Rear-Left (ID = 1)	(−1.0, 0.9, 0.5)	(130, 0, 0)
Rear-Right (ID = 2)	(−1.0, −0.9, 0.5)	(−130, 0, 0)
Front-Right (ID = 3)	(3.729, −0.9, 0.5)	(−50, 0, 0)
Front-Left (ID = 4)	(3.729, 0.9, 0.5)	(50, 0, 0)

Table 3.3: Measurement noise definitions.

Measurement	σ	μ	Unit
Range-Rates	0.1	0	m/s
Bounding Box Position	0.15	0	m
Bounding Box Orientation	0.05	0	rad
Azimuth	0.01	0	rad

The noise for each measurement can be found in Table 3.3.

3.2 Motion Model Implementation

The performance of different motion models are investigated individually by Extended Kalman Filter implementation before any interacting model filter method are evaluated.
The reference frame all motion models are derived from can be seen in Figure 3.2

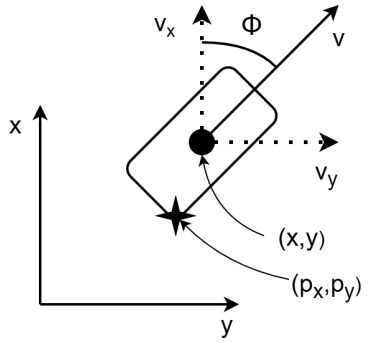


Figure 3.2: Motion models reference frame.

where $a(t) = \dot{v}(t)$, $\omega(t) = \dot{\phi}(t)$, object positions are denoted as (x, y) and radar detection position as (p_x, p_y).
The general discrete time model is defined as

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k, \quad \mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{3.1}$$

where it can be noted that the input components $\mathbf{u}_k$ from the original discrete time model (2.14) is unmodeled due to lack of information to describe such dynamics.

3.2.1 Use of Measurements

All motion models evaluated in this thesis fuse two complementary radar based measurement sources, traditional point-level radar detections providing range-rate information, and object level pseudo-measurements originating from a radar perception module that supplies position, orientation and size estimates.

Traditional Radar Detections A single range-rate measurement from a radar detection can then be expressed according to

$$\dot{r} = [\cos\theta \quad \sin\theta](\begin{bmatrix} v_x \\ v_y \end{bmatrix} + \omega \begin{bmatrix} -(p_y - y) \\ (p_x - x) \end{bmatrix}), \tag{3.2}$$

where (p_x, p_y) is the position of the detection and θ the azimuth angle of the detection [14]. The nonlinear term induced by the object yaw-rate can be ignored in low yaw-rate motion and captured by measurement uncertainties, however in high yaw rate scenarios the nonlinearity of the measured object motion can have large affects on the resulting measured radial velocity.
The tracked object velocities v_x, v_y can be described as the tangential velocity v of the object with the help of the heading ϕ of the object.

$$v_x = v\cos\phi, \quad v_y = v\sin\phi. \tag{3.3}$$

Given equations (3.2), (3.3) and N number of radar detections the systems of equations can be defined

$$\begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \vdots \\ \dot{r}_N \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ \cos\theta_2 & \sin\theta_2 \\ \vdots & \vdots \\ \cos\theta_N & \sin\theta_N \end{bmatrix} \begin{bmatrix} v\cos\phi \\ v\sin\phi \end{bmatrix} + \omega \begin{bmatrix} -\cos\theta_1\,(p_{y,1} - y) + \sin\theta_1\,(p_{x,1} - x) \\ -\cos\theta_2\,(p_{y,2} - y) + \sin\theta_2\,(p_{x,2} - x) \\ \vdots \\ -\cos\theta_N\,(p_{y,N} - y) + \sin\theta_N\,(p_{x,N} - x) \end{bmatrix}. \tag{3.4}$$

The unknowns of (3.4) are x, y, v, ϕ, ω which means given $N > 5$ we have an overdetermined system that can be solved using numerous linear regression methods.
Motion models that does not include heading or yaw-rate in the state vector cannot be updated with measurements according to equation (3.4), therefore the radar detecions must be reduced according to equation (3.5) where loss of information is unavoidable when yaw-rate is nonzero.

$$\begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \vdots \\ \dot{r}_N \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ \cos\theta_2 & \sin\theta_2 \\ \vdots & \vdots \\ \cos\theta_N & \sin\theta_N \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \tag{3.5}$$

Instantaneous Bounding Box Measurements In addition to the radar detections, object-level pseudo measurements are available in the form of instantaneous bounding box (BB) estimates. Each bounding box provides an estimate of object center position, width, length, and orientation. These pseudo measurements originate from a higher-level radar perception stage and are treated as measurements within the tracking framework.
Compared to traditional radar detections, bounding box measurements provide strong constraints on object position and orientation. They are therefore used primarily to update the positional states and heading-related quantities of the tracked objects, while velocity estimation is dominated by the radar Doppler information.

It is important to note that the object heading ϕ is defined in this work as the direction of the velocity vector, i.e.,

$$\phi = \arctan\left(\frac{v_y}{v_x}\right),$$

which is not necessarily aligned with the measured bounding box orientation. In scenarios involving significant lateral slip or aggressive maneuvers, the orientation of the bounding box may differ from the instantaneous velocity direction. This discrepancy is treated as measurement uncertainty and is accounted for through the measurement noise modeling.

Measurement uncertainty The measurement variances modeled in $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ for radar detection range rates ($\dot{r}$) along with bounding box position (x, y) and heading ϕ in measurement update Kalman step is set equally accross all motion models according to the gaussian noise added to the simulated matlab data according to table 3.3. This results in measurement variance tuning according to

$$R_{\dot{r}} = 1.0 \text{ m}^2/\text{s}^2, \quad R_{x,y} = 1.5 \text{ m}^2, \quad R_{\phi} = 0.1 \text{ rad}^2. \tag{3.6}$$

3.2.2 CCA

Prediction From the continuous state vector (2.19) and model dynamics (2.20) the discrete motion model according to (3.1) is defined as

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{x}_k = \begin{bmatrix} x_k \\ v_{x,k} \\ a_{x,k} \\ y_k \\ v_{y,k} \\ a_{y,k} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

where T is the sample time. The discrete time process noise is defined as $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ where the discrete time covariance matrix results in

$$\mathbf{Q}_k = \begin{bmatrix} \frac{T^5}{20}\sigma_{\dot{a}_x}^2 & \frac{T^4}{8}\sigma_{\dot{a}_x}^2 & \frac{T^3}{6}\sigma_{\dot{a}_x}^2 & 0 & 0 & 0 \\ \frac{T^4}{8}\sigma_{\dot{a}_x}^2 & \frac{T^3}{3}\sigma_{\dot{a}_x}^2 & \frac{T^2}{2}\sigma_{\dot{a}_x}^2 & 0 & 0 & 0 \\ \frac{T^3}{6}\sigma_{\dot{a}_x}^2 & \frac{T^2}{2}\sigma_{\dot{a}_x}^2 & T\sigma_{\dot{a}_x}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{T^5}{20}\sigma_{\dot{a}_y}^2 & \frac{T^4}{8}\sigma_{\dot{a}_y}^2 & \frac{T^3}{6}\sigma_{\dot{a}_y}^2 \\ 0 & 0 & 0 & \frac{T^4}{8}\sigma_{\dot{a}_y}^2 & \frac{T^3}{3}\sigma_{\dot{a}_y}^2 & \frac{T^2}{2}\sigma_{\dot{a}_y}^2 \\ 0 & 0 & 0 & \frac{T^3}{6}\sigma_{\dot{a}_y}^2 & \frac{T^2}{2}\sigma_{\dot{a}_y}^2 & T\sigma_{\dot{a}_y}^2 \end{bmatrix}, \tag{3.8}$$

according to [18], where $\sigma_{\dot{a}_x}^2$ and $\sigma_{\dot{a}_y}^2$ representing the noise power spectral densities of jerk in lateral and longitudinal direction in object point of view and are tuneable parameters. This is done due to the typical front wheel steering vehicle having different jerk dynamics in longitudinal and lateral direction.

Measurement Update The radar detections are utilized according to equation (3.5) where the nonlinear yawing object motion is ignored, this yields the range rate measurement $z_k = \dot{r}$ for a single detection and the observation matrix can be defined as

$$\mathbf{H} = \frac{\partial h(x_{k|k-1})}{\partial \mathbf{x}} = \begin{bmatrix} 0 & \cos\theta & 0 & 0 & \sin\theta & 0 \end{bmatrix}. \tag{3.9}$$

The bounding box position measurements z_k and observation matrix $\mathbf{H}$ follows as

$$\mathbf{z}_k = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.10}$$

3.2.3 CTCA

Prediction Deriving of the discrete time model (3.1) from the continuous model (2.23) is done according to [23]. The discrete state vector is defined as

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & v_k & a_k & \phi_k & \omega_k \end{bmatrix}^T, \tag{3.11}$$

where where the discrete model dynamics is defined as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k + \mathbf{w}_k, \tag{3.12}$$

where

$$\Delta\mathbf{x}_k = \begin{bmatrix} \frac{1}{\omega_k^2}(a_k \cos\phi_{k+1} - a_k \cos\phi_k + v_{k+1}\omega_k \sin\phi_{k+1} - \omega_k v_k \sin\phi_k) \\ \frac{1}{\omega_k^2}(a_k \sin\phi_{k+1} - a_k \sin\phi_k - v_{k+1}\omega_k \cos\phi_{k+1} + v_k\omega_k \cos\phi_k) \\ a_k T \\ 0 \\ \omega_k T \\ 0 \end{bmatrix}. \tag{3.13}$$

Note Δx_k and Δy_k that can explode in magnitude as $\omega_k \to 0$. Therefore a guard is implemented where the expressions for Δx_k and Δy_k are replaced with equations (3.14) and (3.15) calculated by limits [23], when the yaw-rate $|\omega|$ is less or equal to some limit L rad/s.

$$\Delta x_k = (v_k T + \frac{a_k T^2}{2}) \cos\phi_k, \quad \omega_k \to 0 \tag{3.14}$$

$$\Delta y_k = (v_k T + \frac{a_k T^2}{2}) \sin\phi_k, \quad \omega_k \to 0 \tag{3.15}$$

Comparing these with the discrete CCA motion model (3.7) it can be seen that the CTCA model has been reduced to a unidirectional CCA model when yaw-rate $\omega \approx 0$.
The discrete time process noise is defined as $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ where the discrete time covariance matrix results in (3.16) according to [23]. Here

$$\mathbf{Q}_k = \begin{bmatrix} \frac{T^5}{20}(\sigma_a^2\cos^2\phi + \sigma_\omega^2 v^2\sin^2\phi) & \frac{T^5}{20}\sin\phi\cos\phi\,(\sigma_a^2 - \sigma_\omega^2 v^2) & \frac{T^4}{8}\sigma_a^2\cos\phi & \frac{T^3}{6}\sigma_a^2\cos\phi & -\frac{T^4}{8}\sigma_\omega^2 v\sin\phi & -\frac{T^3}{6}\sigma_\omega^2 v\sin\phi \\ \frac{T^5}{20}\sin\phi\cos\phi\,(\sigma_a^2 - \sigma_\omega^2 v^2) & \frac{T^5}{20}(\sigma_a^2\sin^2\phi + \sigma_\omega^2 v^2\cos^2\phi) & \frac{T^4}{8}\sigma_a^2\sin\phi & \frac{T^3}{6}\sigma_a^2\sin\phi & \frac{T^4}{8}\sigma_\omega^2 v\cos\phi & \frac{T^3}{6}\sigma_\omega^2 v\cos\phi \\ \frac{T^4}{8}\sigma_a^2\cos\phi & \frac{T^4}{8}\sigma_a^2\sin\phi & \frac{T^3}{3}\sigma_a^2 & \frac{T^2}{2}\sigma_a^2 & 0 & 0 \\ \frac{T^3}{6}\sigma_a^2\cos\phi & \frac{T^3}{6}\sigma_a^2\sin\phi & \frac{T^2}{2}\sigma_a^2 & T\,\sigma_a^2 & 0 & 0 \\ -\frac{T^4}{8}\sigma_\omega^2 v\sin\phi & \frac{T^4}{8}\sigma_\omega^2 v\cos\phi & 0 & 0 & \frac{T^3}{3}\sigma_\omega^2 & \frac{T^2}{2}\sigma_\omega^2 \\ -\frac{T^3}{6}\sigma_\omega^2 v\sin\phi & \frac{T^3}{6}\sigma_\omega^2 v\cos\phi & 0 & 0 & \frac{T^2}{2}\sigma_\omega^2 & T\,\sigma_\omega^2 \end{bmatrix}, \tag{3.16}$$

where σ_a^2 and σ_ω^2 represents the noise power spectral densities of jerk and yaw-acceleration and are tuneable parameters.

Measurement Update For the measurement update kalman step the innovation for a single radar detection is

$$\widetilde{y}_k = z_k - h(\mathbf{x}_{k|k-1}), \tag{3.17}$$

where a single range rate measurement is set as $z_k = \dot{r}$ and $h(\mathbf{x}_{k|k-1})$ can be defined as

$$h(\mathbf{x}_{k|k-1}) = v\cos\phi\cos\theta + v\sin\phi\sin\theta - \omega(p_y - y)\cos\theta + \omega(p_x - x)\sin\theta, \tag{3.18}$$

according to equation (3.4).
The linearized observation matrix $\mathbf{H}$ for a single measurement is defined as

$$\mathbf{H} = \frac{\partial h(\mathbf{x}_{k|k-1})}{\partial \mathbf{x}} = \begin{bmatrix} -\omega\sin\theta \\ \omega\cos\theta \\ \cos\phi\cos\theta + \sin\phi\sin\theta \\ 0 \\ v(\cos\phi\sin\theta - \sin\phi\cos\theta) \\ \sin\theta(p_x - x) - \cos\theta((p_y - y)) \end{bmatrix}^T. \tag{3.19}$$

The bounding box position and heading measurements are applied according to the following measurement and observation matrix

$$\mathbf{z}_k = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.20}$$

3.3 Interacting Multiple Model Implementation

The IMM framework requires several practical steps to enable multiple motion models to operate within a single tracking architecture. While theoretical formulations presented in Section 2.4 defines how mixed estimates, model probabilities and combined outputs are computed, the actual implementation must address additional challenges such as differences in state parameterizations, numerical stability during calculations and the construction of a consistent common state representation.

3.3.1 Model Mixing

In the implemented IMM framework, the interaction step is performed using a native-space formulation rather than by mapping all model states into a single common computational state vector. Since the CCA and CTCA models use different state parameterizations, the mixed initial condition for each model is instead computed in that model's own native state space. This means that the CCA branch is mixed in Cartesian coordinates, while the CTCA branch is mixed in curvilinear coordinates. Cross-model contributions are incorporated by projecting the state estimate and covariance of the other model into the receiving model's state space through nonlinear coordinate transformations and their associated Jacobians. In this way, the IMM recursion preserves the natural representation of each motion model while still allowing statistical interaction between them.

3.3.1.1 State-Space Transformations

To transfer information between the two model branches, explicit transformations are defined between the CTCA and CCA state vectors. The CTCA model state is represented in curvilinear form as

$$\mathbf{x}_{\mathrm{CTCA}} = \begin{bmatrix} x_k \\ y_k \\ v_k \\ a_k \\ \phi_k \\ \omega_k \end{bmatrix}, \tag{3.21}$$

while the CCA model state is represented in Cartesian form as

$$\mathbf{x}_{\mathrm{CCA}} = \begin{bmatrix} x_k \\ v_{x,k} \\ a_{x,k} \\ y_k \\ v_{y,k} \\ a_{y,k} \end{bmatrix}. \tag{3.22}$$

The transformation from CTCA to CCA is defined as

$$T_{\mathrm{CTCA}\to\mathrm{CCA}} : \begin{bmatrix} x_k \\ y_k \\ v_k \\ a_k \\ \phi_k \\ \omega_k \end{bmatrix} \mapsto \begin{bmatrix} x_k \\ v_k \cos\phi_k \\ a_k \cos\phi_k - v_k\omega_k \sin\phi_k \\ y_k \\ v_k \sin\phi_k \\ a_k \sin\phi_k + v_k\omega_k \cos\phi_k \end{bmatrix}. \tag{3.23}$$

The reverse transformation from CCA to CTCA is implemented as

$$T_{\text{CCA}\to\text{CTCA}} : \begin{bmatrix} x_k \\ v_{x,k} \\ a_{x,k} \\ y_k \\ v_{y,k} \\ a_{y,k} \end{bmatrix} \mapsto \begin{bmatrix} x_k \\ y_k \\ \sqrt{v_{x,k}^2 + v_{y,k}^2} \\ \frac{v_{x,k}a_{x,k}+v_{y,k}a_{y,k}}{\sqrt{v_{x,k}^2+v_{y,k}^2}} \\ \text{atan2}(v_{y,k}, v_{x,k}) \\ 0 \end{bmatrix}. \tag{3.24}$$

This reverse mapping is inherently approximate, since the CCA model does not explicitly estimate yaw angle or yaw rate. In the implementation, the heading is reconstructed from the velocity direction when the speed is sufficiently large, while the yaw rate is set to zero. However, during mixing of the CTCA branch, the heading and yaw-rate states are not taken from the projected CCA state. Instead, they are retained directly from the CTCA model, so that only the shared kinematic states (x, y, v, a) are blended across models.

Jacobian-Based Covariance Projection The Jacobians were obtained by first-order linearization of the nonlinear state transformations, that is, by evaluating the partial derivatives of each transformed state component with respect to the original state vector. For the projection from CTCA to CCA, the covariance is propagated using the Jacobian

$$\mathbf{J}_{\text{CTCA}\to\text{CCA}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\phi_k & 0 & -v_k\sin\phi_k & 0 \\ 0 & 0 & -\omega_k\sin\phi_k & \cos\phi_k & -a_k\sin\phi_k - v_k\omega_k\cos\phi_k & -v_k\sin\phi_k \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin\phi_k & 0 & v_k\cos\phi_k & 0 \\ 0 & 0 & \omega_k\cos\phi_k & \sin\phi_k & a_k\cos\phi_k - v_k\omega_k\sin\phi_k & v_k\cos\phi_k \end{bmatrix}, \tag{3.25}$$

such that

$$\mathbf{P}_{\text{CTCA}\to\text{CCA}} \approx \mathbf{J}_{\text{CTCA}\to\text{CCA}}\mathbf{P}_{\text{CTCA}}\mathbf{J}_{\text{CTCA}\to\text{CCA}}^T. \tag{3.26}$$

For the reverse projection, a reduced Jacobian is used:

$$\mathbf{J}_{\text{CCA}\to\text{CTCA}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \cos\phi_k & 0 & 0 & \sin\phi_k & 0 \\ 0 & 0 & \cos\phi_k & 0 & 0 & \sin\phi_k \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.27}$$

such that

$$\mathbf{P}_{\text{CCA}\to\text{CTCA}} \approx \mathbf{J}_{\text{CCA}\to\text{CTCA}}\mathbf{P}_{\text{CCA}}\mathbf{J}_{\text{CCA}\to\text{CTCA}}^T. \tag{3.28}$$

Note that no excess covariance in heading and yaw-rate is transferred from CCA to CTCA due to it not being available.

3.3.2 Model Conditioned Filtering

After the state mixing step, each motion model j performs a model-conditioned EKF recursion using its own nonlinear state-transition and measurement models. Apart from the initialization with mixed state estimates and covariances, this step follows the standard EKF prediction and correction cycle described in Section 2.2.
During the measurement update, the measurement likelihood for each model is evaluated using Equation (2.30). Special care is taken to ensure numerical robustness. The innovation covariance $\mathbf{S}_{j,k}$ is not inverted explicitly; instead, the required linear systems are solved using a Cholesky factorization of $\mathbf{S}_{j,k}$. This approach improves numerical stability and preserves symmetry and positive definiteness of the covariance matrices [8].
If the Cholesky factorization fails due to ill-conditioning, the measurement update is skipped for the affected model and the predicted state estimate is retained. This prevents numerical instabilities from propagating through the IMM recursion.

3.3.3 Mode Probability Update

Following the model-conditioned filter step, the probability associated with each motion model is updated using the measurement likelihoods computed from the innovations. By using the theoretical formulation from Section 2.4.3 a practical implementation may be done.
In the implementation the update is performed in the log domain to improve numerical robustness. For each model j, the log-likelihood of the current measurement is combined with the logarithm of the predicted model probability, resulting in unnormalized log-weights. To prevent model probabilities from collapsing due to numerical underflow, a minimum probability threshold ϵ, is enforced prior to the update.
Normalization is also carried out ensuring stable evaluation even when likelihood values are small or differ significantly between models. This guarantees that the updated mode probabilities form a valid probability distribution.
The complete mode probability update is summarized in Algorithm 1 where the posterior mode probability reflects how each model with their measurements will subsequently be used in the estimation combination step.

3.3.4 Estimation Combination

After updating the mode probabilities, the final step of the IMM cycle is performed, namely the estimate combination step. The theoretical foundation of this step follows the formulation in Section 2.4.4.
In the standard IMM formulation, all mode-conditioned filters are assumed to operate on a common state space, which allows the combined estimate to be computed directly. In the present implementation, however, an inconsistency with the theoretical formulation arises due to the use of motion models with different state representations. As in the interaction step, the main challenge lies in merging the individual filter estimates into a unified state space suitable for combination.

Algorithm 1 IMM Mode Probability Update (Log-Domain).

Require: Prior model probabilities $\mu_{j,k}$, log-likelihoods $\ell_{j,k}$, minimum probability threshold ε
Ensure: Updated model probabilities $\hat{\mu_{j,k}}$

for $j = 1, \ldots, M$ **do**
    $\mu_{j,k} \leftarrow \max(\mu_{j,k}, \varepsilon)$
    $a_j \leftarrow \log(\mu_{j,k}) + \ell_{j,k}$
end for

$m \leftarrow \max_j a_j$

for $j = 1, \ldots, M$ **do**
    $w_j \leftarrow \exp(a_j - m)$
end for

$s \leftarrow \sum_{j=1}^{M} w_j$

if $s > 0$ **then**
    for $j = 1, \ldots, M$ **do**
        $\hat{\mu_{j,k}} \leftarrow w_j / s$
    end for
end if

Due to this limitation, the final IMM output is formed through an asymmetric estimation-combination step. The shared object-level states, such as position, speed, and tangential acceleration, are obtained by probability-weighted averaging of the two model estimates. Denoting the posterior mode probabilities by μ_{CTCA} and μ_{CCA}, the fused mean for a shared state can be expressed as

$$\hat{\mathbf{x}}_{\mathrm{IMM}} = \mu_{\mathrm{CTCA}} \hat{\mathbf{x}}_{\mathrm{CTCA}} + \mu_{\mathrm{CCA}} \hat{\mathbf{x}}_{\mathrm{CCA}}. \tag{3.29}$$

In the current implementation, however, the heading and yaw rate are not fully fused across models. Instead, they are taken directly from the CTCA branch, since only that model explicitly represents yaw rate and maintains a dynamically consistent curvilinear turning state.
The covariance combination is performed in the CCA state space. To achieve this, the CTCA posterior covariance is first transformed into CCA coordinates, after which the standard IMM moment-matching expression is applied,

$$\mathbf{P}_{\mathrm{IMM}} = \mu_{\mathrm{CCA}} \left(\mathbf{P}_{\mathrm{CCA}} + \mathbf{d}_{\mathrm{CCA}} \mathbf{d}_{\mathrm{CCA}}^T \right) + \mu_{\mathrm{CTCA}} \left(\mathbf{P}_{\mathrm{CTCA}\to\mathrm{CCA}} + \mathbf{d}_{\mathrm{CTCA}} \mathbf{d}_{\mathrm{CTCA}}^T \right), \tag{3.30}$$

where

$$\mathbf{d}_{\mathrm{CCA}} = \hat{\mathbf{x}}_{\mathrm{CCA}} - \hat{\mathbf{x}}_{\mathrm{IMM}}^{\mathrm{CCA}}, \qquad \mathbf{d}_{\mathrm{CTCA}} = \hat{\mathbf{x}}_{\mathrm{CTCA}\to\mathrm{CCA}} - \hat{\mathbf{x}}_{\mathrm{IMM}}^{\mathrm{CCA}}. \tag{3.31}$$

Here, $\hat{\mathbf{x}}_{\text{IMM}}^{\text{CCA}}$ denotes the fused mean expressed in CCA coordinates, $\hat{\mathbf{x}}_{\text{CCA}}$ is the posterior CCA mean, and $\hat{\mathbf{x}}_{\text{CTCA}\to\text{CCA}}$ is the CTCA posterior mean projected into the CCA space. The vectors $\mathbf{d}_{\text{CCA}}$ and $\mathbf{d}_{\text{CTCA}}$ account for the spread between the individual model estimates and the fused mean, ensuring that the final covariance reflects both the internal uncertainty of each model and the disagreement between them.
The reasoning that the covariance is fused in CCA state space lies in numerical stability. Projecting the CCA uncertainty into CTCA at especially low speeds will cause singularities in the heading transform, causing inflated covariances and skewing the final result.

3.4 Key Performance Indicators

The main KPI used to compare motion models, filter tuning and pseudo measurements implementation is P-GOSPA. Additionally the RMSE of position, tangential velocity and heading is also deployed as a means of understanding how the trackers behave rather than evaluating overall performance.

3.4.1 P-GOSPA

The P-GOSPA metric (2.35) can be used to evaluate and compare multiple tracker states of multiple tracks to multiple ground truth tracks where the number of estimated tracks necessarily is not the same as the number of ground truth tracks, by penalizing redundant, missing or lost tracks. This is done by applying a cost to localization, missed ground truths and false tracks for each scan and time average it to a scalar KPI for an entire scenario. We have chosen to compare position, tangential velocity and heading between each tracker since these states is what defines how a tracked objects bounding box is located and oriented which is important for ADAS features such as adaptive cruise control, blind spot detection with others.
As suggested by Yuxuan Xia et al [24] the Wasserstein distance is used to calculate the localization cost metric for gaussian state probability $p_x = \mathcal{N}(\mathbf{m}_x, \mathbf{P}_x)$ and ground truth probability $p_y = \mathcal{N}(\mathbf{m}_y, \mathbf{P}_y)$ which is defined as

$$W_2^2(p_x, p_y) = \|\mathbf{m}_x - \mathbf{m}_y\|^2 + \text{tr}(\mathbf{P}_x + \mathbf{P}_y - 2(\mathbf{P}_y{}^{1/2}\mathbf{P}_x\mathbf{P}_y{}^{1/2})^{1/2}). \tag{3.32}$$

Since the states considered in P-GOSPA have different units and physical meaning the states are all normalized by its acceptable magnitude of deviation to make them comparable. The normalization is defined according to

$$p_{norm} = 0.5, \quad v_{norm} = \frac{4}{3.6}, \quad \phi_{norm} = 5\frac{\pi}{180}, \tag{3.33}$$

where $p_{norm} = \sqrt{x_{norm}^2 + y_{norm}^2}$. These were chosen according to the acceptable errors of 0.5 m in absolute position, 4 km/h tangential velocity and 5 degrees of heading angle. The CCA tangential velocity covariance is derived from velocity covariances in x and y direction through a first order error propagation

$$\sigma^2_{v,\text{cca}} \approx \frac{v_x^2\sigma^2_{v_x} + 2v_x v_y \sigma_{v_x v_y} + v_y^2\sigma^2_{v_y}}{v_x^2 + v_y^2}, \quad \text{when } v_x^2 + v_y^2 > 0. \tag{3.34}$$

Tuning of P-GOSPA (2.35) is done according to

- $c = 5$
  Reasonable cut-off distance considering the compared states normalized by the acceptable state deviation.
- $p = 2$
  Quadratic outlier sensitivity puts more penalty on outliers (common in MOT).
- $\alpha = 2$
  Ensuring symmetry between FP/FN and bad associations.

3.5 Motion Model Benchmarking

Benchmarking of models are conducted through a number of scenarios that are created to provoke specific motion conditions that either challenge or favour the explored motion models. Each scenario are designed so that they isolate a specific aspects of target dynamics. Since different motion models rely on different assumptions about target dynamics, the scenarios are chosen to reveal cases in which these assumptions hold and cases in which they break down. The GT trajectories of each scenario can be seen in Appendix section A.1
The different models are compared in predicted vs ground truth trajectory by comparing position, velocity and heading, which is evaluated with the P-GOSPA KPI. Since the heading is not a property of the CCA motion model heading has been defined as the direction of the velocity vector according to arctan $\frac{v_y}{v_x}$, and additionally heading for each motion model is compared to GT as Dirac components without a covariance. The CCA models covariance tuning parameters defined in equation are set to $q_{lon} = 1.0$, $q_{lat} = 0.5$, and the CTCA tuning parameters defined in equation (3.16) are set according to $\sigma_{\dot{a}}^2 = 1.0$, $\sigma_{\dot{\omega}}^2 = 0.1$. The measurement variances are set according to the defined measurement uncertainties defined in equation (3.6). Since the goal of this initial benchmarking purely is to gain an initial understanding of the models strengths and weaknesses the covariance tuning will not be iterated further until later in the report, although the crude tuning and its potential affects will be considered in following results.

3.5.1 Scenario 1 - Constant speed crossing 2 targets

During the constant speed crossing 2 targets scenario the host is stationary whilst two targets crossing perpendicular to host from different directions. When both targets line up in front of host the furthest target gets occluded by the nearest resulting in loss of radar detections for some scans.
As can be observed in Figure 3.3 the P-GOSPA trajectory for both motion models are quite uneventful with both the CCA and CTCA model performing good tracking. Notably is the P-GOSPA metrics observed in table 3.4 which suggests the CTCA model is the clear better performer. This is due to the state covariances being

higher for the CCA model compared to the CTCA model. Rerunning the P-GOSPA metrics calculations and assuming Dirac tracks and GT, the CCA model result in a P-GOSPA score of 0.588 and the CTCA model 0.602. This suggests that if simply comparing pure state absolute error over the scenario the models have similar tracking performance. The higher state uncertainty originally observed with the CCA model might purely be due to poor covariance matrix tuning, which is further evaluated in section 4.2.

Table 3.4: Scenario 1 P-GOSPA metrics.

Model	Localization cost	Missed cost	False Cost	P-GOSPA
CCA	0.900	0	0	0.900
CTCA	0.553	0.009	0	0.559

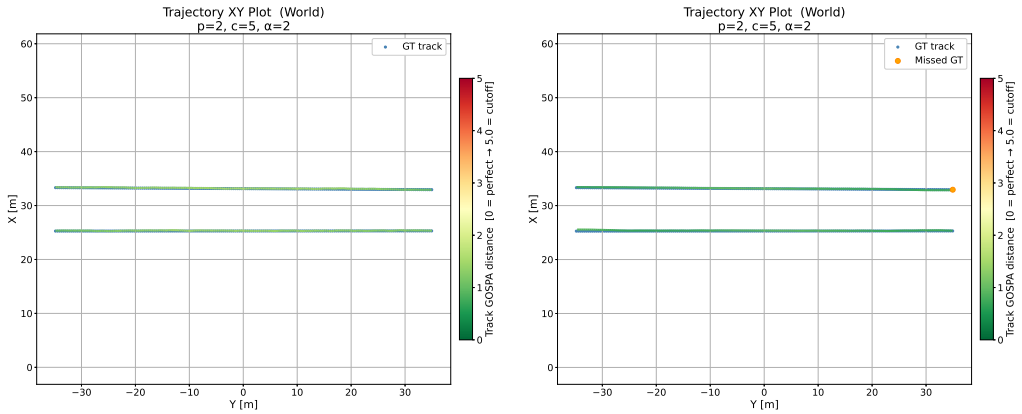


Figure 3.3: Scenario 1 CCA trajectory (left) vs CTCA trajectory (right) in global coordinate frame.

3.5.2 Scenario 2 - Highway overtake 2 targets

The highway overtake 2 targets scenario consists of straight highway driving where the host overtakes one target before later being overtaken by another target. The second target gets out and then in to the hosts range of view again resulting in 3 different trajectories for the two targets, each with constant velocity.
The motion models is observed in Figure 3.4 to have similar performance in tracking without any lost or missed tracks. Table 3.5 shows the P-GOSPA summary over the scenario where the CTCA model is observed to have slightly better tracking which is again down to state uncertainties.

Table 3.5: Scenario 2 P-GOSPA metrics.

Model	Localization cost	Missed cost	False Cost	P-GOSPA
CCA	1.288	0	0	1.288
CTCA	0.909	0	0	0.909

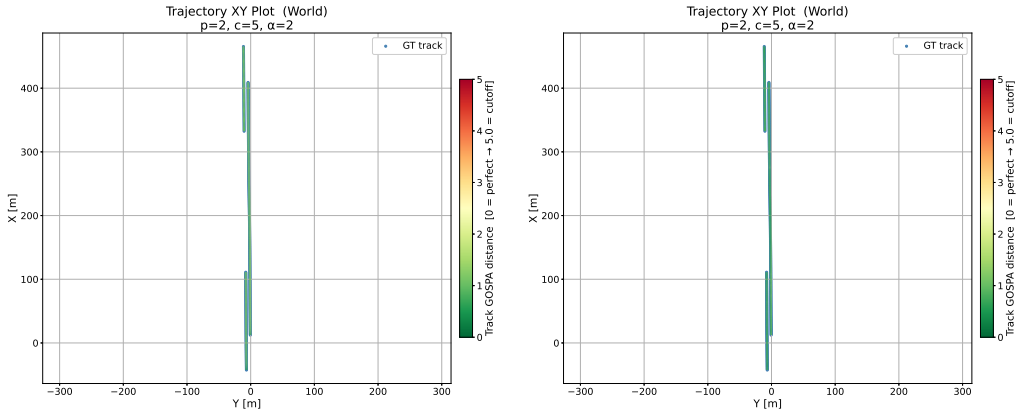


Figure 3.4: Scenario 2 CCA trajectory (left) vs CTCA trajectory (right) in global coordinate frame.

3.5.3 Scenario 3 - Host straight target turns right

In scenario 3 the target makes a 90 degree right hand turn in front of host and proceeds to drive away from host. After the target have made the right hand turn the host continues to drive straight into the intersection. Therefore this scenario captures varying acceleration due to braking and a non zero yaw-acceleration is introduced.

Figure 3.5 shows that the CCA model can't keep up in targets 90 degree turn where the P-GOSPA localisation penalty exceeds the cutoff distance and the estimated track is dissociated with the GT resulting in a missed and false track. After the turn the CCA model is reassociated and is observed to handle the targets unidirectional acceleration perpendicular to host better, where the CTCA model is instead struggling. The CTCA model however manages to keep the estimated track within P-GOSPA:s cutoff distance during the entire scenario which results in the clear advantage according to the P-GOSPA summary in table 3.6. Figure 3.6 shows GT BB vs track BB for a number of scans for each motion model, where it is observed that the CCA motion models main issue is rotating the track as GT is turning. The reason for still keeping up with GT position relatively well is the measurement BB position updates.

Table 3.6: Scenario 3 P-GOSPA metrics.

Model	Localization cost	Missed cost	False Cost	P-GOSPA
CCA	2.586	0.231	0.231	2.913
CTCA	1.310	0	0	1.310

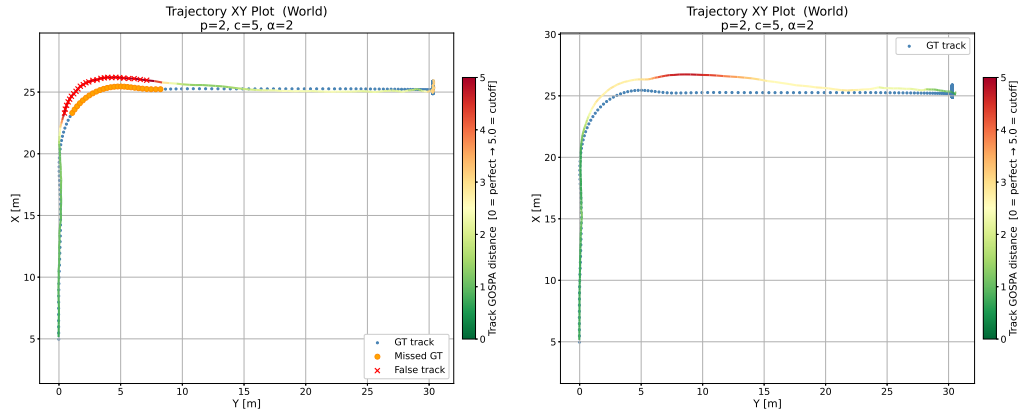


Figure 3.5: Scenario 3 CCA trajectory (left) vs CTCA trajectory (right) in global coordinate frame.

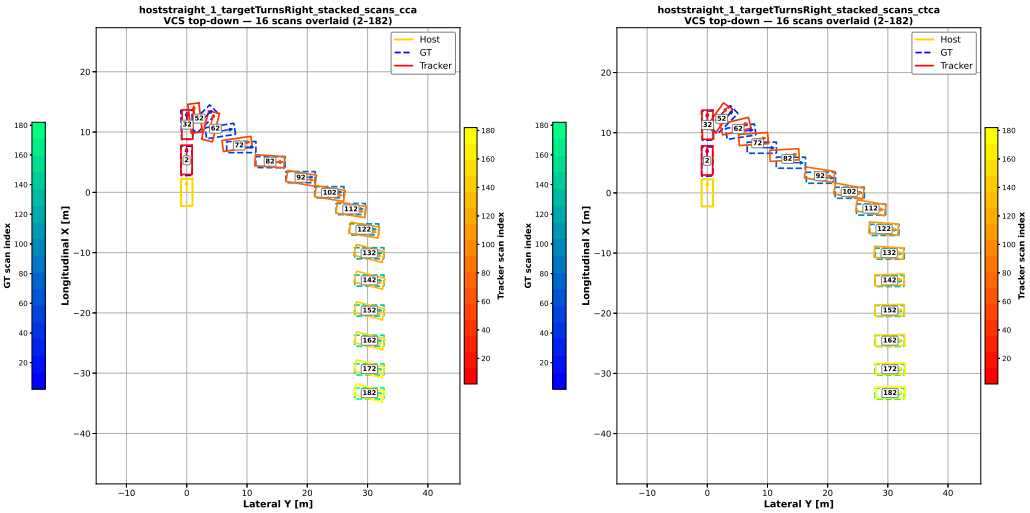


Figure 3.6: Scenario 3 CCA stacked scans (left) vs CTCA stacked scans (right) in host coordinate frame.

3.5.4 Scenario 4 - Roundabout 1 Target

In the fourth scenario a constant turn is evaluated within a roundabout. The target vehicle enters the roundabout from the left relative host, drives around and exits in the same direction from which it entered. Inside the roundabout the target keeps constant velocity and yaw-rate, however upon entering and exiting the target has a nonzero acceleration and yaw-acceleration.
Figure 3.7 shows a clear advantage to the CTCA model where the CCA model results in missed and false tracks due to exceeding the cutoff distance in P-GOSPA localization cost. The benefit here of the CTCA model is the ability to keep a relatively low heading error over the entire scenario compared to the CCA model which suffers from a heading error and are therefore lagging behind the GT in position as well, which is observed from Figure 3.8 and 3.9. The summarized P-GOSPA metrics can be seen in 3.7.

Table 3.7: Scenario 4 P-GOSPA metrics.

Model	Localization cost	Missed cost	False Cost	P-GOSPA
CCA	1.013	0.376	0.376	1.546
CTCA	0.694	0	0.007	0.701

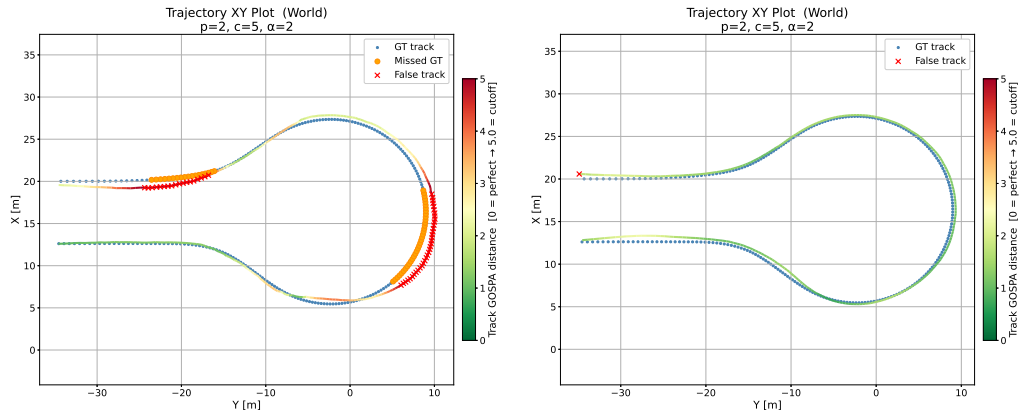


Figure 3.7: Scenario 4 CCA trajectory (left) vs CTCA trajectory (right) in global coordinate frame.

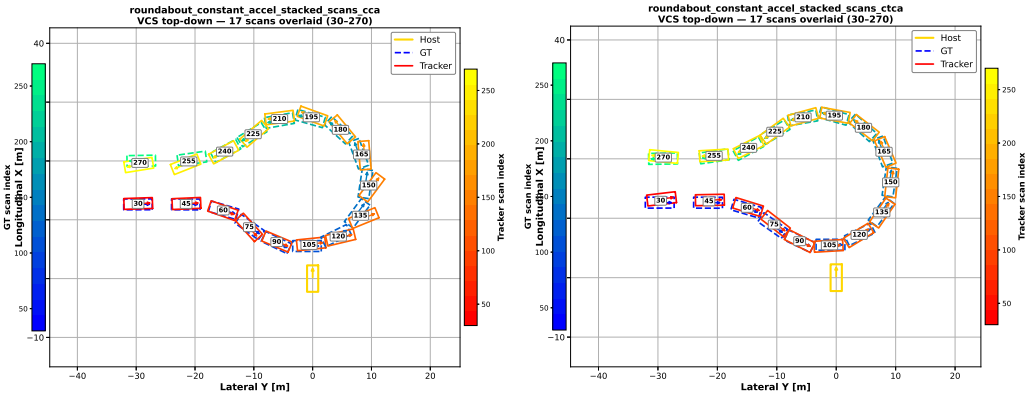


Figure 3.8: Scenario 4 CCA stacked scans (left) vs CTCA stacked scans (right) in host coordinate frame.

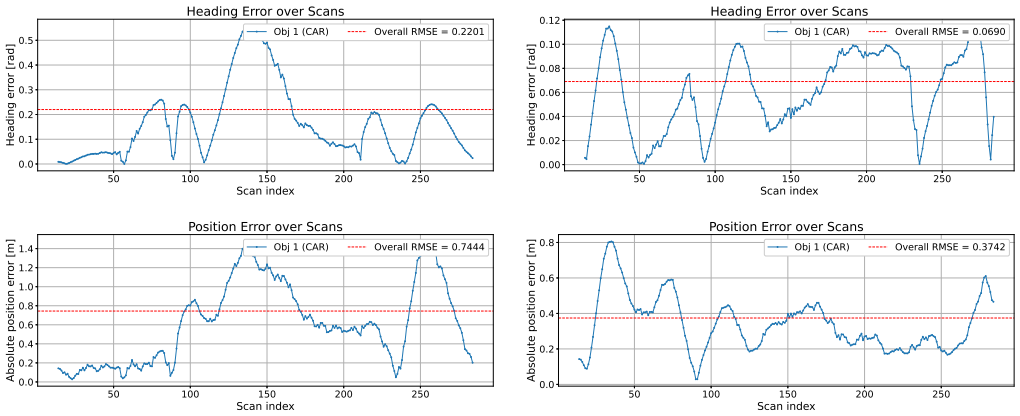


Figure 3.9: Scenario 4 Heading and position error, CCA (left) vs CTCA (right).

3.5.5 Scenario 5 - Stop and go 2 targets

In scenario 5 the host is following a target braking whilst nearing an intersection. At the same time another target is braking before entering the intersection perpendicular from host to the left. All targets and host comes to a complete stop. Target in front of host accelerates straight through the intersection and the target to the left accelerates perpendicular to host through the intersection. Host lastly accelerates straight through the intersection.

By observing Figure 3.10 it can be concluded that both motion models perform good tracking throughout the scenario where the advantage leans towards the CTCA model if looking at table 3.8. Running the P-GOSPA scenario evaluation again without considering state covariances the CCA model achieves a score of 1.228 and the CTCA model a score of 1.303, which suggests that the CCA model has better tracking if considering states RMSE only but with higher uncertainties. This is probably due to the crude tuning of the covariance matrix but wont be further evaluated in this initial benchmarking.

Table 3.8: Scenario 5 P-GOSPA metrics.

Model	Localization cost	Missed cost	False Cost	P-GOSPA
CCA	1.489	0	0	1.489
CTCA	1.063	0	0.009	1.070

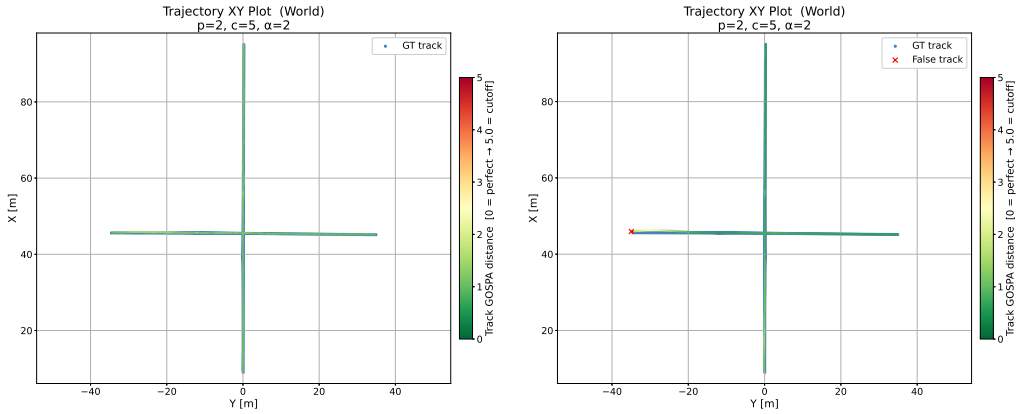


Figure 3.10: Scenario 5 CCA trajectory (left) vs CTCA trajectory (right) in global coordinate frame.

3.5.6 Scenario 6 - Yaw rate 1 target

Scenario 6 focuses on yawing motion where the targets yaw-rate is constantly changing. Here the target vehicle drives in a S shape away from the host, keeping a constant velocity but having a oscillating yaw-rate. This enables an evaluation of the motion models ability to adapt to yaw-rate changes.

This scenario shows the adaptability of the CTCA model which is observed in Figure 3.11 to follow the GT trajectory relatively well compared to the CCA model which clearly struggles to make the turns by inducing object yaw rates. This is visualized in Figure 3.12 where the GT and tracker bounding boxes are plotted throughout the scenario by stacking scans. The summarized P-GOSPA comparison can be seen in table 3.9.

Table 3.9: Scenario 6 P-GOSPA metrics.

Model	Localization cost	Missed cost	False Cost	P-GOSPA
CCA	0.524	1.305	1.412	2.453
CTCA	1.282	0.012	0	1.294

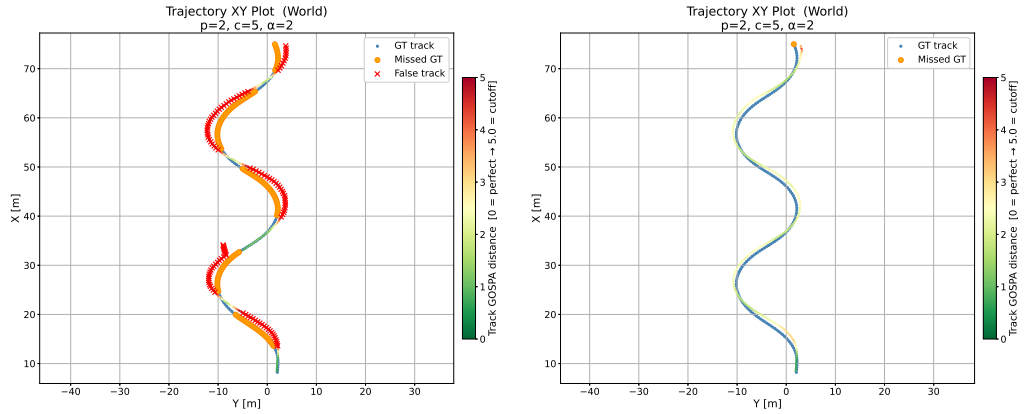


Figure 3.11: Scenario 6 CCA trajectory (left) vs CTCA trajectory (right) in global coordinate frame.

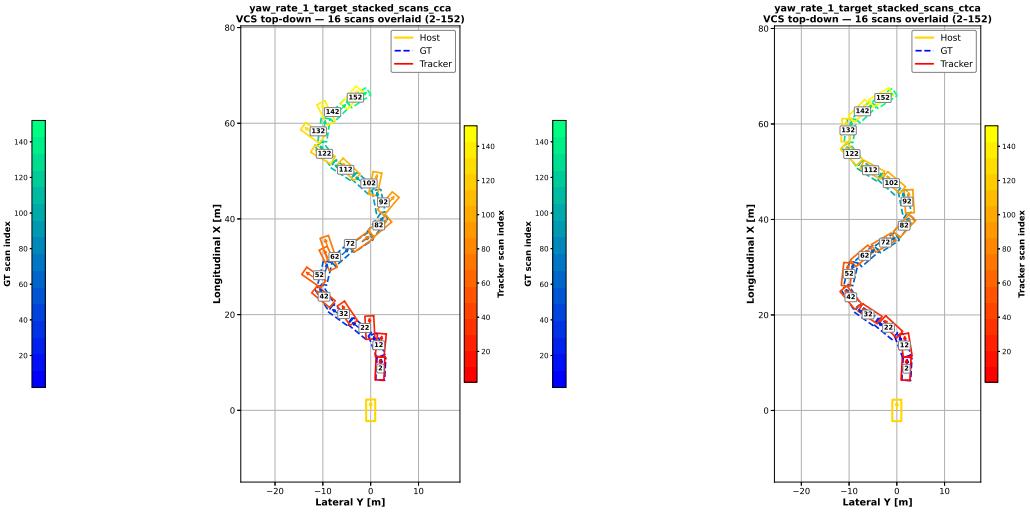


Figure 3.12: Scenario 6 CCA stacked scans (left) vs CTCA stacked scans (right) in host coordinate frame.

3.5.7 Scenario Benchmarking Conclusions

From the conducted scenario benchmarking, it is evident that the CTCA motion model provides a more flexible and robust representation of vehicle dynamics due to the explicit inclusion of heading and yaw-rate in the state vector. As a consequence the CTCA model consistently outperforms the CCA model in scenarios involving curved trajectories such as intersection turns, roundabout maneuvers, and oscillatory yaw-rate motion. In these cases the CCA model frequently violates its underlying assumptions leading to increased localization error, track loss and inaccurate state estimates.

In contrast, for scenarios characterized by predominantly unidirectional motion with limited curvature such as constant velocity highway driving or simple crossing scenarios, the CCA model achieves comparable, and in some cases marginally lower

state estimation error when covariances are ignored. This suggests that under idealized motion conditions, the simpler Cartesian acceleration model can describe target kinematics efficiently. However these advantages are often offset in the full P-GOSPA evaluation by higher posterior state uncertainties which are strongly influenced by the chosen covariance tuning.

3.6 Automated Tuning using P-GOSPA

The task of finding the perfect tuning mentioned in section 1.1.3 is a well known issue, which is encountered in this application with MOT using EKF. As mentioned some of the issues are defining a well behaved KPI metric, having representative data and limiting the number of parameters considered. All of this contribute to a highly nonlinear objective function that should be optimized effectively resulting in tuning of a black box according to Figure 3.13.

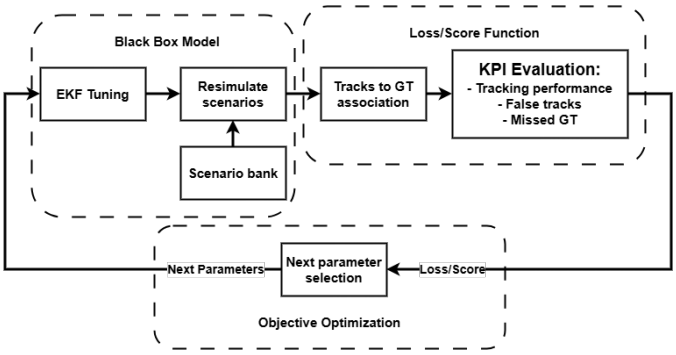


Figure 3.13: Automated tuning pipeline interpreted as black box optimization problem.

3.6.1 Next Parameter Selection

The strategy of choosing the next parameter candidates is CMA-ES where an overview of the implementation can be seen in Algorithm 2 where checks for parameter bounds and early stopping has been added.

Algorithm 2 CMA-ES implementation pseudo-code.

Require: Objective function $f(\mathbf{x})$
Require: Initial mean $\mathbf{m}^{(0)} \in \mathbb{R}^n$ (log-parameters)
Require: Initial step-size $\sigma^{(0)}$
Require: Bounds $(\mathbf{l}, \mathbf{u})$ in log-space
Require: Population size λ, number of parents μ
Require: Maximum epochs $T_{\max}$
Initialize covariance $\mathbf{C}^{(0)} \leftarrow \mathbf{I}$
Initialize evolution paths $\mathbf{p}_\sigma \leftarrow \mathbf{0}$, $\mathbf{p}_c \leftarrow \mathbf{0}$
$f_{\text{best}} \leftarrow \infty$, $\mathbf{x}_{\text{best}} \leftarrow \varnothing$
for $t = 0$ to $T_{\max} - 1$ **do** ▷ Sampling
    for $k = 1$ to λ **do**
        Sample $\mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
        $\mathbf{x}_k \leftarrow \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{C}^{(t)^{1/2}} \mathbf{z}_k$
        if bounds exist **then**
            $\mathbf{x}_k \leftarrow \text{clip}(\mathbf{x}_k, \mathbf{l}, \mathbf{u})$
        end if
        Evaluate $f_k = f(\mathbf{x}_k)$ ▷ Run tracker and P-GOSPA evaluation
        if $f_k < f_{\text{best}}$ **then**
            $f_{\text{best}} \leftarrow f_k$, $\mathbf{x}_{\text{best}} \leftarrow \mathbf{x}_k$
        end if
    end for
    ▷ Selection
    Sort $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ by ascending f_k
    Select top μ individuals indexed by $i = 1, \ldots, \mu$
    ▷ Mean Update
    $\mathbf{m}^{(t+1)} \leftarrow \mathbf{m}^{(t)} + \sigma^{(t)} \sum_{i=1}^{\mu} w_i \mathbf{y}_i$
    ▷ Step-Size Update
    Update evolution path $\mathbf{p}_\sigma$
    Adapt $\sigma^{(t+1)}$ via cumulative step-size adaptation
    ▷ Covariance Update
    Update evolution path $\mathbf{p}_c$
    if bounds exist **then**
        $\mathbf{m}^{(t+1)} \leftarrow \text{clip}(\mathbf{m}^{(t+1)}, \mathbf{l}, \mathbf{u})$
    end if
    ▷ Stopping Criteria
    if $\sigma^{(t+1)} < \sigma_{\min}$ **then**
        break
    end if
    if relative improvement $< \varepsilon$ over window **then**
        break
    end if
end for
return $\mathbf{x}_{\text{best}}, f_{\text{best}}$

3.6.2 Proof of Concept

Different methods can be used to show that the chosen tuning optimization method has potential. First and foremost given the same tuning scenario the method should end up with the same optima multiple times regardless of where the initial parameter exploration is.
Another method is to actually test a grid of parameters, plot a heat map of P-GOSPA scores and compare it to the output of the tuning optimization method. To make this feasible a smaller grid is defined within reasonable parameter bounds and limit to a single scenario that are to be evaluated. Assuming simulating a scenario, writing results, extracting results and calculating KPI takes 7 seconds using a 160×160 grid would result in an evaluation time of approximately 50 hours. For this the CTCA model is chosen to be evaluated together with a roundabout scenario with two targets. The host is at standstill, one car taking the third exit whilst the other takes a shortcut through the roundabout. The trajectories can be seen in the following Figure

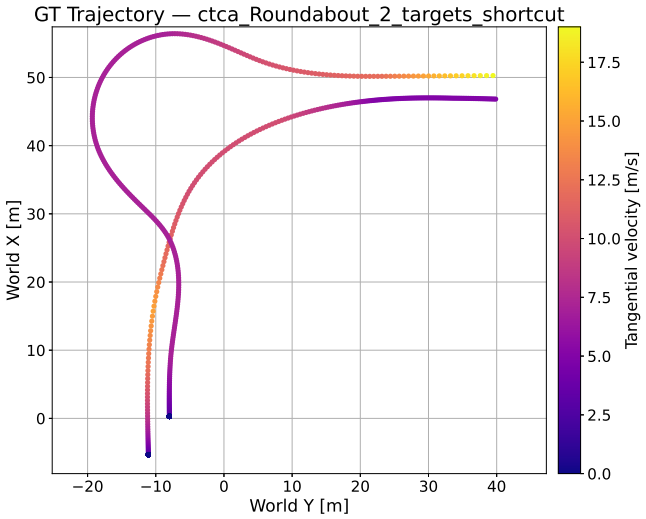


Figure 3.14: Proof of concept grid tuning scenario in global coordinate frame.

To ensure this method is feasable the tuning of $(q_1, q_2) = (\sigma_{\dot{a}^2}, \sigma_{\dot{\omega}^2})$ are done in log-space between the bounds of $[-6, 4]$ with 160 number of steps on each axis. The same bounds is then set for the CMA-ES algorithm 2 which is compared to the created grid tuning heat map.

3.6.3 Final Tuning

The final tuning of the whole filter pipeline is done in three steps

- Tuning of CCA model
- Tuning of CTCA model
- Tuning of IMM with the best CCA and CTCA tuning

Two sets of scenarios are created, one for CCA and CTCA model, that are meant to reflect the motion models strengths whilst keeping some general driving scenarios in

both sets for robustness in the tuning avoiding over fitting to a single motion pattern. The motion models are tuned on respective scenario set before being combined in the IMM for tuning where all scenarios are merged into a large set. One test scenario is kept unseen to the CCA, CTCA and IMM model tuning that is used to compare pure CCA, pure CTCA and IMM.
The parameters q_1 and q_2 that are evaluated in the tuning process are mapped to different parameters for the CTCA, CCA and IMM filter.

- CCA: $q_1 = \sigma^2_{\dot{a}_x}$, $q_2 = \sigma^2_{\dot{a}_y}$ where $\sigma^2_{\dot{a}_x}$ and $\sigma^2_{\dot{a}_y}$ stems from equation (3.8)
- CTCA: $q_1 = \sigma^2_{\dot{a}}$, $q_2 = \sigma^2_{\dot{\omega}}$ where $\sigma^2_{\dot{a}}$ and $\sigma^2_{\dot{\omega}}$ stems from equation (3.16)
- IMM: q_1, q_2 tunes the Markov transition matrix $\Pi = [\pi_{ij}]$ defined in section 2.4.1 according to equation (3.35).

$$\mathbf{\Pi} = \begin{bmatrix} q_1 & 1-q_1 \\ 1-q_2 & q_2 \end{bmatrix} = \begin{bmatrix} \pi_{\text{CTCA}\to\text{CTCA}} & \pi_{\text{CTCA}\to\text{CCA}} \\ \pi_{\text{CCA}\to\text{CTCA}} & \pi_{\text{CCA}\to\text{CCA}} \end{bmatrix}. \tag{3.35}$$

3.6.3.1 CCA scenario set

The CCA set of scenarios concentrate on heavy accelerating motions in different radar azimuth angles. The scenarios are meant to mimic highway driving, overtaking and low speed high accelerations in crossings. Trajectory plots of the scenarios can be found in Appendix A.2.1.

Constant Speed Crossing 2 Targets This is the same scenario described in section 3.5.1.

Crossing 2 Targets The crossing 2 targets scenario is built similarily to scenario described in section 3.5.1 with the exception of having constant acceleration instead of constant velocity for each target.

Curvy Road 1 Target This scenario consists of highway driving with slight turning in constant velocity. The host is driving behind the target through the entire scenario.

Highway Constant Speed Constant velocity highway driving where the host is overtaken by one target whilst driving behind another target through the entire scenario.

Highway Overtake 2 Targets This is the same scenario described in section 3.5.2.

Overtake 1 Target Host is overtaking one target during highway driving. The target then accelerates to overtake host again.

Stop and Go 2 Targets This is the same scenario described in section 3.5.5.

3.6.3.2 CTCA scenario set

The CTCA set of scenarios focus more on curved trajectories in different environments such as roundabouts, narrow crossings and slightly turning highways with overtaking. Trajectory plots of the scenarios can be found in Appendix A.2.2.

Highway 1 Target Constant Turn Highway constant turn driving with the host following one target before later then overtaking the target. Both host and target are accelerating and decelerating through the scenario.

Highway 2 Targets Constant Turn Overtake Highway constant turn driving where the host are overtaking two targets before being overtaken later. The scenario includes both constant velocity driving in constant turn and later combining constant turn with varying acceleration.

Highway Overtake 2 targets This is the same scenario described in section 3.5.2.

Host Straight 1 Target Turns Right This is the same scenario described in section 3.5.3.

Junction Multiple Targets Junction with multiple targets turning, making U-turns and driving straight through. This results in a diverse scenario including varying accelerations with varying yaw-rates challenging the CTCA motion model assumptions.

Roundabout 1 Target Stopping Roundabout scenario with a vehicle stopping in the middle of the roundabout to combine acceleration changes in a nearly constant turn scenario.

Roundabout 2 Targets Roundabout scenario with the tracked vehicles taking the second and third exit. The scenario includes acceleration changes while driving straight and yaw-rate changes in nearly constant velocity.

Roundabout 2 Targets Shortcut Roundabout scenario with one tracked vehicle taking the third exit whilst the other takes a shortcut accelerating through the roundabout. This includes acceleration changes in straight driving, nearly constant acceleration in constant turning and change of yaw-rate in entering and exiting the roundabout.

Roundabout 1 Target This is the same scenario described in section 3.5.4.

Yaw Rate 1 Target This is the same scenario described in section 3.5.6.

3.6.3.3 Test scenario

The test scenario is a longer scenario with different sections covering numerous motion trajectories. The first section is biased towards the CTCA model with a roundabout and three vehicles driving in the roundabout in different velocities, accelerations and yaw-rates. Then host is driving to an intersection where three cars are accelerating and driving through in different velocities whilst coming from different azimuth angles. The last section covers a highway/country road with slight constant turning where host follows a car. The ground truth trajectories of the scenario can be seen in Figure 3.15.

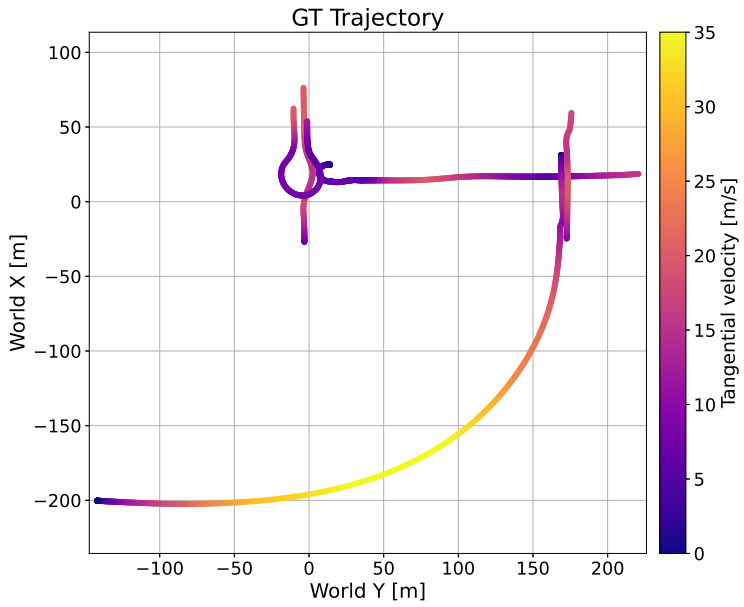


Figure 3.15: Test scenario ground truth trajectories in global coordinate frame.

4

Results

4.1 Automated tuning

This section presents the results of the automated tuning approach introduced in Section 3.6. The results are divided into a proof-of-concept study and a final tuning evaluation.

4.1.1 Automated tuning proof-of-concept

The resulting (q_1, q_2) grid from the brute force tuning can be seen in Figure 4.1 and CMA-ES tuning on the same scenario and motion model can be seen in Figures 4.2 and 4.3.

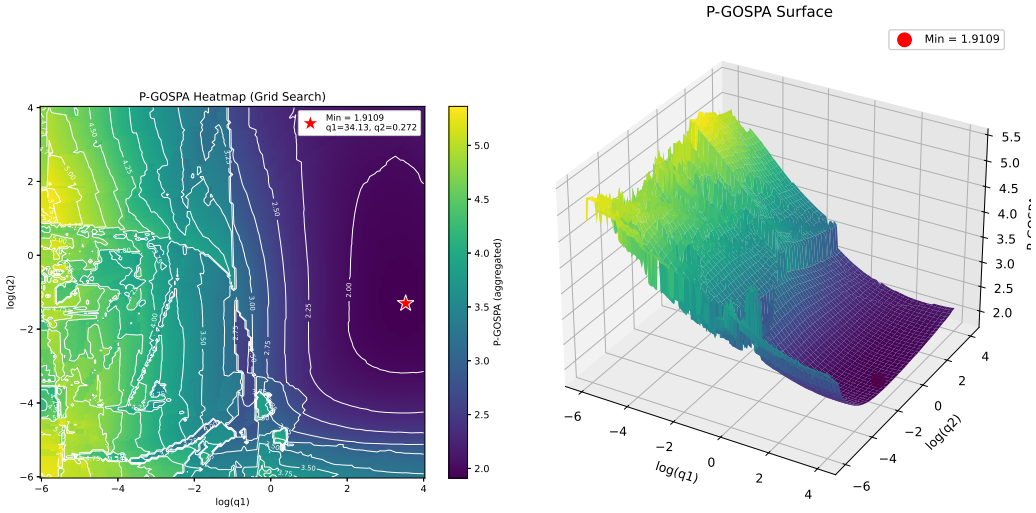


Figure 4.1: P-GOSPA surface over $(\log(q_1), \log(q_2))$ grid.

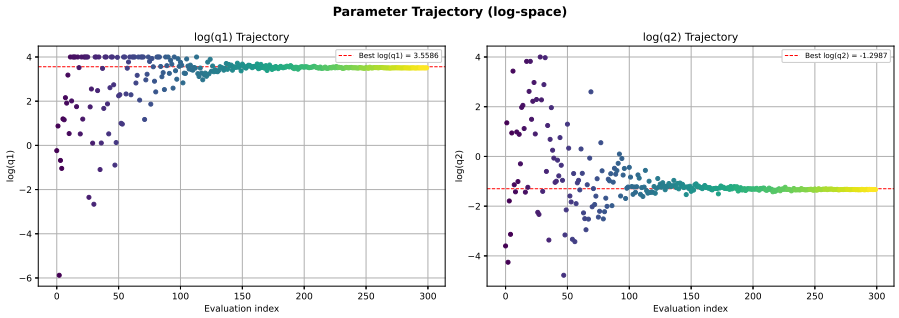


Figure 4.2: CMA-ES proof of concept tuning result.

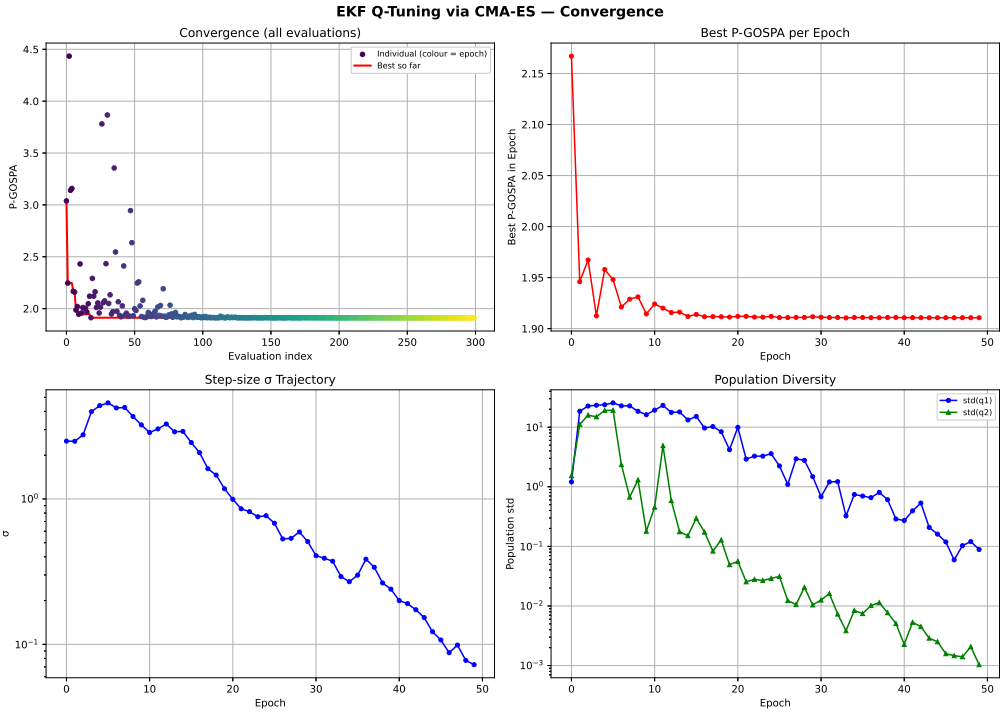


Figure 4.3: CMA-ES proof of concept tuning metrics.

Comparison of the grid tuning heatmap 4.1 and CMA-ES tuning iterations 4.2 can be seen in table 4.1 which shows that the CMA-ES algorithm actually manages to find a lower minimum than the grid tuning approach due to the coarse steps in the defined grid.

Table 4.1: Best tuning iterations comparison.

	$\sigma_{\dot{a}}^2$	$\sigma_{\dot{\omega}}^2$	P-GOSPA
Grid Tuning:	34.130	0.272	1.9109
CMA-ES:	35.114	0.273	1.9105

4.1.2 Final Tuning

The tuning results of the CCA and CTCA model along with the IMM filter can be seen in table 4.2.

Table 4.2: CMA-ES tuning best parameters.

	CCA	CTCA	IMM
q_1	0.2266	18.4912	0.7276
q_2	0.0555	1.1204	0.4529

The tuning convergence for each motion model and IMM filter tuning can be seen in Figures 4.4, 4.5 and 4.6.

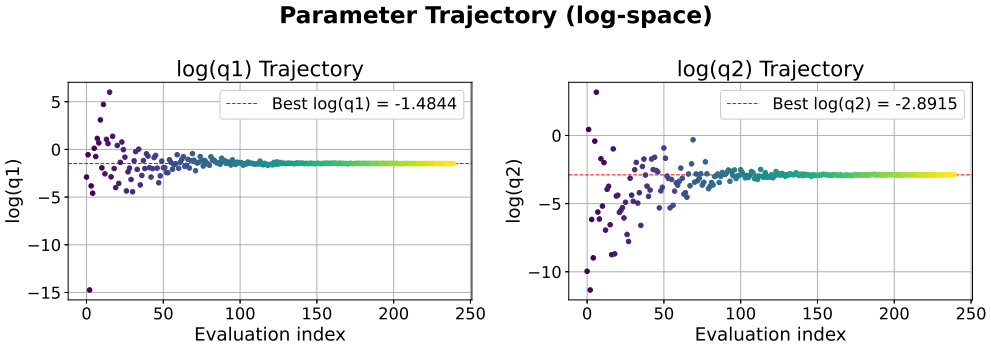


Figure 4.4: Final CCA tuning iterations.

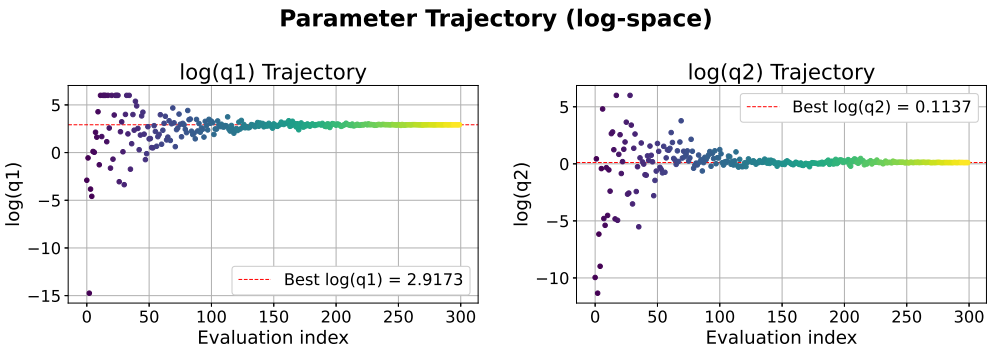


Figure 4.5: Final CTCA tuning iterations.

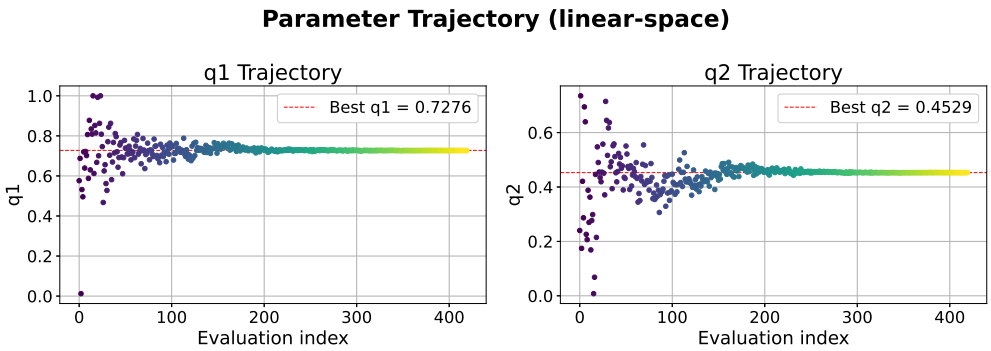


Figure 4.6: Final IMM tuning iterations.

4.1.3 Test Scenario

The final tuning according to table 4.2 is used to compare the CCA, CTCA and IMM against each other. For the IMM the individual models are tuned according to the final tuning as well.

The results from the different models on the test scenario can be seen in table 4.3.

Table 4.3: Test scenario P-GOSPA results per model

CCA	CTCA	IMM
3.55	1.54	1.45

As the P-GOSPA scores for each model suggests the CCA model is lagging behind both the CTCA model and IMM. The difference between the CTCA and IMM is smaller with slight advantage to the IMM.

4.1.3.1 CCA

The P-GOSPA trajectory for the CCA model can be seen in Figure 4.7 where it is apparent that the CCA model is lacking in ability to make turns in both the roundabout and highway constant turn section. This leads to the object bounding box not overlapping with any new radar detections and thus the EKF update only relies on predictions where the tracks coast away from ground truths and new tracks are initialized. This results in false tracks and missed ground truths where the decomposition and summary of the applied P-GOSPA score can be seen in Figure 4.8. Otherwise unidirectional trajectories is handled well with some ability to do lane changes and slight turning.

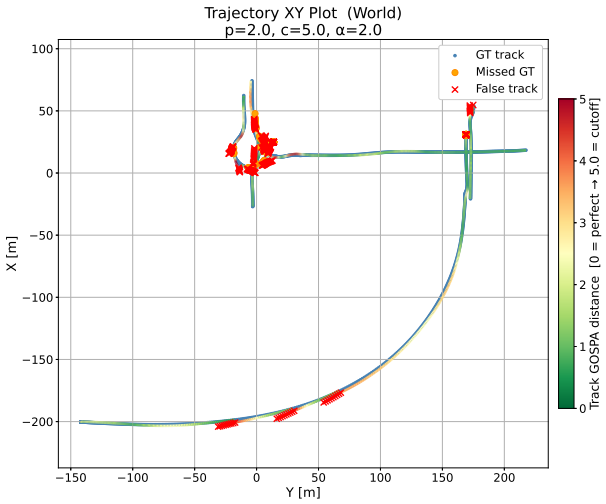


Figure 4.7: Test scenario P-GOSPA trajectory CCA in global coordinate frame.

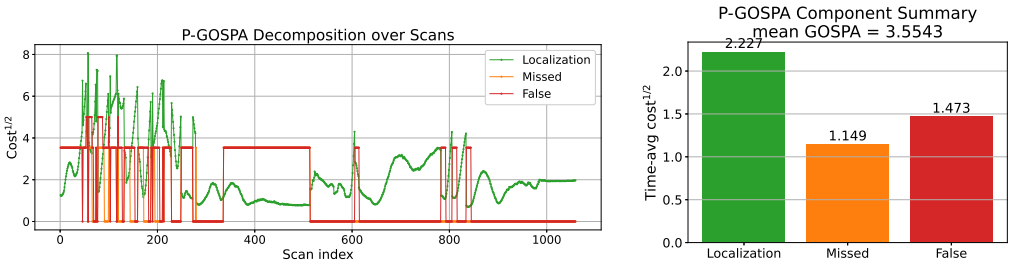


Figure 4.8: Test scenario P-GOSPA metrics using CCA.

4.1.3.2 CTCA

As is shown in the P-GOSPA trajectory in Figure 4.9 the CTCA model is overall capable of handling more diverse cases than the CCA model. False and missed tracks in the P-GOSPA trajectory is due to object initialization which only occurs for a few scans according to Figure 4.10 until the tracks are corrected by measurement update.

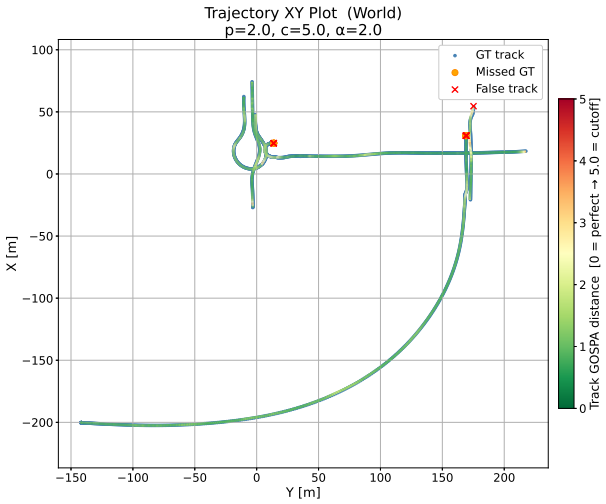


Figure 4.9: Test scenario P-GOSPA trajectory CTCA in global coordinate frame.

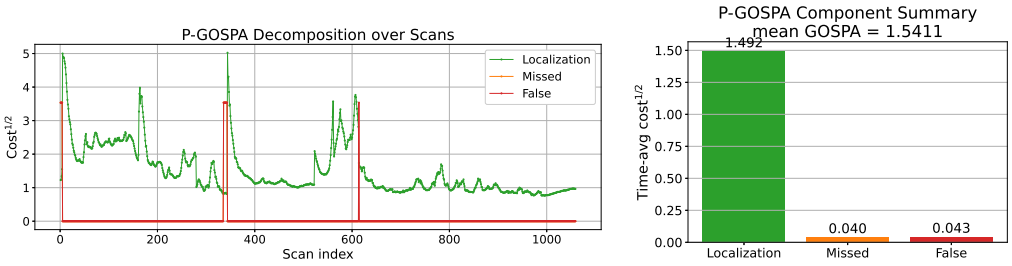


Figure 4.10: Test scenario P-GOSPA metrics using CTCA.

4.1.3.3 IMM

The P-GOSPA trajectory plot in Figure 4.11 for the IMM shows similar performance to the CTCA whereas the P-GOSPA metrics in Figure 4.12 shows a slight advantage to the IMM. Similarily to the CTCA model lost and missed tracks are caused by object initialization.

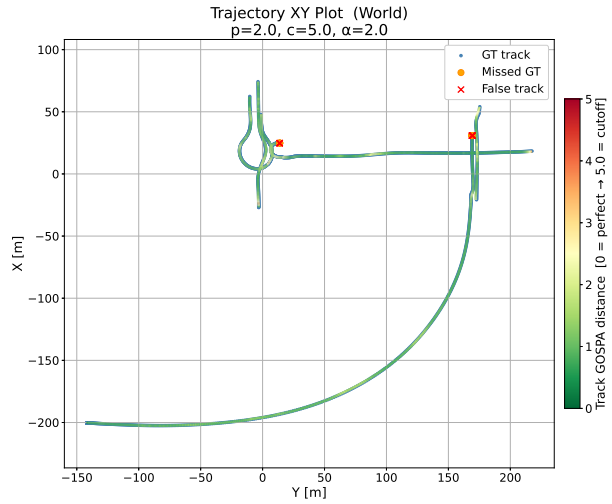


Figure 4.11: Test scenario P-GOSPA trajectory IMM in global coordinate frame.

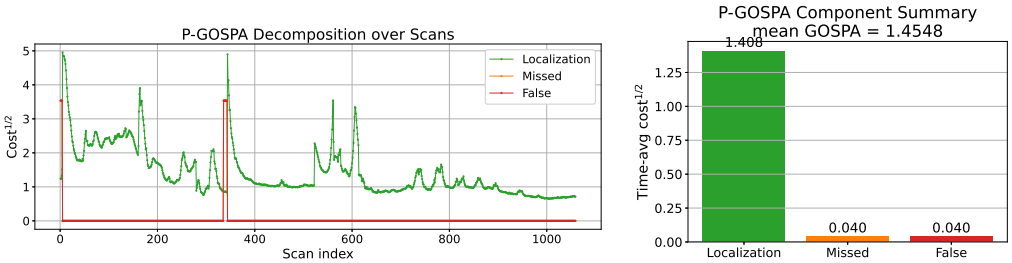


Figure 4.12: Test scenario P-GOSPA metrics using IMM.

IMM utilization To illustrate how the IMM filter use the available motion models during tracking, the evolution of the mode probabilities is examined for selected targets. The mode probabilities illustrate how the IMM distributes belief among its motion models over time and provide insight into how the chosen transition probability tuning influences the models and switching behavior.

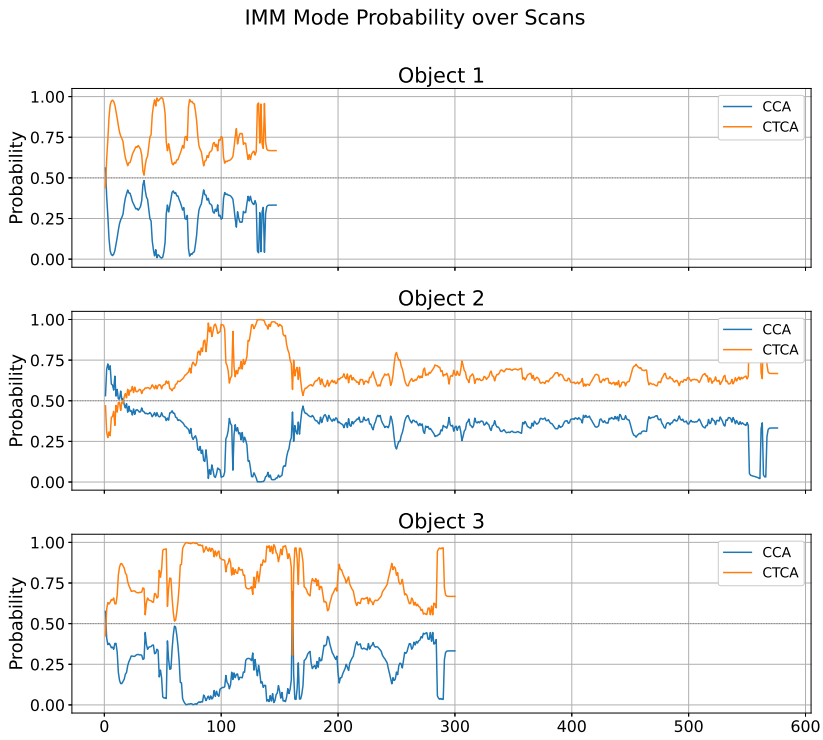


Figure 4.13: IMM mode probabilities for different targets over scans using scenario-trained tuning.

Figure 4.13 shows the mode probability time histories for the three first targets initiated when using transition probabilities tuned offline on a set of separate scenarios. For all targets, the CTCA model attains higher probability during large portions of the tracking interval. Variations in the relative probabilities of the CCA and CTCA models are observed, particularly during early scans and during periods where the probabilities of the two models are closer.

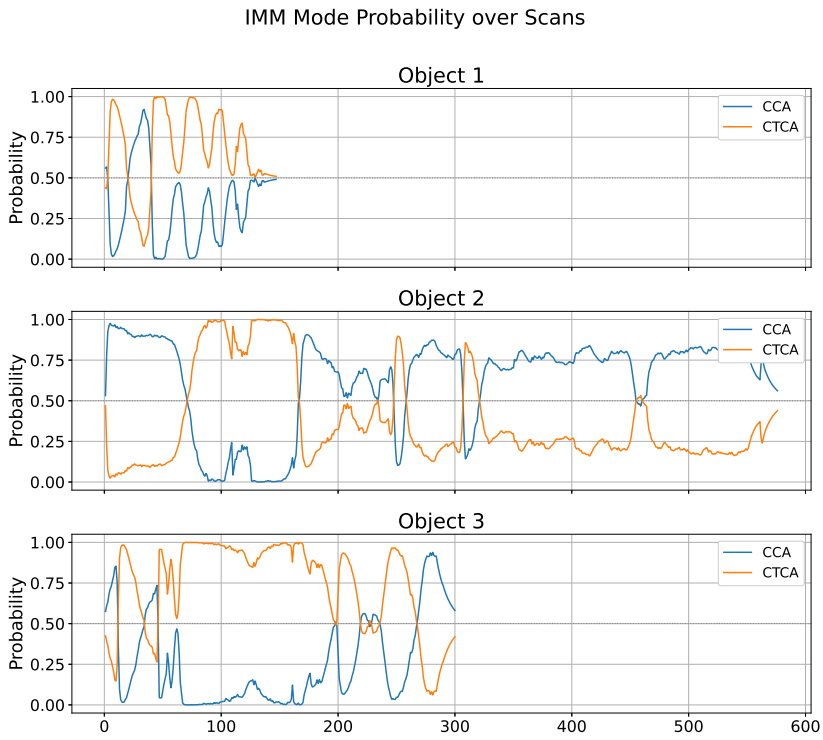


Figure 4.14: IMM mode probabilities for different targets over scans when manually tuned.

Figure 4.14 presents the corresponding mode probabilities when a manually tuned Markov transition matrix is used. The self-transition probabilities are set to $q_1 = q_2 = 0.95$, resulting in equal self-transition behavior for both motion models. In this configuration, the mode probabilities exhibit more frequent changes between the CCA and CTCA models across the target lifetimes. For some targets, extended intervals can be observed where the dominant mode alternates between scans. This results in a more adaptive switching strategy with the final P-GOSPA score of 1.53.

4.2 Motion Models

To evaluate tracker behaviour and performance, a set of test and verification scenarios was employed to obtain meaningful and comparable results across the different tracking methods. Since the performance of a Kalman filter is highly dependent on parameter tuning, as previously demonstrated, it is essential that each filter is tuned for the specific scenario under consideration. Otherwise, the evaluation would risk producing biased or misleading results. By applying the automated tuning method described in Section 3.6, a fair and consistent comparison between the different motion models can be achieved.

4.2.1 One Scenario Tuning - Optimally Tuned Models

For a given scenario, certain motion models are inherently better suited for tracking than others. When optimally tuned, these models are expected to yield a lower

P-GOSPA score relative to less suitable alternatives. By applying scenario-specific optimal tuning to a single scenario, the theoretically lowest achievable P-GOSPA score can be obtained, enabling an assessment of which motion model is most appropriate for that particular scenario. The objective of this evaluation is to isolate the effect of the motion model by minimizing the impact of suboptimal parameter tuning, thereby assessing the theoretical best case performance of each model for the given scenario. During this experiment the trajectories of each individual scenario are fixed, this means that the optimal tuning will be optimized against that fixed trajectory, different trajectories in the same scenario may give different results, but in these tests it is one trajectory per vehicle evaluated.

4.2.1.1 Test Scenario 1 – Stop and Go 2 Targets

The first scenario evaluated corresponds to one of the tuning scenarios from Section 3.6.3.1, the scenario in question is "Stop and Go 2 Targets". In theory, this scenario should favor a motion model that does not take heading into account, as the targets travel in straight lines at highway speeds. It is therefore expected that the CCA model will perform at least as well as the turn-capable alternatives, since explicit heading dynamics should provide little additional benefit in this case.

Parameter Tuning The CMA-ES tuning process was applied independently to each motion model for this scenario.
The tuning process converges to stable parameter values. For the CTCA model, as no turning behavior is present, the optimized process noise parameters suppress heading related dynamics, with the yaw-acceleration noise driven to a near-zero value. These trends are consistent with the tuned parameter values reported in Table 4.4, where the heading-related noise terms remain very small for the straight-line models used in this scenario. Parameter trajectory plots can be found in Appendix A.4.

Table 4.4: Process noise parameters for test scenario 1.

Motion model	q_1	q_2
CCA	0.1413	2.417×10^{-6}
CTCA	5.9078	1.597×10^{-5}
IMM	0.3040	4.03×10^{-2}

Tracking Performance The tracking performance obtained using the tuned parameters is presented below. Figure 4.15 shows a trajectory plot of the target vehicles using the CCA model. Due to the easily identifiable straight-line trajectories, the trajectory plots do not differ significantly across the three compared models and are therefore only shown for CCA.
The P-GOSPA results over time and the component breakdown for each model are shown in Figures 4.16–4.18. Across all three models, the score is dominated by localization error, while missed and false tracks are effectively absent. This indicates that the main challenge in this scenario is estimation accuracy rather than

track management. The small separation between the three curves further supports the expectation that explicit turn modeling is not critical for straight motion.

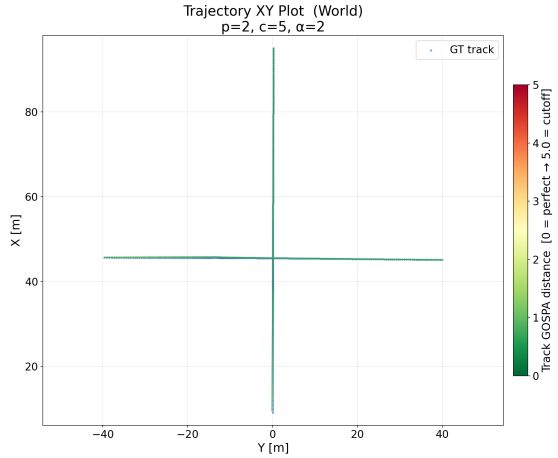


Figure 4.15: Target trajectory's over the scenario with P-GOSPA scores using CCA motion model. The trajectory plots for CTCA and IMM are visually similar and therefore omitted.

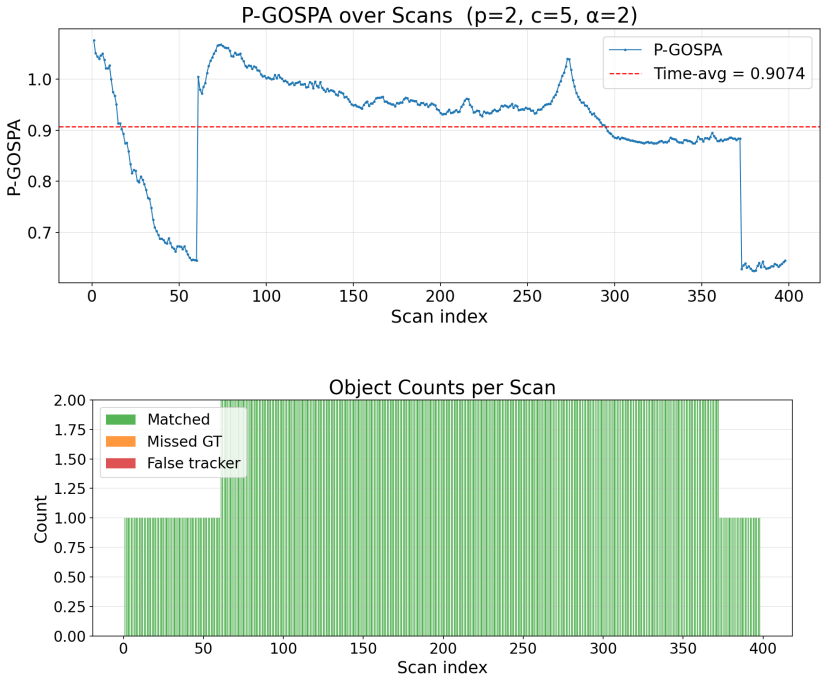


Figure 4.16: P-GOSPA results for test scenario 1 using the CCA motion model. The upper plot shows the P-GOSPA trajectory over time, the lower plot shows number of correct, missed and false targets.

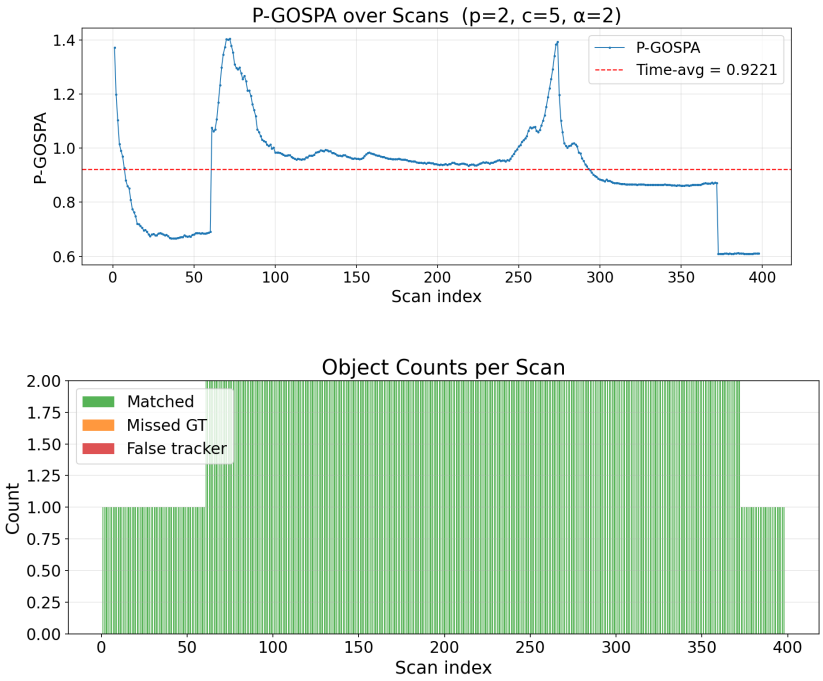


Figure 4.17: P-GOSPA results for test scenario 1 using the CTCA motion model. The upper plot shows the P-GOSPA trajectory over time, the lower plot shows number of correct, missed and false targets.

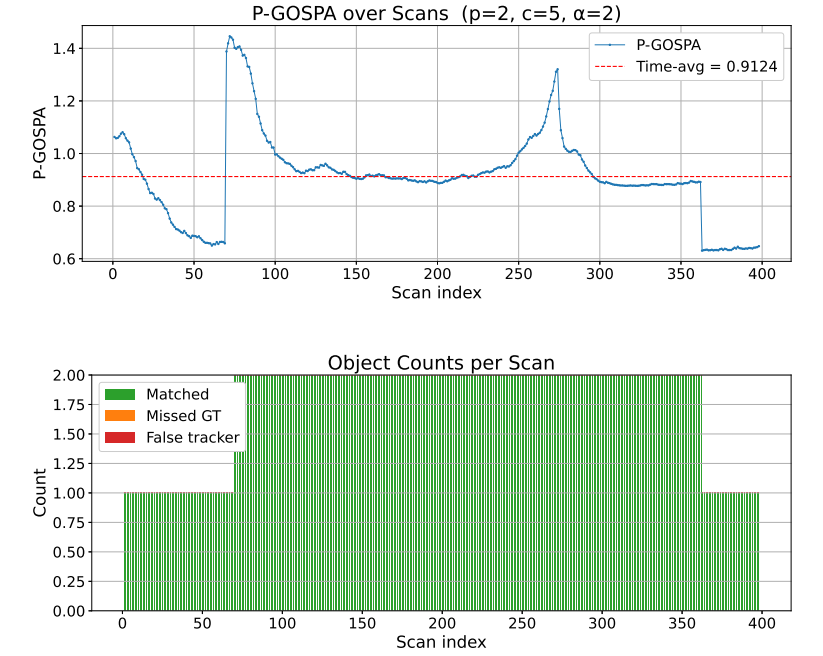


Figure 4.18: P-GOSPA results for test scenario 1 using the IMM filter. The upper plot shows the P-GOSPA trajectory over time, the lower plot shows number of correct, missed and false targets.

Summary Overall, all motion models and filters produce very similar results in this scenario. The P-GOSPA results indicate stable tracking performance throughout the scenario for all three models, with the localization component dominating the overall score. The absence of missed or false tracks across all models suggests that this straight-line highway scenario is well handled regardless of model choice.
Nevertheless, the ranking in Table 4.5 is still informative. The CCA model achieves the lowest score, the IMM filter follows closely, and the CTCA model performs slightly worse.

Table 4.5: Result of optimally tuned P-GOSPA performance and corresponding process noise parameters for test scenario 1.

Motion model	P-GOSPA score	q_1	q_2
CCA	0.9074	0.1413	2.417×10^{-6}
CTCA	0.9221	5.9078	1.597×10^{-5}
IMM	0.9124	0.3040	4.03×10^{-2}

4.2.1.2 Test Scenario 2 – Roundabout Constant Turn

Test Scenario 2 considers a single target traversing a roundabout trajectory with approximately constant speed and sustained curvature. The maneuver is characterized by a prolonged, smooth yaw-rate with only minor variations in longitudinal acceleration. This scenario represents a case for constant-turn motion and is therefore well suited for evaluating the benefits of motion models that explicitly include heading and yaw-rate dynamics.

Parameter Tuning The same automated tuning method is used here to identify each motion model's optimal score for this scenario.
The optimized parameters for the CCA model indicate an increased effective process noise compared to Scenario 1, reflecting the need to compensate for curvature through stochastic modeling rather than explicit turn dynamics. In contrast, the CTCA model converges to relatively low yaw-acceleration noise, consistent with the smooth and sustained turning behavior in this scenario. The IMM filter converges to stable parameters as well, indicating that a mixed-model formulation can exploit the constant-turn hypothesis without requiring the entire trajectory to be explained by a single model. The final tuning can be found in Table 4.6

Table 4.6: Process noise parameters for test scenario 2.

Motion model	q_1	q_2
CCA	0.6675	0.2275
CTCA	6.1080	0.1521
IMM	0.7597	0.0050

Tracking Performance The tracking performance for each model is shown in Figures 4.19–4.21. The two left-hand plots show the P-GOSPA statistics over time while the right plot shows the P-GOSPA trajectory over the scenario.

Comparing the single models in Figure 4.19 and Figure 4.20, it is clear that this scenario is much more sensitive to the choice of motion model. In particular, the CCA model exhibits a substantially higher score than both CTCA and IMM, showing that curvature cannot be represented sufficiently well through increased process noise alone. The CTCA and IMM results are nearly identical, which suggests that once turn dynamics are explicitly modeled, only a limited additional gain is obtained from model switching in this single-target maneuver.

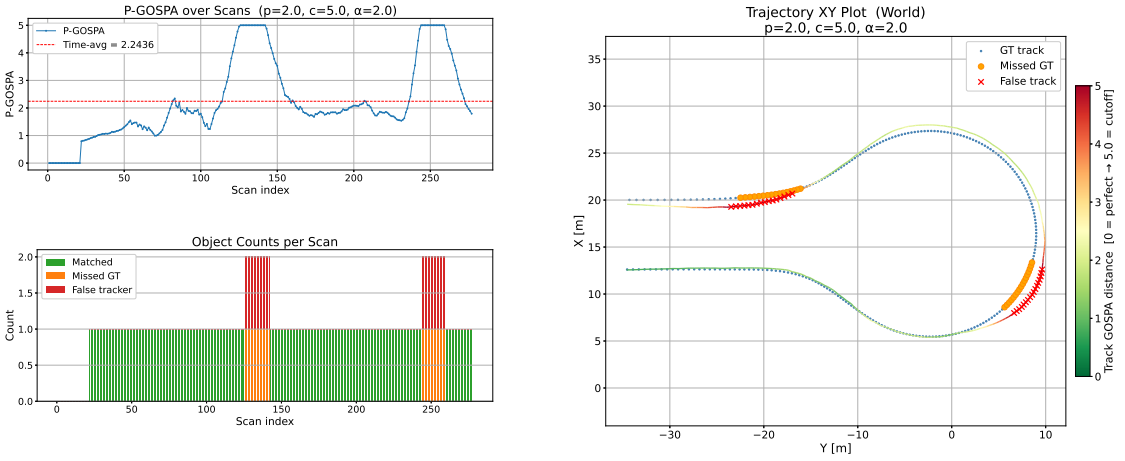


Figure 4.19: P-GOSPA results for test scenario 2 using the CCA motion model. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

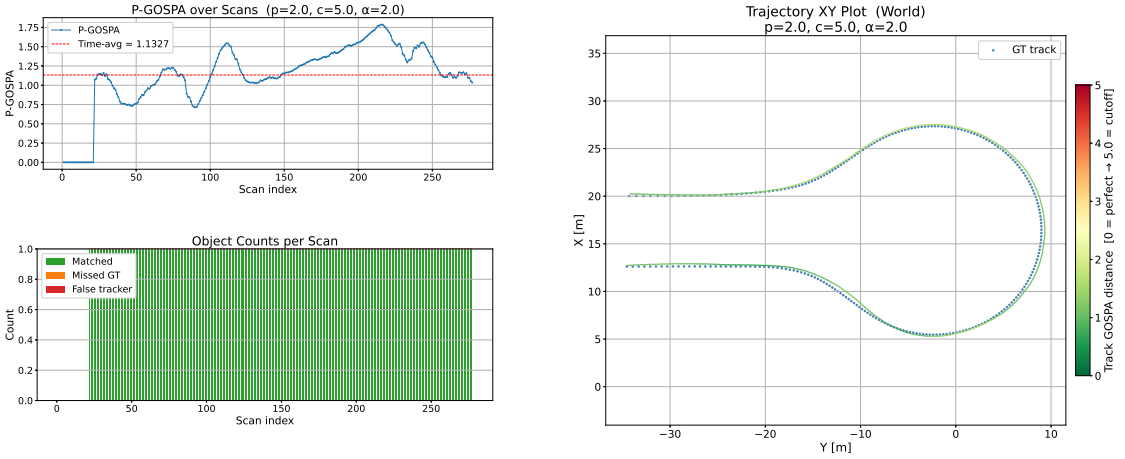


Figure 4.20: P-GOSPA results for test scenario 2 using the CTCA motion model. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

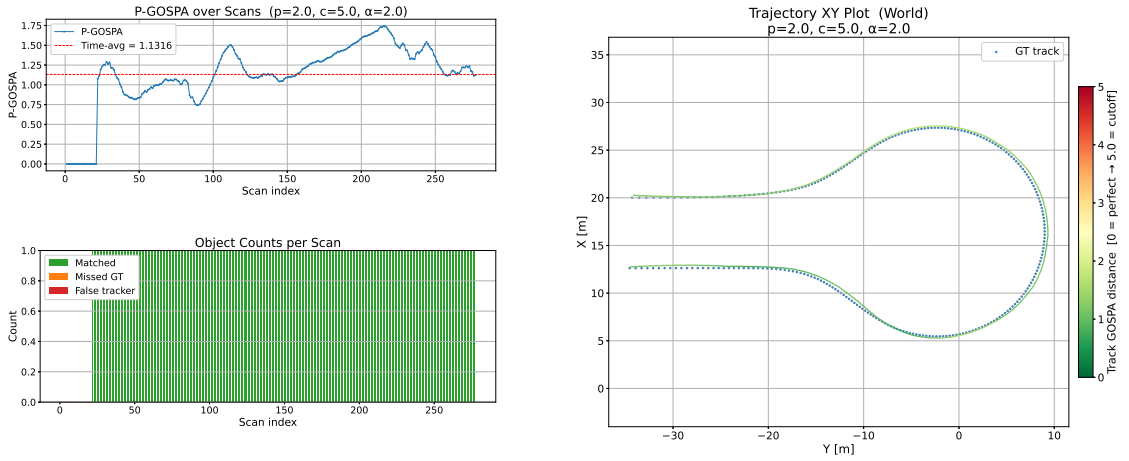


Figure 4.21: P-GOSPA results for test scenario 2 using the IMM filter. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

Summary For the constant roundabout maneuver, the results in Table 4.7 show a clear separation between the CCA model and the two turn-capable approaches. Both CTCA and IMM substantially outperform CCA, confirming that explicit turn dynamics are important for this type of motion. Although the IMM filter achieves the lowest overall score, the difference relative to CTCA is very small, so the main conclusion is not that IMM is dramatically better, but rather that turn-aware modeling is essential in this scenario. This stands in clear contrast to Scenario 1, where the simpler straight-line model was already sufficient.

Table 4.7: Result of optimally tuned P-GOSPA performance and corresponding process noise parameters for test scenario 2.

Motion model	P-GOSPA score	q_1	q_2
CCA	2.2436	0.6675	0.2275
CTCA	1.1327	6.1080	0.1521
IMM	1.1316	0.7597	0.0050

4.2.1.3 Test Scenario 3 – Roundabout Multiple Target Paths

The final scenario considered is a multi-target roundabout case in which the observed targets follow different motion patterns within the same scenario. This creates a more demanding evaluation setting than the previous two cases, since both straight and turning motion must be handled simultaneously. It is therefore expected that the IMM filter will be most suitable here, as neither a purely straight-line model nor a purely constant-turn model should be able to represent all target behaviors equally well.

Parameter Tuning This tuning step differs from the two previous scenarios because the target set contains multiple motion types. As a result, neither the CCA

nor the CTCA model can be expected to match the full scenario perfectly when tuned in isolation. Instead, each single-model tuning process must converge to a compromise between straight and curved motion. The IMM filter is better suited to this setting because its sub-models can represent different motion regimes, rather than forcing one model structure to explain the full scenario alone. The final tuning is seen in 4.8 and the resulting parameter trajectories can be found in Appendix A.4.

Table 4.8: Process noise parameters for test scenario 3.

Motion model	q_1	q_2
CCA	0.5609	0.2666
CTCA	71.8657	0.1636
IMM	0.9631	0.3807

Tracking Performance The tracking performance for each model is shown in Figures 4.22–4.24. As in Scenario 2, the left-hand plots show the P-GOSPA statistics over time, while the right-hand plots show the resulting target trajectories.

Compared with the previous scenarios, the separation between the models is more pronounced here. The CCA model yields a substantially higher overall P-GOSPA score, indicating that a single straight-line motion hypothesis is not sufficient for this mixed-motion scenario.

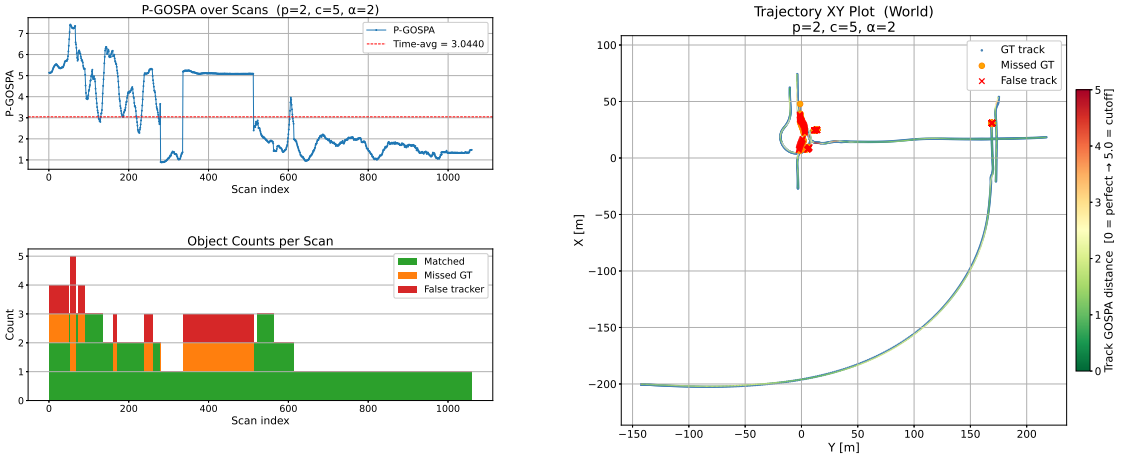


Figure 4.22: P-GOSPA results for test scenario 3 using the CCA motion model. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

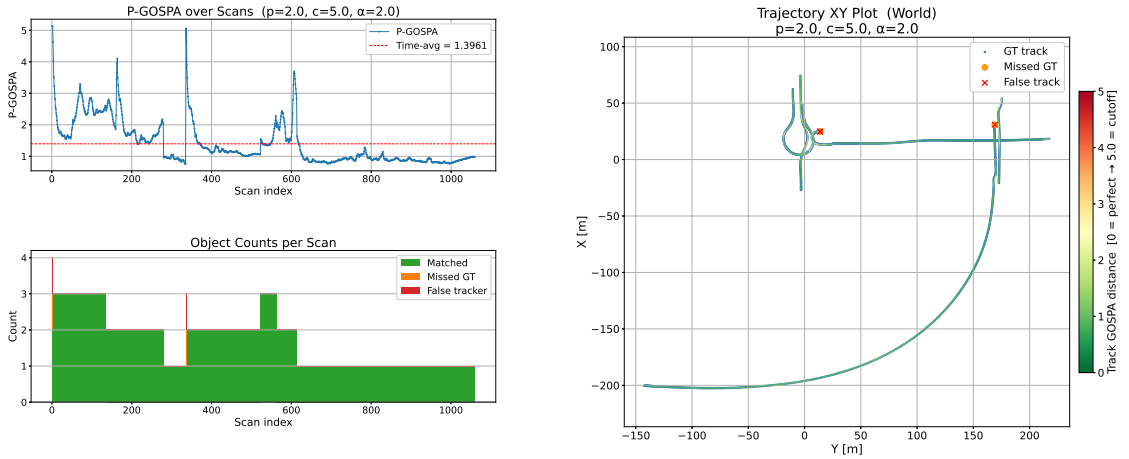


Figure 4.23: P-GOSPA results for test scenario 3 using the CTCA motion model. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

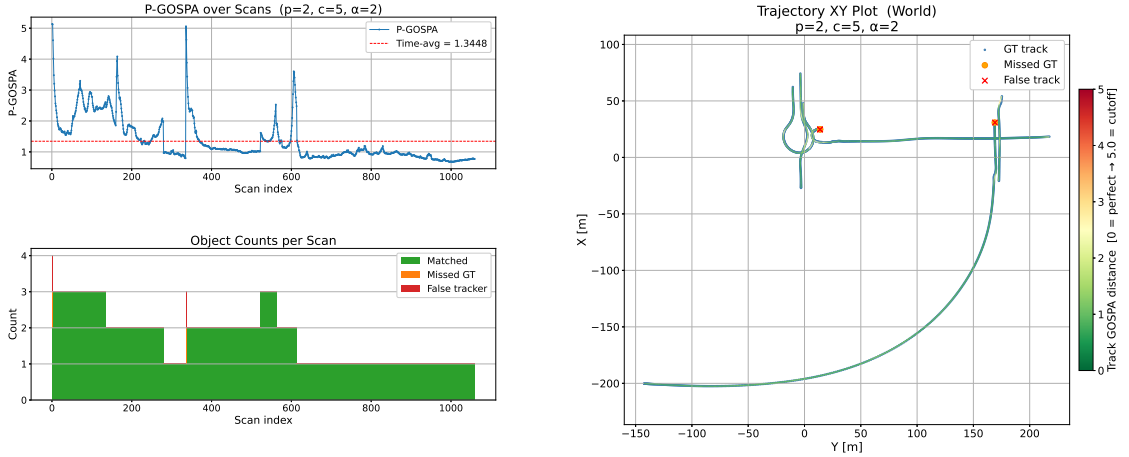


Figure 4.24: P-GOSPA results for test scenario 3 using the IMM filter. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

Summary The results in Table 4.9 show that the IMM filter provides the lowest overall P-GOSPA score in this scenario, closely followed by the CTCA model, while the CCA model achieves a significantly higher score. This ranking is consistent with the mixed-motion nature of the scenario.

Table 4.9: Result of optimally tuned P-GOSPA performance and corresponding process noise parameters for test scenario 3.

Motion model	P-GOSPA score	q_1	q_2
CCA	3.0440	0.5609	0.2666
CTCA	1.3961	71.8657	0.1636
IMM	1.3448	0.9631	0.3807

4.2.1.4 Cross-Scenario Comparison

The cross-scenario comparison summarizes the tracking performance of the optimally tuned CCA, CTCA, and IMM configurations across all evaluated scenarios using the P-GOSPA metric and its constituent components.

Across the evaluated scenario set, clear differences in failure modes are observed between the single-model filters. The CCA motion model exhibits acceptable performance in scenarios dominated by straight-line or near-constant velocity motion. However, once sustained turning or constant curvature is introduced, CCA performance degrades sharply. In these cases, the increase in P-GOSPA is caused not only by elevated localization error but also by frequent target loss, leading to increased missed-target penalties. Track fragmentation and re-initialization events are observed repeatedly in turning scenarios, indicating a reduced ability of the CCA model to maintain target existence under such motion conditions.

The CTCA motion model achieves substantially lower P-GOSPA scores in scenarios characterized by sustained turning motion, such as roundabouts and curved trajectories. In these scenarios, improvements are observed primarily in the localization component, while missed-target penalties remain low, indicating that tracks are maintained continuously throughout turning maneuvers. In scenarios where the target motion remains approximately straight, CTCA performance is comparable to CCA, with no systematic increase in missed or false tracks.

Across the full scenario set, the variability in CCA performance is dominated by turning-induced failure cases in which target loss contributes significantly to the overall P-GOSPA score. In contrast, CTCA performance remains bounded across all evaluated scenarios, with reduced sensitivity to motion curvature and fewer track discontinuities.

The IMM filter demonstrates the most uniform behavior across scenarios. In cases where one of the single motion models degrades, the IMM avoids the corresponding failure mode and maintains continuous tracks, resulting in lower missed-target costs and reduced P-GOSPA variability across the scenario set. Overall, the cross-scenario results show clear differences in robustness between the evaluated motion model configurations.

4.2.2 Single Motion Model IMM Filter

As an additional control experiment, an alternative IMM configuration was evaluated in which multiple instances of the same motion model were used with different covariance tunings instead of heterogeneous motion models such as CCA and CTCA. This setup will be called Single Motion Model IMM (SMM-IMM). In Section 4.2.1.1 it can be seen that the CTCA model, with well tuned covariance parameters, is able to describe the same motion regime as the CCA model in a close proximity. Using multiple CTCA models with different tunings allows the filter to represent different levels of motion aggressiveness, ranging from low-curvature to high-curvature motion regimes. This configuration also avoids state-space transformations between heterogeneous motion models.

4.2.2.1 SMM-IMM Filter Setup

The underlying IMM formulation and filtering procedure remain unchanged, the configuration differs only in the number of models and their respective covariance tunings.
The transition matrix is

$$\mathbf{\Pi} = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}. \tag{4.1}$$

Model	$\sigma_{\dot{a}}^2$	$\sigma_{\dot{\omega}}^2$
Model 1	10.0	0.01
Model 2	50.8657	0.1636
Model 3	150.0	1.0

Table 4.10: Tuning parameters for the three CTCA models.

The tuning parameters can be seen in Table 4.10. The tuning parameters were not optimized using CMA-ES. Instead, the optimally tuned CTCA configuration from Section 4.2.1.3 was used as a reference to manually construct three distinct tuning regimes.

4.2.2.2 SMM-IMM Filter Performance

This IMM configuration was evaluated on Scenario 3 (Section 4.2.1.3) using the same evaluation methodology and KPIs as in the previous experiments.

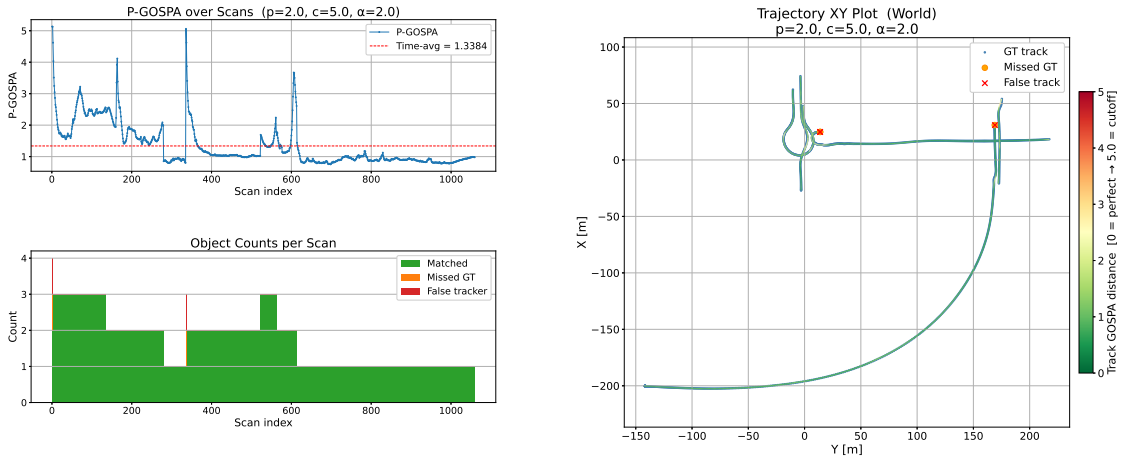


Figure 4.25: P-GOSPA results for test scenario 3 using the single motion model IMM filter. The left column shows the P-GOSPA statistics, while the right plot shows the trajectory results.

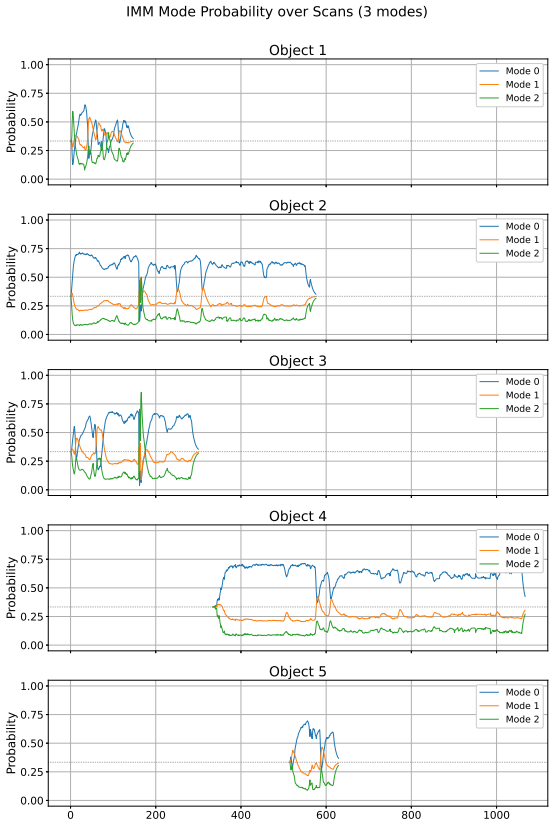


Figure 4.26: IMM mode probabilities over scans.

As shown in Figure 4.25, the overall tracking performance of this configuration is comparable to that of the traditional IMM. For Scenario 3, the single motion model IMM achieves an overall P-GOSPA score of 1.3384, which is slightly lower than the score obtained with the traditional IMM configuration (1.3448). The corresponding mode probability evolution is shown in Figure 4.26.

5
Conclusion

5.1 P-GOSPA

The discovery of P-GOSPA:s well behaved nature in a MOT setting across a large number of scenarios is the main reason the work in the end leaned towards automating the covariance tuning. A key property that makes P-GOSPA suitable for automated optimization is its decomposability and smooth behavior with respect to changes in filter parameters. Although the underlying EKF-based MOT pipeline is highly nonlinear, P-GOSPA tends to vary gradually as covariance parameters are adjusted when aggregated over time and across scenarios. This results in an objective function that while not convex exhibits meaningful structure as shown in the grid based proof of concept evaluation.

Another important advantage of P-GOSPA is its flexibility. The metric allows tuning such as cut-off distances, cardinality penalties, and the weighting between localization and association errors. Further the distance function itself can be tailored to include only the relevant components of the object state along with state uncertainties. In Section 3.5, the effect of excluding state uncertainties from the distance function was briefly evaluated. In that case, the metric reduces to regular GOSPA, where the GT and estimated object states are represented as deterministic finite sets, rather than as MB set densities as in P-GOSPA. This comparison highlights that on the one hand, P-GOSPA can incorporate uncertainty information and is therefore more expressive when both the estimates and the reference data are uncertain. On the other hand, its performance depends on the quality and calibration of the estimated covariance matrices. If these covariances are not fully comparable across motion models, or if they do not accurately reflect the true estimation uncertainty, P-GOSPA may penalize differences in covariance modeling rather than differences in tracking performance alone. In such cases, regular GOSPA may provide a fairer measure of pure MOT performance, since it evaluates only the estimated states without including uncertainty information.

The main motivation for using P-GOSPA in the later tracker evaluations was its relevance to future work involving real data. In this work only simulated data with known GT were evaluated, however when using real measurements, GT would likely need to be obtained through another estimation process, for example using LiDAR-based annotations or sensor fusion which would itself introduce uncertainty. In such a setting, representing the GT as a Dirac density may be overly restrictive, whereas P-GOSPA provides a principled way to account for uncertainty in both the GT and the estimated tracks. Therefore, while regular GOSPA may be preferable when the

objective is to isolate deterministic tracking accuracy, P-GOSPA becomes attractive when uncertainty in the state representation is itself relevant to the evaluation.

5.2 CMA-ES

This work demonstrates that CMA-ES is a practical and effective method for automatic tuning of EKF parameters where the objective function is considered unknown. By formulating the tuning problem as a black-box optimization task and utilizing P-GOSPA as a loss function, CMA-ES is observed to successfully navigate the optimization landscape. When compared against a grid search over process noise parameters CMA-ES consistently converges to the same region of the parameter space.
The final tuning procedure further illustrates the scalability and flexibility of the approach. By tuning the CCA and CTCA motion models independently on scenario sets tailored to their respective strengths, and optimizing the IMM transition probabilities with both models combined the methodology avoids overfitting to a single motion pattern. The resulting tuned parameters lead to a clear performance hierarchy in the unseen test scenario where the CCA model underperforms in curved motion, the CTCA model handles a broader class of trajectories, and the IMM achieves the best overall P-GOSPA score by adaptively blending the two models.
Although the true perfect tuning of the different motion models is unknown, the methodology used and results achieved suggests that the overall objective to find a covariance tuning achieving good results given the scenario sets, motion models and evaluation method is met.

5.3 Motion Models

This thesis investigated the performance of different motion models within an EKF based automotive radar tracking framework by evaluating them across a diverse set of driving scenarios under near optimal tuning conditions. The results demonstrate that motion model choice has a decisive impact on tracking robustness, particularly in scenarios involving constant turning motion.
The CCA model exhibits acceptable performance in scenarios dominated by straight-line motion or velocity changes. In such cases, the model is able to maintain target tracks with moderate localization accuracy. However, once sustained turning motion is introduced, the CCA model consistently fails to preserve track continuity. This degradation manifests not only as increased localization error but, more critically, as repeated target loss and re-initialization. While increasing the P-GOSPA cut-off distance would reduce the magnitude of the corresponding missed-target penalties, it does not eliminate the underlying issue, as the tracker loses association between the track and incoming measurements. These failures lead to large missed-target penalties in P-GOSPA and dominate the overall tracking performance. The observed behavior indicates that the CCA model is structurally unable to represent constant turn dynamics, regardless of tuning quality.
In contrast, CTCA motion model demonstrates robust tracking behavior in scenarios

characterized by curved trajectories and sustained yaw-rate motion. In these cases, CTCA maintains continuous tracks and significantly reduces localization error compared to CCA. Across all evaluated turning scenarios, CTCA avoids the systematic trajectory deviations and track losses observed with the Cartesian model. In scenarios with approximately straight motion, CTCA performs comparably to CCA, however the inclusion of curvilinear dynamics introduces a slight model mismatch that can result in marginally reduced accuracy under purely linear motion.

The cross-scenario analysis further shows that differences between the motion models cannot be attributed solely to covariance tuning. By enforcing automated, scenario-specific near-optimal tuning, the comparison isolates the effect of model structure rather than parameter choice. Even under optimal tuning conditions, the CCA model consistently breaks down in turning scenarios, while CTCA maintains bounded error and continuous target existence. This confirms that the observed performance differences arise from fundamental modeling assumptions rather than sub-optimal parametrization.

The results must also be interpreted in relation to the assumed target class and sensing conditions, particularly the measurement rate and sensor noise characteristics. Although these factors were not explicitly varied in this study, they are known to influence the relative importance of the motion model, as mentioned in Section 2.3. For lower update rates, the motion between consecutive measurements becomes more nonlinear, which increases the reliance on the motion model for accurate prediction. Under such conditions, simplified models such as CCA can be expected to be more prone to track loss during turning motion. Conversely, for higher update rates or lower measurement noise, the influence of the motion model is reduced, and simpler models may still yield acceptable performance.

While CTCA provides clear advantages for vehicle tracking due to its ability to represent curvilinear motion, its benefits may diminish for targets with less structured dynamics, such as pedestrians or cyclists. In these cases, the effectiveness of the CCA model may increase due to its flexibility and reduced reliance on well-defined heading and yaw-rate states. Furthermore, variations in vehicle characteristics, such as wheelbase, articulated trailers, or rear-wheel steering, may introduce modeling errors that reduce the validity of the CTCA assumptions.

This highlights that the suitability of a motion model is not absolute, but inherently coupled to both the target dynamics and the sensing configuration.

These findings highlight the complementary nature of the evaluated motion models. The CCA model offers simplicity and computational efficiency but lacks the ability to represent curvilinear vehicular motion. The CTCA model accurately captures such dynamics but relies on reliable observability of heading and yaw-rate states to achieve its performance benefits. No single motion model was found to provide uniformly robust performance across all evaluated scenarios.

Overall, the results confirm that motion modeling assumptions play a central role in tracking performance for automotive radar systems. The pronounced failure of the CCA model in turning scenarios motivates the use of adaptive multi-model approaches, which are further evaluated through the IMM framework in this thesis.

5.4 IMM

The IMM framework was evaluated in this thesis as a means of combining complementary motion models to improve overall tracking robustness and predictive capability. By maintaining multiple motion hypotheses in parallel and weighting them according to their measurement likelihoods, the IMM provides a principled statistical framework for handling targets whose motion does not adhere to a single, fixed dynamic model. The central hypothesis investigated in this work was that, under optimal tuning, an IMM filter should perform at least as well as its constituent single-model trackers, and potentially outperform them in scenarios involving changes in target dynamics.

The results presented in Section 4.2 reveal several important insights. While the IMM demonstrates clear benefits in scenarios with mixed or changing motion behaviour, it cannot be regarded as a strictly loss-free extension of the single-model filters. In particular, for Scenario 1, the IMM was observed to perform slightly worse than the best single motion model according to the selected KPIs. This finding contradicts the common assumption that IMM filtering will always yield equal or superior performance compared to its individual components.

Several factors may explain this behaviour. One contributing aspect lies in the design choices required when combining models with different state representations. In the implemented IMM, heading and yaw-rate updates are primarily taken from the CTCA branch, while the CCA branch does not explicitly estimate these states. In scenarios dominated by straight-line motion, this can introduce unnecessary uncertainty or reduced accuracy when the curvilinear model contributes information that is not fully consistent with the dominant dynamics. Additionally, the probabilistic mixing process itself can dilute the state estimate when model probabilities are similar, even if one model is clearly more appropriate for the motion at hand.

Despite these limitations, the IMM shows clear advantages in scenarios characterized by transitions between distinct motion regimes. In such cases, the IMM consistently provides more stable tracking and improved robustness compared to relying on a single motion model. The mode probability evolution further indicates that the IMM correctly identifies and adapts to changes in target behaviour, assigning higher probability to the CTCA model during constant turns and favouring the CCA model during straight or low-curvature motion. This behaviour confirms that the IMM effectively exploits the complementary strengths of the included motion models.

Another important observation is that the performance of the IMM is highly dependent on tuning quality. Without well-optimized process noise parameters and transition probabilities, the theoretical advantages of the IMM can be diminished or even negated. The use of automated tuning via CMA-ES was therefore essential for enabling a fair comparison and for allowing the IMM to operate close to its optimal performance.

The control experiment using a single motion model with multiple tunings demonstrates that parameter diversity alone can achieve the robustness observed with the traditional IMM configurations. This indicates that part of the IMM benefit originates from adaptive selection among different levels of motion responsiveness, rather than exclusively from structural differences between motion models. At the same

time, the results suggest that further improvements may be achievable by extending the IMM framework with additional model instances tuned for specific motion regimes. This enables the use of the IMM filter as an adaptive parameter switching function.
In summary, the IMM framework provides meaningful benefits in tracking scenarios involving monotone or non-stationary target dynamics, offering increased robustness and adaptivity compared to single-model trackers. However, in scenarios with a single, well defined motion pattern, an optimally tuned single motion model may achieve comparable or better performance at lower computational and implementation complexity. These findings highlight that the IMM should be regarded as a useful but not universally superior solution, whose applicability must be considered in relation to scenario characteristics and system design constraints.

5.5 Future work

The following subsections cover some of the ideas that arose from the thesis work that weren't explored.

5.5.1 P-GOSPA

The use of P-GOSPA as a key performance indicator in this thesis could be explored further to better reflect what meaningful multi-object tracking performance actually means. Defining an appropriate KPI is inherently challenging, as the notion of "good tracking" is application dependent and somewhat subjective.
One aspect not explicitly considered in this work is the relative importance of objects with respect to the host vehicle. In practical ADAS applications, objects that are close to or approaching the host are significantly more critical than distant objects. The current formulation of P-GOSPA treats all objects equally, which may overemphasize errors associated with distant targets that are frequently initialized and lost near the detection range limit. Introducing a distance weighting in the localization cost could therefore prioritize performance on safety critical objects and yield a more representative evaluation metric.
Furthermore, preprocessing the GT data to exclude tracks that are not feasible for a tracker to follow such as objects that are occluded for extended periods or lie outside the perception range could improve the evaluation. This would allow P-GOSPA to assess only those objects that a tracker could reasonably be expected to maintain and not penalize track losses that might be expected, thereby yielding a score that more accurately reflects meaningful tracking performance
Overall designing a KPI that truly reflects MOT performance is nontrivial and more application-driven metrics are likely needed for meaningful evaluation.

5.5.2 Diversity of valuated objects

The results and conclusions drawn in this thesis are to some extent biased towards the CTCA motion model due to the narrow scope of only evaluating front-wheel-

driven passenger vehicles. While such vehicles constitute a large portion of real-world traffic, their motion can often be well approximated by smooth, curved trajectories. To further strengthen the validity and generality of the conclusions, future work should extend the evaluation to a broader set of object classes exhibiting different motion characteristics. For example cars with trailers, pedestrians, cyclists, motorcycles, trucks and buses. These object types often display motion patterns that deviate significantly from the assumptions underlying the CTCA model.
Expanding the diversity of evaluated objects would also provide a more realistic testbed for the IMM framework. The benefits of multi-model approaches are expected to become more pronounced when no single motion model can adequately describe all object types. By including a wider range of motion behaviors, future studies can better assess the ability of IMM-based systems to adaptively select or combine motion hypotheses in response to varying dynamics.

5.5.3 CMA-ES

It was chosen to limit the scope of tuning to only two parameters of the process noise covariance matrix for each motion model due to the existing state-dependent structure of the covariance matrices for both the CCA and CTCA models. However CMA-ES is a general optimization method and could in principle be used to tune the full covariance matrix without relying on the analytical model structure.
While this could potentially improve performance by better matching the covariance to the underlying data, it comes with a trade-off. For the CTCA model in particular the theoretically derived covariance matrix captures important dependencies on velocity, heading, and yaw-rate. Replacing this structure with a fully static covariance matrix would remove these physically meaningful relationships and instead rely purely on empirical fitting.
This raises an interesting question from an engineering perspective: whether a fully data-driven covariance matrix can outperform a physically derived one, or if the loss of structure leads to poorer generalization across scenarios. A hybrid approach could also be considered, where the analytical structure is retained but augmented with additional tunable scaling factors or cross-covariance terms. Exploring these alternatives could provide further insight into the balance between model-based design and data-driven optimization in tracking filter tuning

5.5.4 IMM

Brief exploration of an IMM pipeline using multiple instances of the same motion model with different process noise covariance tunings was conducted in section 4.2.2.2 which showed promising results. Instead of combining fundamentally different motion models, this configuration effectively creates a dynamically tuned model that can switch between different noise assumptions depending on the current motion.
The idea is similar to manually tuning process noise as a function of operating conditions (e.g., velocity or curvature), but handled implicitly through the IMM framework. Each model instance represents a different tuning hypothesis and the

filter adaptively weights them based on measurement consistency. This allows the system to respond to changing dynamics without explicitly defining state dependent tuning rules.
From an implementation perspective this approach is also attractive. Since all models share the same state representation there is no need for transformations between different state spaces, and the mixing step becomes significantly simpler. This avoids the approximation errors and added complexity introduced by nonlinear state mappings as required when combining structurally different motion models.
However while this improves adaptability to varying dynamics, it does not introduce new motion hypotheses, meaning that model mismatch can still occur if none of the predefined tunings represent the true motion well enough.

Bibliography

[1] Alper Akca and M. Önder Efe. "Multiple Model Kalman and Particle Filters and Applications: A Survey". In: *IFAC-PapersOnLine* 52.3 (2019). 15th IFAC Symposium on Large Scale Complex Systems LSS 2019, pp. 73–78. ISSN: 2405-8963. URL: https://www.sciencedirect.com/science/article/pii/S2405896319300977.

[2] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, 2001. ISBN: 978-0471416555.

[3] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.10 (2012), pp. 281–305. URL: http://jmlr.org/papers/v13/bergstra12a.html.

[4] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.

[5] W. Dale Blair, Mark A. Richards, and David A. Long. "Radar Measurements". In: *Principles of Modern Radar: Basic Principles*. Ed. by Mark A. Richards, James A. Scheer, and William A. Holm. Raleigh, NC, USA: SciTech Publishing / Institution of Engineering and Technology, 2010. Chap. 18, pp. 1–36. ISBN: 978-1-891121-52-4.

[6] Henk A.P. Blom and Yaakov Bar-Shalom. "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients". In: (1988). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0024057021&doi=10.1109%2f9.1299&partnerID=40&md5=8d6555da2e8dc784bc7bda86c2b86881.

[7] Zhaozhong Chen et al. "Kalman Filter Tuning with Bayesian Optimization". In: *arXiv preprint arXiv:1912.08601* (2019).

[8] Zdeněk Dostál et al. "Cholesky decomposition of a positive semidefinite matrix with known kernel". In: *Applied Mathematics and Computation* 217.13 (2011), pp. 6067–6077. ISSN: 0096-3003. URL: https://www.sciencedirect.com/science/article/pii/S0096300310012713.

[9] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*. 2023. arXiv: 1604.00772 [cs.LG]. URL: https://arxiv.org/abs/1604.00772.

[10] R.E. Kalman. “A new approach to linear filtering and prediction problems”. In: *Journal of Fluids Engineering, Transactions of the ASME* 82.1 (1960), pp. 35–45. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85024429815&doi=10.1115%2f1.3662552&partnerID=40&md5=ca5123b460e8fa10bf95b6bc29e17c66.

[11] Ehsan Majma et al. “The Interactive Multiple Model Strategy With Matrix Form Mode Probability for Models With Different State Vectors”. In: *IEEE Access* 13 (2025), pp. 96635–96645. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105007298698&doi=10.1109%2fACCESS.2025.3575641&partnerID=40&md5=5e340816f8886f0e6d0550facfd520a5.

[12] MathWorks. *Driving Scenario Designer App*. https://se.mathworks.com/help/driving/ref/drivingscenariodesigner-app.html. Accessed: 2026-02-13.

[13] MathWorks. *Radar Data Generator*. https://se.mathworks.com/help/radar/ref/radardatagenerator.html. Accessed: 2026-02-13.

[14] Anton Ödqvis Niclas Carlström. “Target Identification Using Low Level Radar Measurements”. MA thesis. Chalmers University of Technology, 2018. URL: https://odr.chalmers.se/server/api/core/bitstreams/0ec7bab5-d67f-459e-a248-f355f1c7e9ea/content.

[15] Michael Parker. “Chapter 20 - Automotive RadarWith contributions by Ben Esposito.” In: *Digital Signal Processing 101 (Second Edition)*. Second Edition. Newnes, 2017, pp. 253–276. ISBN: 978-0-12-811453-7. DOI: https://doi.org/10.1016/B978-0-12-811453-7.00020-2. URL: https://www.sciencedirect.com/science/article/pii/B9780128114537000202.

[16] Sujeet Milind Patole et al. “Automotive Radars: A Review of Signal Processing Techniques”. In: *IEEE Signal Processing Magazine* 34.2 (2017), pp. 22–35. DOI: 10.1109/MSP.2016.2628914.

[17] Nikola Petrović, Milena Petrović, and Vladimir Milovanović. “Radar Signal Processing Architecture for Early Detection of Automotive Obstacles”. In: *Electronics* 12.8 (2023). DOI: 10.3390/electronics12081826.

[18] X. Rong Li and V.P. Jilkov. “Survey of maneuvering target tracking. Part I. Dynamic models”. In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003), pp. 1333–1364. DOI: 10.1109/TAES.2003.1261132.

[19] X. Rong Li and V.P. Jilkov. “Survey of maneuvering target tracking. Part V. Multiple-model methods”. In: *IEEE Transactions on Aerospace and Electronic Systems* 41.4 (2005), pp. 1255–1321. DOI: 10.1109/TAES.2005.1561886.

[20] Simo Särkkä and Lennart Svensson. *Bayesian Filtering and Smoothing*. 2nd ed. Cambridge University Press, 2023.

[21] Robin Schubert, Eric Richter, and Gerd Wanielik. “Comparison and evaluation of advanced motion models for vehicle tracking”. In: *2008 11th international conference on information fusion*. IEEE. 2008, pp. 1–6.

[22] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ: John Wiley & Sons, 2006.

[23] Daniel Svensson. “Derivation of the discrete-time constant turn rate and acceleration motion model”. In: *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. 2019, pp. 1–5. DOI: 10.1109/SDF.2019.8916654.

[24] Yuxuan Xia et al. “Probabilistic GOSPA: A Metric for Performance Evaluation of Multiobject Filters With Uncertainties”. In: *IEEE Transactions on Aerospace and Electronic Systems* 61.5 (2025), pp. 15113–15121. DOI: 10.1109/TAES.2025.3580523.

A

Appendix

A.1 Benchmarking Scenario Trajectories

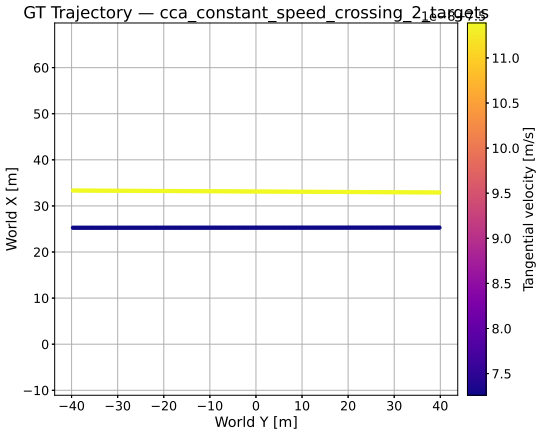


Figure A.1: Constant speed crossing 2 targets

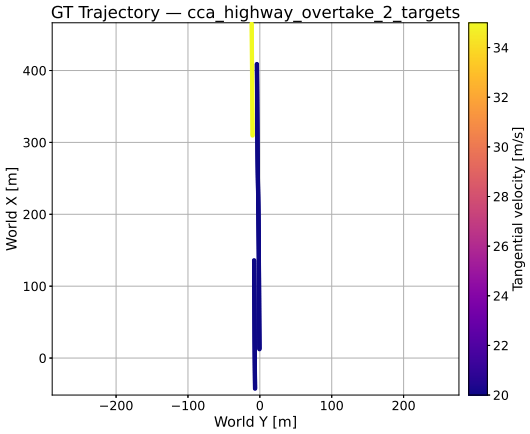


Figure A.2: Highway overtake 2 targets

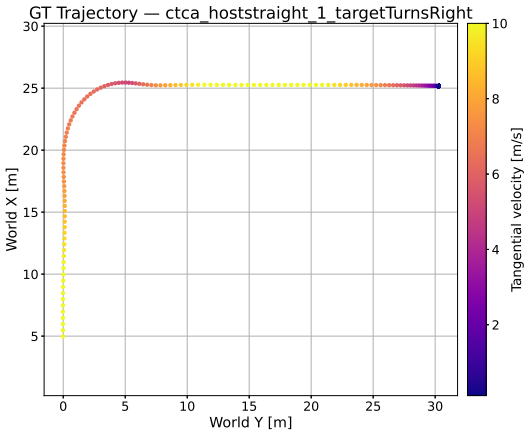


Figure A.3: Hoststraight 1 target turns right

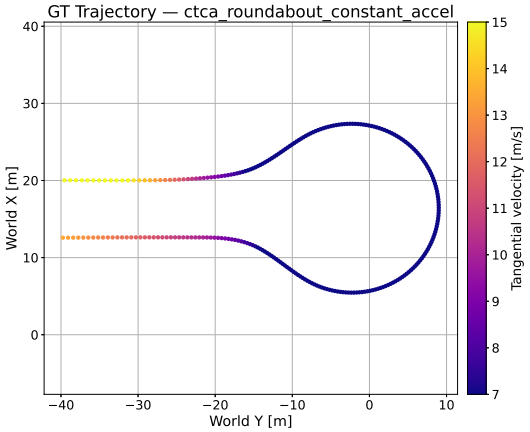


Figure A.4: Roundabout constant acceleration

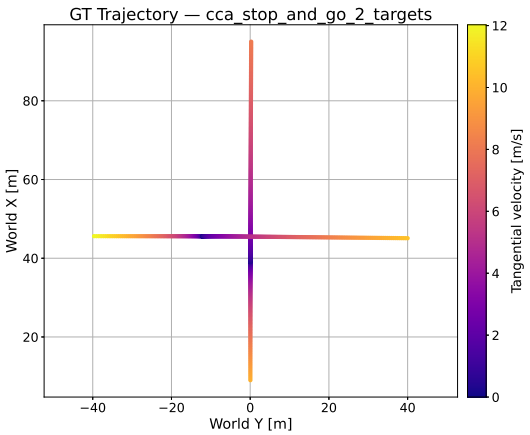


Figure A.5: Stop and go 2 targets

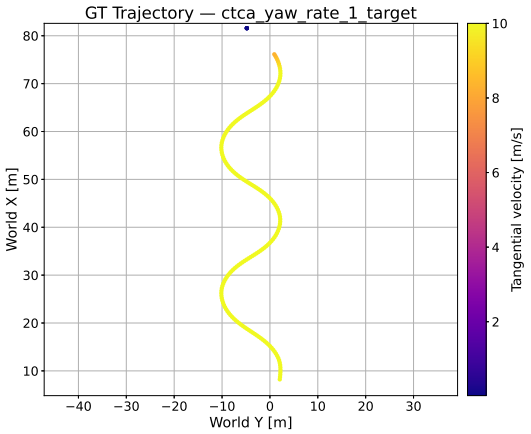


Figure A.6: Yaw rate 1 target

A.2 Final Tuning Scenario Trajectories

A.2.1 CCA

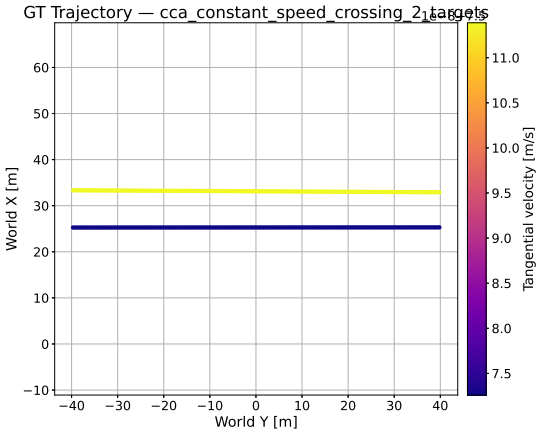


Figure A.7: Constant speed crossing 2 targets

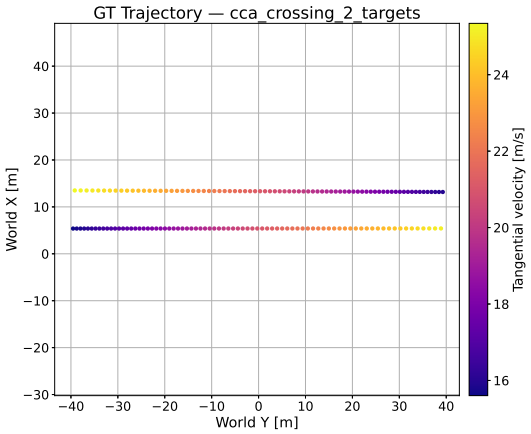


Figure A.8: Crossing 2 targets

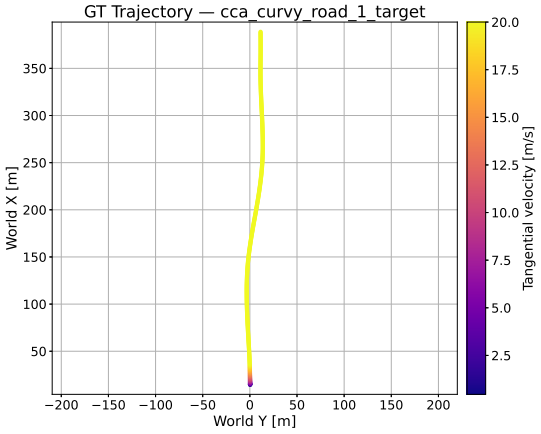


Figure A.9: Curvy road 1 target

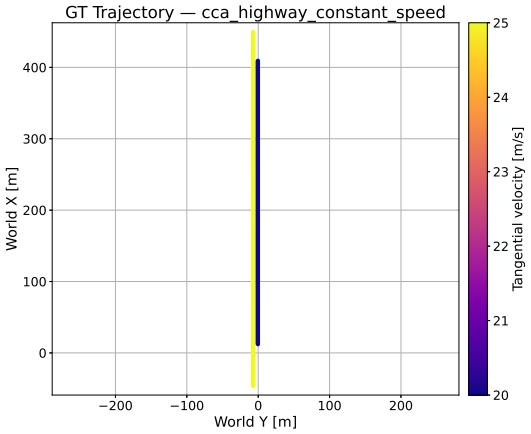


Figure A.10: Highway constant speed

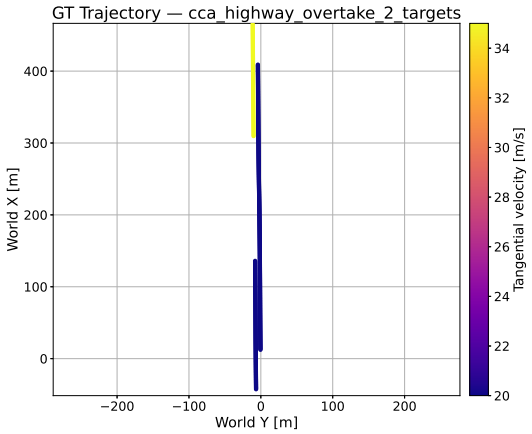


Figure A.11: Highway overtake 2 targets

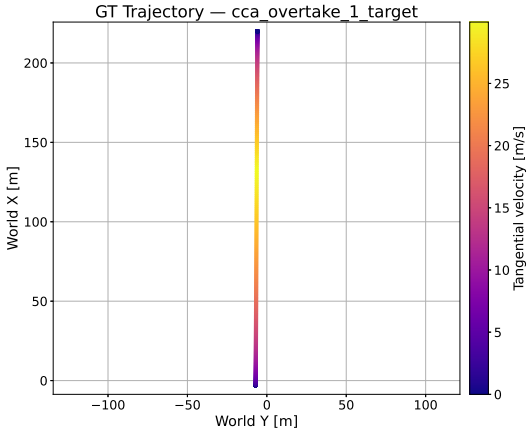


Figure A.12: Highway overtake 2 targets

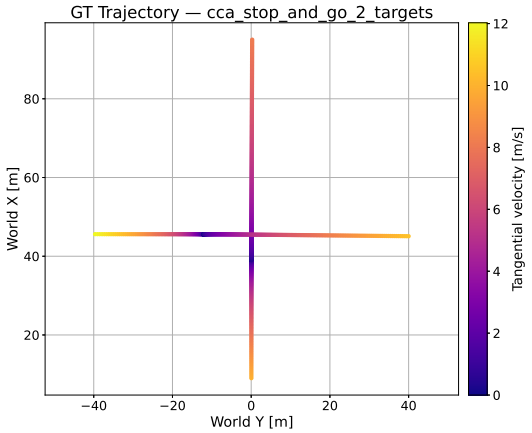


Figure A.13: Stop and go 2 targets

A.2.2 CTCA

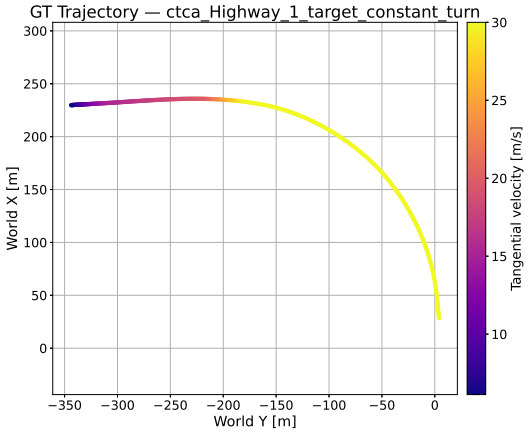


Figure A.14: Highway 1 target constant turn

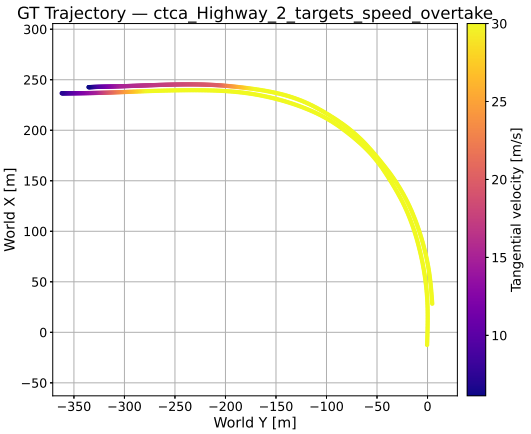


Figure A.15: Highway 2 targets speed overtake

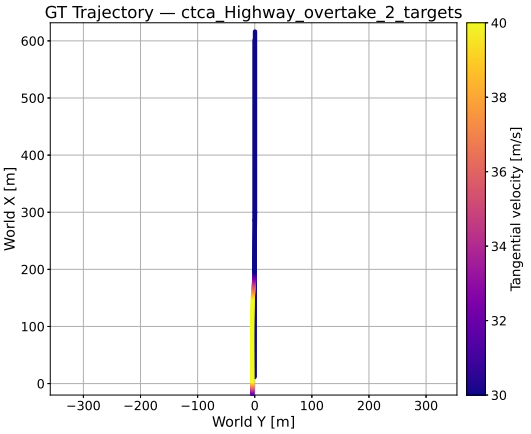


Figure A.16: Highway overtake 2 targets

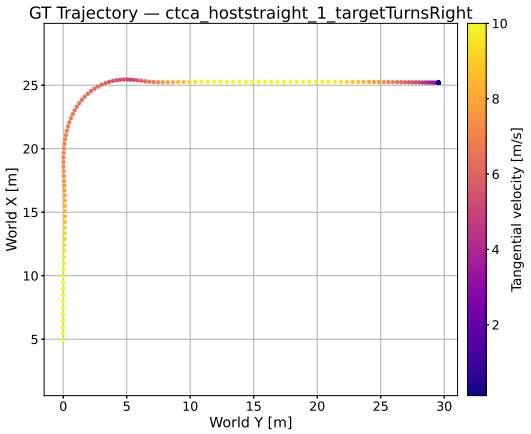


Figure A.17: hoststraight 1 target Turns Right

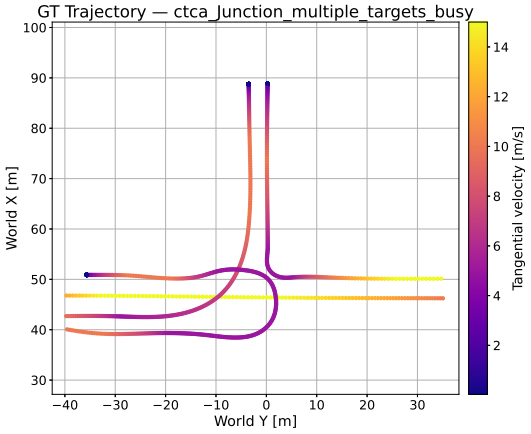


Figure A.18: Junction multiple targets busy

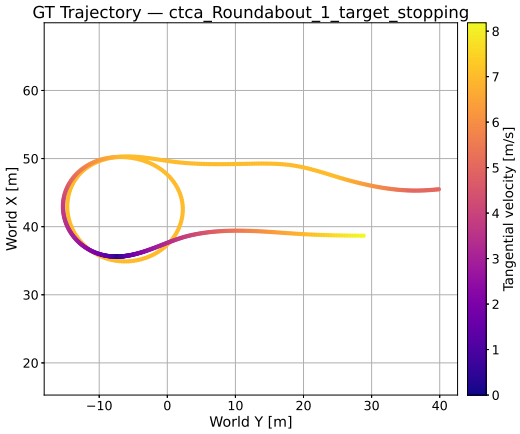


Figure A.19: Roundabout 1 target stopping

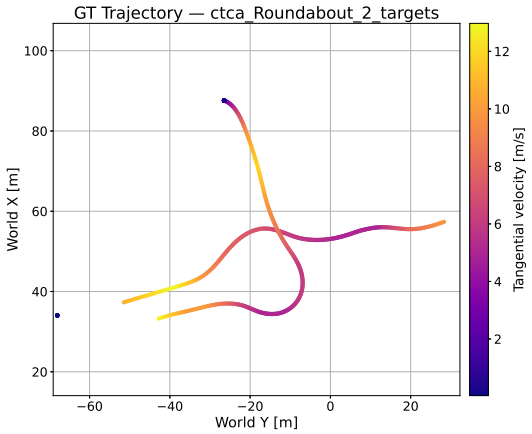


Figure A.20: Roundabout 2 targets

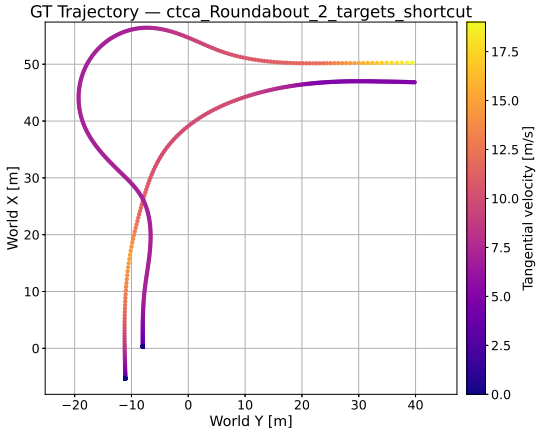


Figure A.21: Roundabout 2 targets shortcut

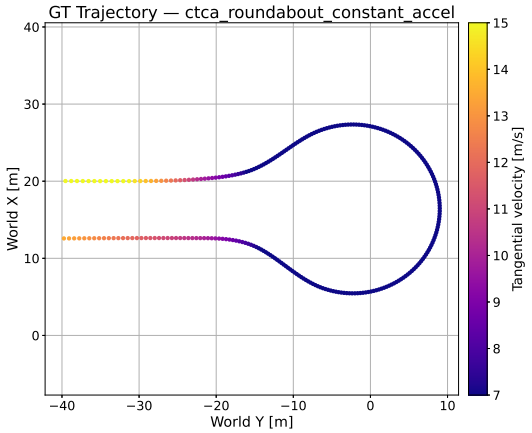


Figure A.22: roundabout constant accel

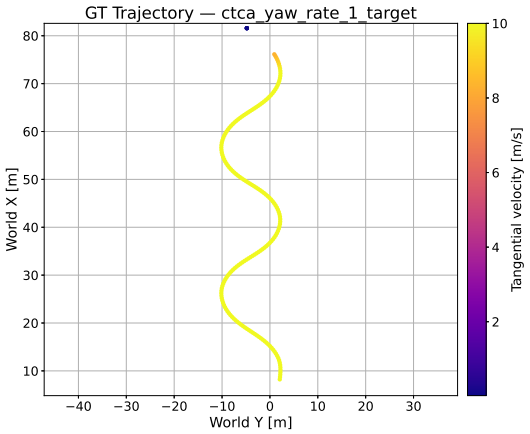


Figure A.23: yaw rate 1 target

A.3 Final Tuning Metrics

A.3.1 CCA

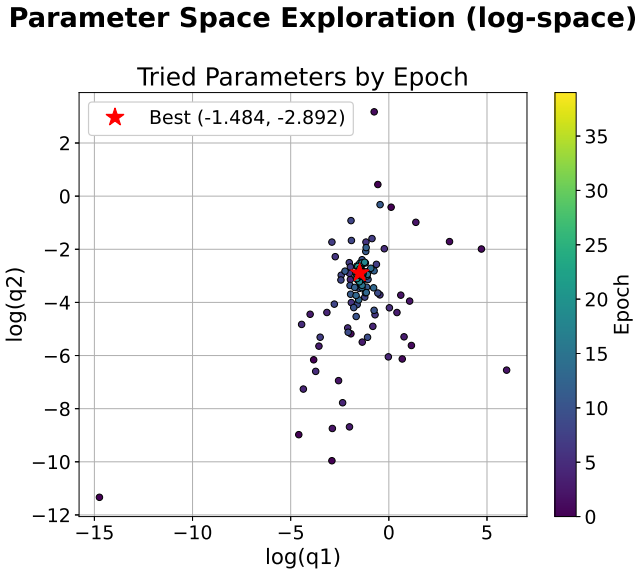


Figure A.24: Final CCA tuning parameter space

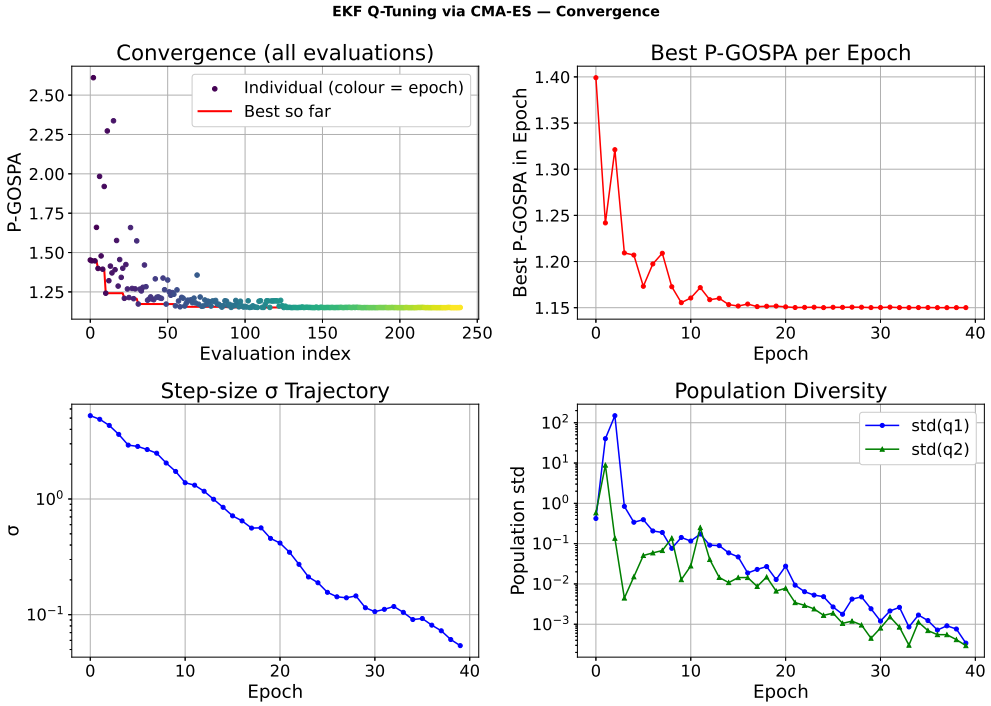


Figure A.25: Final CCA tuning convergence

A.3.2 CTCA

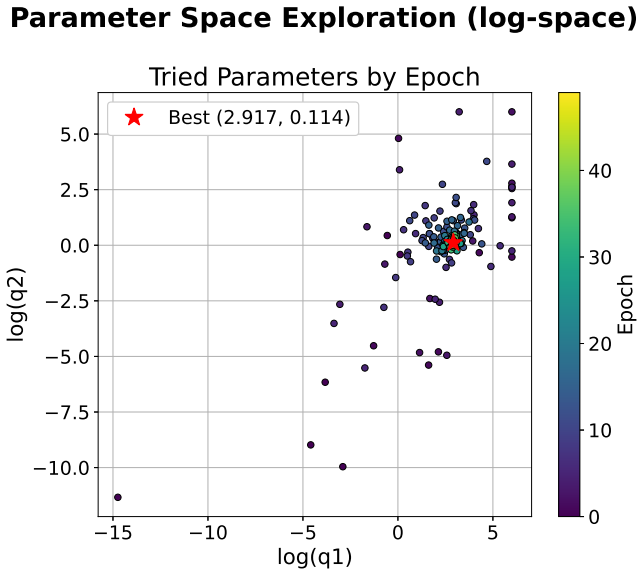


Figure A.26: Final CTCA tuning parameter space

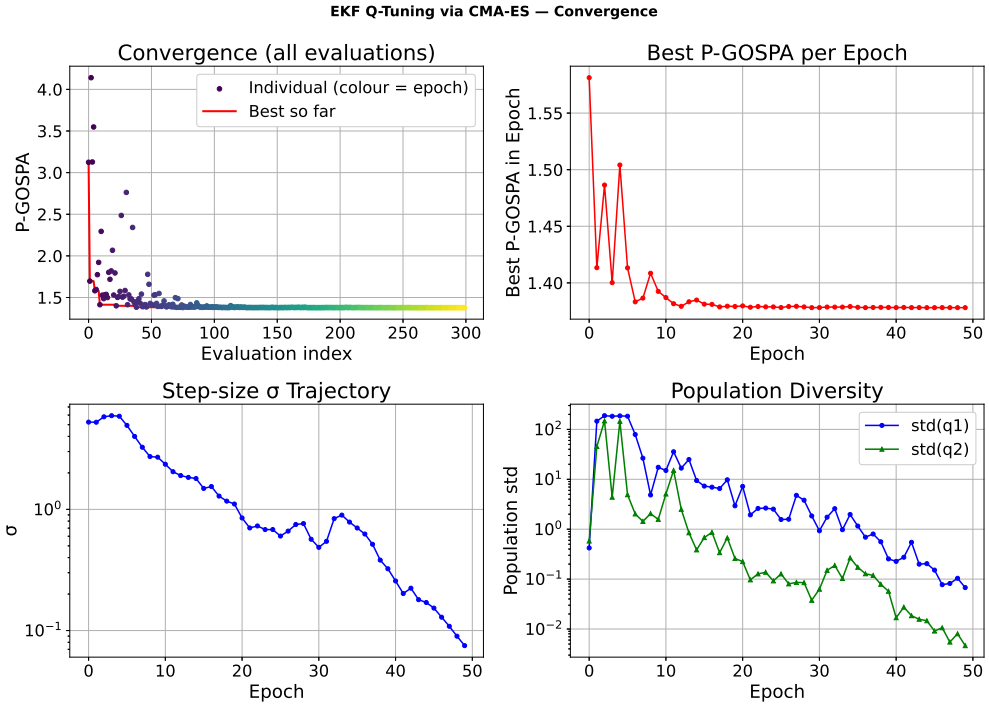


Figure A.27: Final CTCA tuning convergence

A.3.3 IMM

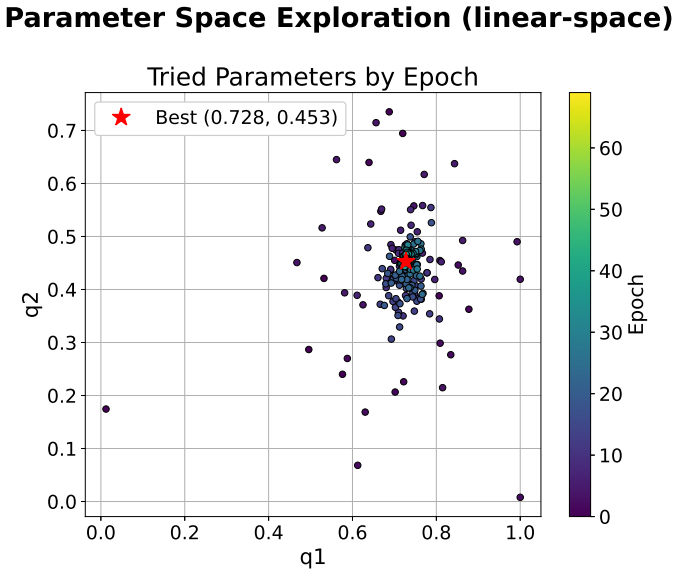


Figure A.28: Final IMM tuning parameter space

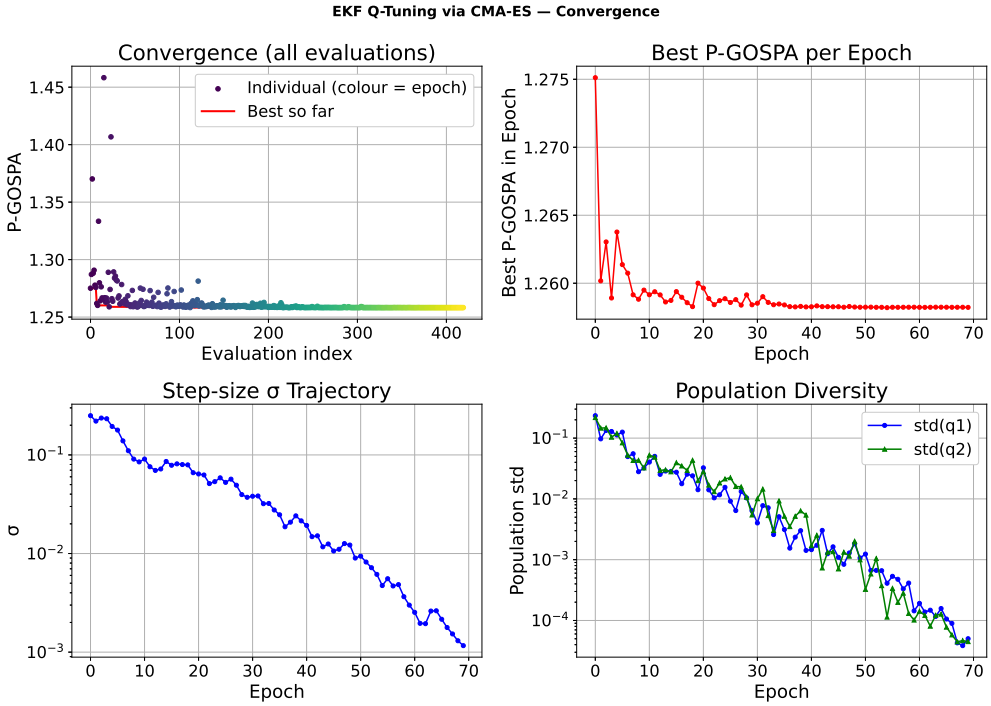


Figure A.29: Final IMM tuning convergence

A.4 One Scenario Tuning - Optimally Tuned Models

A.4.1 Scenario 1

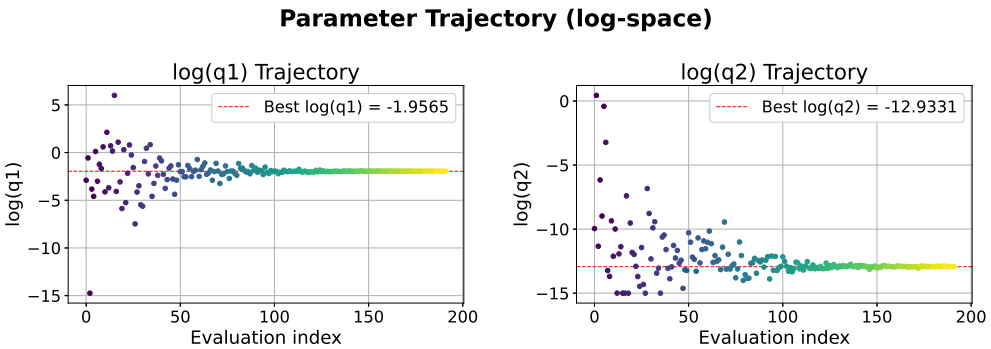


Figure A.30: Tuning parameter trajectories for the CCA model in test scenario 1.

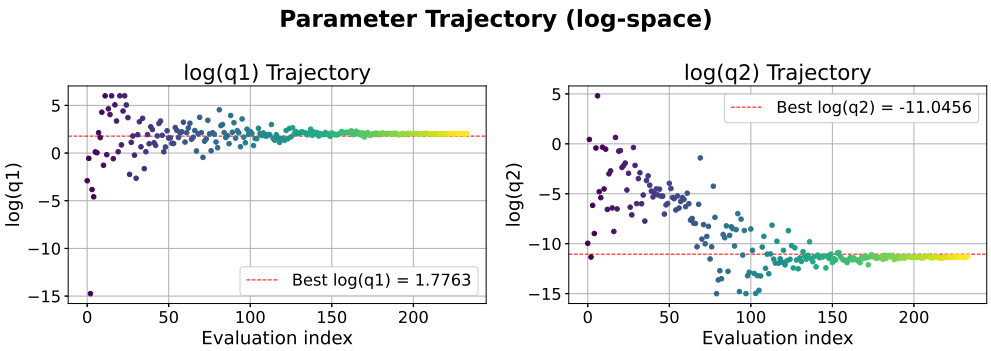


Figure A.31: Tuning parameter trajectories for the CTCA model in test scenario 1.

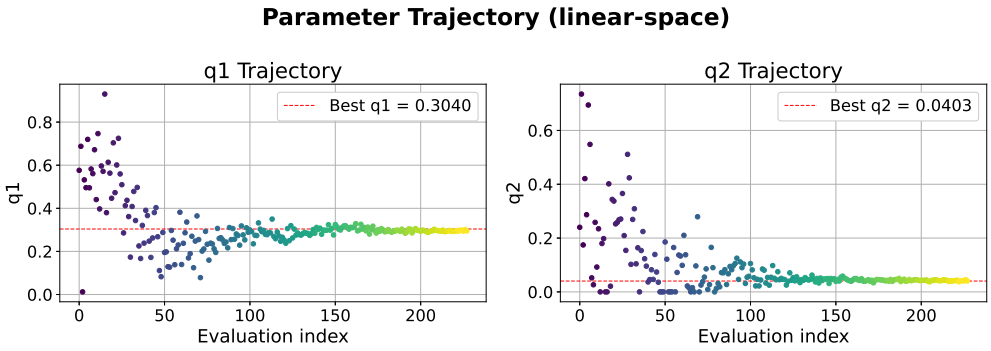


Figure A.32: Tuning parameter trajectories for the IMM filter in test scenario 1.

A.4.2 Scenario 2

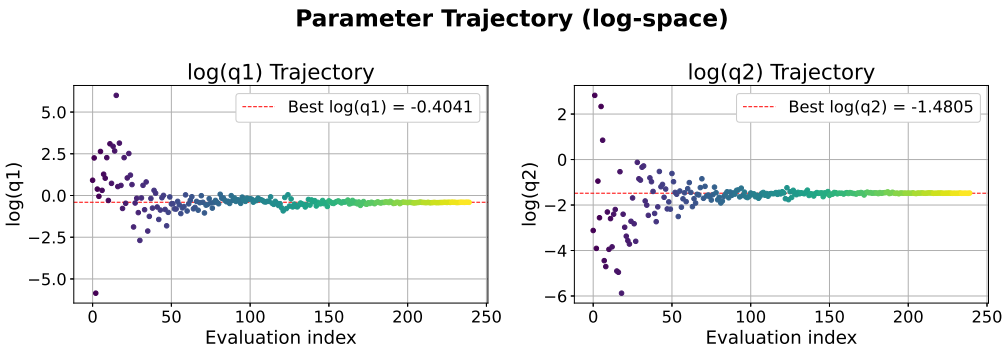


Figure A.33: Tuning parameter trajectories for the CCA model in test scenario 2.

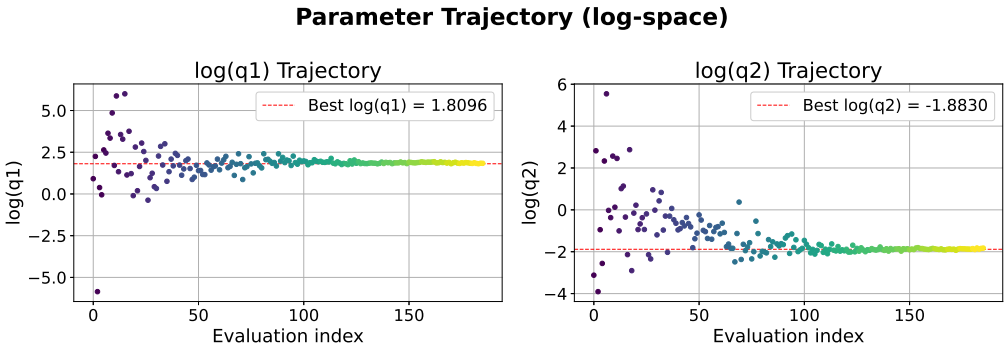


Figure A.34: Tuning parameter trajectories for the CTCA model in test scenario 2.

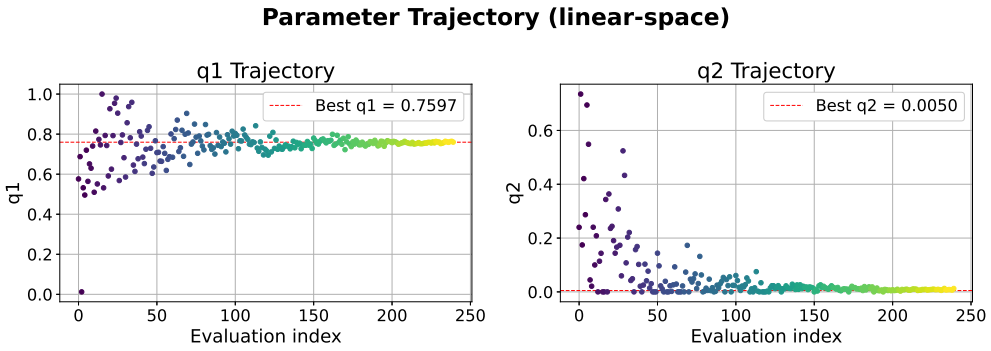


Figure A.35: Tuning parameter trajectories for the IMM filter in test scenario 2.

A.4.3 Scenario 3

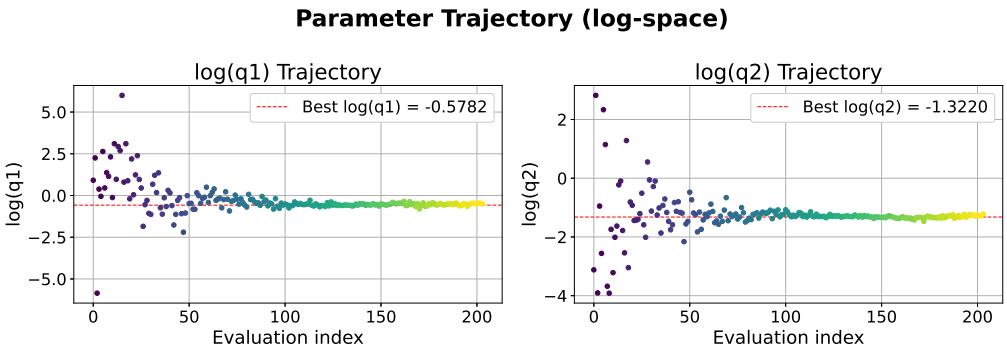


Figure A.36: Tuning parameter trajectories for the CCA model in test scenario 3.

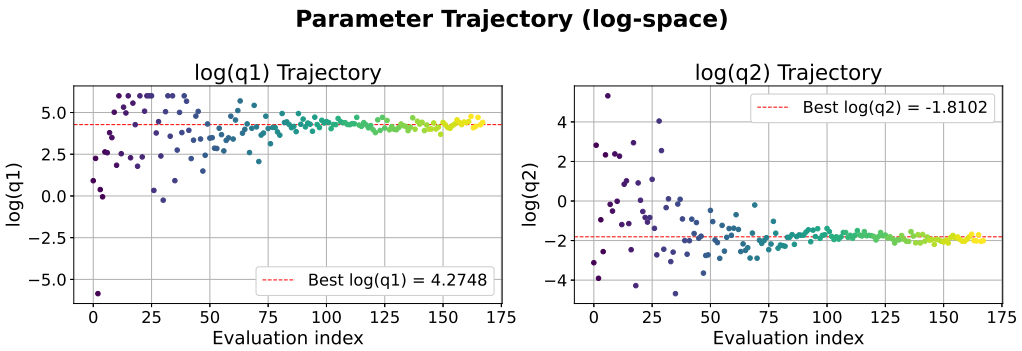


Figure A.37: Tuning parameter trajectories for the CTCA model in test scenario 3.

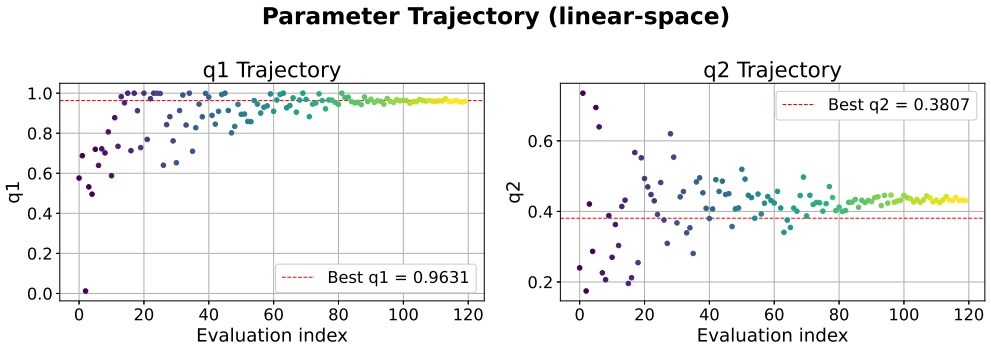


Figure A.38: Tuning parameter trajectories for the IMM filter in test scenario 3.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY