Probabilistic Filtering Techniques for Multi-Target Tracking in Radar Systems

Fall 2024
KOM505E Probability Theory and Stochastic Processes
Term Project Report

İbrahim Halil Bayzan* Bilgehan Tanyeri† Buse Genç‡
*bayzan17@itu.edu.tr, 504221559
†tanyeri18@itu.edu.tr, 504241137
‡gencb@itu.edu.tr 504221129
Istanbul Technical University, Turkey

I. Problem Description: Multi-Target Tracking with Radar

In this project, we will explore the problem of multi-target tracking (MTT) with radar systems, focusing on tracking enemy aircraft while ensuring continuous and accurate target estimation. MTT involves estimating the states (e.g., position, velocity) of multiple moving targets simultaneously. Given the dynamic and uncertain nature of real-world environments, MTT is inherently challenging. All code developed for this project is available on GitHub and will remain accessible until **3rd February 2024**.

Challenges in Multi-Target Tracking

One of the primary challenges in MTT is the correct association of measurements to targets. In a typical scenario, the radar provides measurements of multiple targets at each time step, which introduces ambiguities in associating measurements with existing tracks. For instance, when two targets are close to each other, a received measurement could reasonably belong to either target, leading to potential misassignments.

Another complexity arises from the presence of false alarms and clutter in radar data, which can result in spurious measurements. A robust tracking algorithm must effectively distinguish between true target signals and false detections to avoid errors in track updates. Developing an algorithm that minimizes false assignments while maintaining continuity in tracking is essential to avoid track swapping or loss of track.

Nonlinear Radar Measurements

In our radar model, measurements are obtained in spherical coordinates, consisting of range r, azimuth angle ϕ, and elevation angle θ. The transformation from these nonlinear measurements to the target's state in Cartesian coordinates introduces nonlinearity into the tracking problem. Estimating target position and velocity accurately from these parameters requires managing the nonlinear transformations and their impact on estimation errors.

Additionally, due to the periodic nature of angles (ϕ and θ), angular measurements may require modular adjustments. For example, when a target crosses the azimuth boundary at $\phi = \pm\pi$, it is essential to apply modular arithmetic to avoid abrupt discontinuities in angular values. The nonlinearity and periodicity of these measurements complicate the development of tracking algorithms and require specialized approaches to maintain accurate estimation.

Objectives of Multi-Target Tracking

The primary objective of our radar-based MTT system is to maintain continuous and accurate tracks on enemy aircraft despite the presence of the aforementioned challenges. By ensuring robust track association, mitigating false alarms, and accurately handling nonlinear measurement transformations, we aim to develop a tracking system capable of supporting effective and reliable multi-target tracking. The focus will be on achieving high track continuity and estimation accuracy over extended time periods, enabling a more resilient radar-based tracking capability.

II. Assumptions

Simulation Assumptions

For this multi-target tracking simulation, we make the following assumptions to represent the environment and target behavior:

1. **Initial Positions and Velocities**: The initial positions and velocities of all targets, as well as the ownship, are randomly sampled from within two defined rectangular prisms. These prisms define boundaries for position (x, y, z) and velocity (v_x, v_y, v_z) axes.

2. **Measurement Time Distribution**: The time intervals between consecutive measurements follow a Gaussian distribution with specified mean μ and standard deviation σ. Outliers in measurement times are removed to ensure realistic sampling intervals.

3. **Measurement Generation and Transformation**: Each measurement is generated in the ownship's local frame, represented in spherical coordinates (range r, azimuth ϕ, elevation θ), with no Doppler shift considerations. The transformation from Cartesian coordinates to spherical coordinates involves nonlinear mappings.

4. **Noise Characteristics**: - **Measurement Noise**: Measurement errors are modeled as Gaussian noise, affecting the r, ϕ, θ values. - **Process Noise**: The process noise for target motion is also assumed to be Gaussian, influencing state propagation.

5. **False Alarm Model**: False alarms occur according to a Poisson distribution, with the number of false alarms λ per unit time. Their spatial positions are uniformly sampled from within the initial position prism.

6. **Radar Field of View**: For ease of simulation, it is assumed that the radar has infinite field of view, this does not affect the performance of the filters, it just makes it easier for us to find reasonable scenarios where we can check the filtering algorithm.

Filtering Assumptions

For filtering and state estimation, we implement a set of filtering algorithms—Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Interacting Multiple Model (IMM), and Particle Filter (PF). These filters assume the following:

1. **Gaussian Assumptions**: Both the measurement noise and process noise are generally assumed to be Gaussian-distributed, which simplifies the update and prediction steps, particularly for EKF and UKF. For IMM and PF, additional Gaussian mixture or particle representations handle uncertainty.

2. **Measurement and State Models**: The measurement models in this study utilize a nonlinear representation in polar coordinates, specifically r, ϕ, θ, which has been described previously. These nonlinear measurements are processed differently by each filter: the EKF linearizes the measurement model to estimate the target states, whereas the UKF uses sigma points to sample from the prior distribution, transforms them through the nonlinear measurement function, and uses these transformed points for state estimation. The PF represents the target state distribution with particles and employs the state model to propagate these particles over time. The IMM combines multiple EKFs, each using the same measurement and state models but with different parameters to capture varying process noise levels across modes.

All filters assume a constant velocity state model, a widely used approach in the literature for target tracking problems. This model simplifies state propagation and provides a consistent basis for comparing the performance of the different filters.

3. **Filter-Specific Parameters**: - The IMM utilizes a mode transition matrix to manage multiple motion models. - The PF relies on a resampling technique to maintain a suitable particle distribution and prevent particle degeneracy.

For more detailed descriptions and assumptions related to each filtering algorithm, please refer to the appendix. Additionally, for more details about the simulation setup and implementation, please refer to the provided codebase. A sample scenario from the simulation is illustrated in Figure 1.

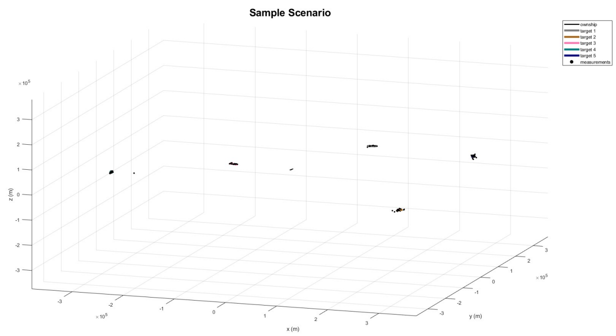

Fig. 1. A sample scenario from the simulation

III. Methodology and the Solution

A. Filter Integration and Comparison

The tracking algorithm described in Algorithm 1 maintains a modular structure, enabling the seamless integration of various filtering techniques. This modularity ensures that the core tracking logic—such as track prediction, measurement association, and track confirmation—remains unchanged, regardless of the filter used. The filtering operation, which updates tracks with new measurements, is the only component that varies between the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Interacting Multiple Model (IMM), and Particle Filter (PF).

Each filtering technique offers distinct methods for addressing the nonlinearities and uncertainties inherent in multi-target tracking. EKF and UKF utilize Gaussian noise assumptions and differ primarily in their handling of nonlinear transformations. IMM employs a mode-switching mechanism to capture varying target motion dynamics, while PF represents state distributions with particles, offering flexibility but at a higher computational cost.

Despite their differences, integrating these filters into the tracking framework involves modifying only the filtering step in Algorithm 1. This flexibility allows for a direct comparison of their performance without altering the broader structure of the tracking process.

B. Analysis of Filter Performance

The performance of each filtering technique is evaluated using positional and velocity errors, as shown in Figure 2. These errors are calculated from a 3D rectangular prism with axes representing the number of Monte Carlo runs (n_{monte}), the number of targets (n_{target}), and time. The component inside the prism is either the positional or velocity error at each combination of these axes.

To derive the appended graphs, we first compute the mean error along the n_{monte} and n_{target} axes for each time step. This results in a 1D time series of errors. The overall error

Algorithm 1 Tracking Algorithm Pseudocode

Require: Measurements `meas`, tracking parameters `params`
Ensure: Tracking logs `logs` containing track history and confirmed tracks

Initialize previous time and track counter
for each measurement in `meas` **do**
  Delete old tracks: Remove tracks not updated within a specified time limit
  Predict tracks: Propagate each existing track state based on the system model and elapsed time
  Calculate association metrics: Determine the distance or association score between each track and the current measurement
  Assign measurement to track or create new track:
  if no existing track matches measurement **then**
    Initialize a new track with the measurement
  else
    Update the assigned track with the measurement using EKF, UKF, PF or IMM
  end if
  Confirm tracks: Mark tracks as confirmed if they meet the required update threshold
  Store confirmed tracks in `logs`
end for

values displayed in the figure titles are obtained by taking an additional mean across time.

Figure 2 highlights the following key observations:

- **Interactive Multiple Model (IMM)**: The IMM achieves the lowest positional and velocity errors, outperforming all other filters. Its superior performance can be attributed to its ability to adapt to varying target dynamics by switching between multiple motion models. This adaptability makes IMM the most robust choice under the simulated conditions.
- **Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF)**: EKF and UKF produce similar results, with slightly better performance than the Particle Filter (PF). This similarity arises because the non-linearities in the system are not severe and the state distribution is inherently close to a gaussian distribution, allowing both EKF and UKF to give acceptable results using their assumptions.
- **Particle Filter (PF)**: The PF exhibits the highest errors among the filters. Its particle-based representation struggles in this simulation due to the Gaussian noise assumption of the simulator, which is not ideal for the PF's flexibility. However, the PF's performance could improve significantly if the noise probability density function were altered to better match the PF's strengths, such as handling highly non-Gaussian or multimodal noise distributions.

IV. Conclusion

In this study, we demonstrated the effectiveness of various filtering techniques for multi-target tracking with radar, emphasizing their integration into a modular tracking framework. Our results showed that the IMM filter outperforms EKF, UKF, and PF due to its adaptability to varying motion dynamics, confirming its status as a state-of-the-art method in the field. The analysis further highlighted that the performance of each filter depends on system characteristics, particularly the noise model and non-linearities, underscoring the importance of selecting appropriate filtering techniques for specific scenarios.

V. Group Work

Each member of the group contributed equally.

VI. Appendix

A. Kalman Filter Framework and Motivation

The Kalman Filter is a recursive Bayesian estimator that provides optimal estimates of the system state by minimizing the covariance of the estimation error. This derivation presents the Kalman Filter from a Bayesian perspective, utilizing conditional probabilities and Gaussian assumptions.

State and Measurement Models

State Equation (Process Model)

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

Measurement Equation

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

Where:

- $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector at time k.
- $\mathbf{F}_{k-1} \in \mathbb{R}^{n\times n}$ is the state transition matrix.
- $\mathbf{B}_{k-1} \in \mathbb{R}^{n\times p}$ is the control input matrix.
- $\mathbf{u}_{k-1} \in \mathbb{R}^p$ is the control vector.
- $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is the process noise.
- $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement vector at time k.
- $\mathbf{H}_k \in \mathbb{R}^{m\times n}$ is the observation matrix.
- $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the measurement noise.

Assumptions:

- The initial state $\mathbf{x}_0$ is Gaussian with mean $\hat{\mathbf{x}}_0$ and covariance $\mathbf{P}_0$.
- The process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$ are independent of each other and of past states and measurements.
- The relationships are linear, and all noise terms are Gaussian.

Predict the State

Using the state equation (1), the **prior** estimate of the state $\mathbf{x}_k$ given observations $\mathbf{z}_{1:k-1}$ is:

$$\mathbf{x}_k|\mathbf{z}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{3}$$

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{4}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1} \tag{5}$$

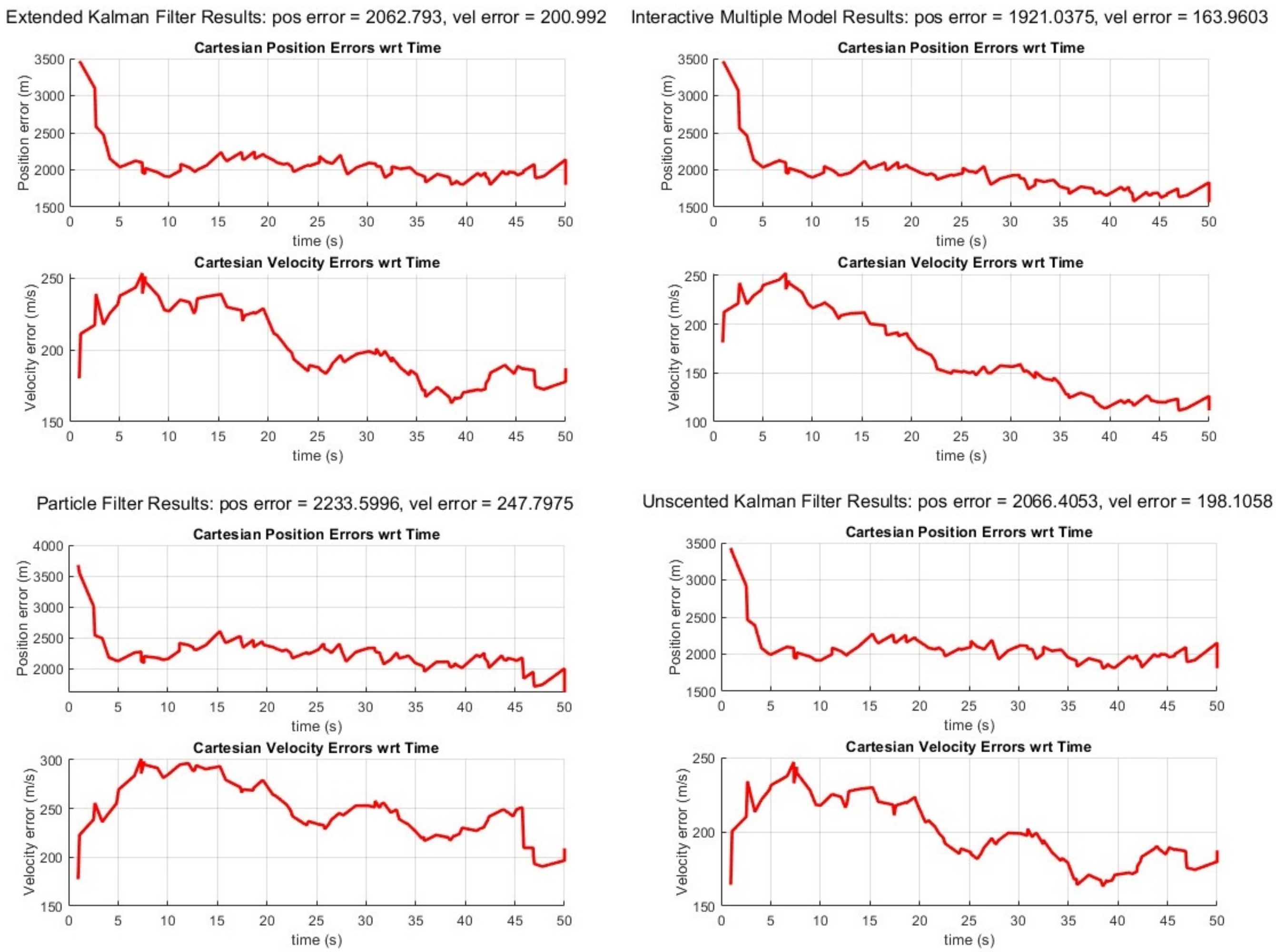

Fig. 2. Error graphs for each filter

Predict the Measurement

Using the measurement equation (2), predict the measurement $\mathbf{z}_k$ based on the predicted state:

$$\mathbf{z}_k|\mathbf{z}_{1:k-1} \sim \mathcal{N}(\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k) \tag{6}$$

Where:

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k \tag{7}$$

Compute the Kalman Gain

The **Kalman Gain $\mathbf{K}_k$** determines how much the predictions are adjusted based on the new measurement:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1} \tag{8}$$

Update the State Estimate

Update the prior estimate with the new measurement to obtain the **posterior** estimate:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \tag{9}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{10}$$

Covariance Minimization

The Kalman Gain $\mathbf{K}_k$ is derived to minimize the posterior covariance $\mathbf{P}_k$. This ensures that the estimation error covariance is as small as possible, providing the most accurate state estimate under the given assumptions.

Starting from the update equation for the covariance:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}$$

Substituting the Kalman Gain from (8):

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1}\mathbf{H}_k)\mathbf{P}_{k|k-1}$$

Probabilistic Equivalents of the Filtering Equations

Basic Bayes' Theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

Bayes' theorem provides the foundation for updating the posterior distribution of the state $\mathbf{x}_k$ given the new measurement $\mathbf{z}_k$. Applying Bayes' theorem to the filtering process:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{11}$$

Where:

- $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is the **prior** distribution of the state.
- $p(\mathbf{z}_k|\mathbf{x}_k)$ is the **likelih ood** of the measurement given the state.
- $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ is the **evidence**, ensuring the posterior is normalized.

Prior Distribution:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{12}$$

Prediction of the State:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \tag{13}$$

Prediction of the Measurement:

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}_k) \tag{14}$$

Posterior Distribution via Bayes' Theorem:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{15}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_k) \tag{16}$$

Proof that the Kalman Filter is a Bayesian Filter

Starting from Bayes' theorem and substituting the Gaussian distributions:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}$$

Given that both $p(\mathbf{z}_k|\mathbf{x}_k)$ and $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ are Gaussian, their product is also Gaussian. Thus, the posterior $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is Gaussian, characterized by the updated mean $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_k$ as derived in (9) and (10).

The Kalman Gain $\mathbf{K}_k$ in (8) is derived to minimize the posterior covariance $\mathbf{P}_k$. This is achieved by solving:

$$\mathbf{K}_k = \arg\min_{\mathbf{K}} \operatorname{Tr}\left[(\mathbf{I} - \mathbf{K}\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}\mathbf{H}_k)^\top + \mathbf{K}\mathbf{R}_k\mathbf{K}^\top\right]$$

Taking the derivative with respect to $\mathbf{K}_k$ and setting it to zero yields:

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

Substituting $\mathbf{K}_k$ back into the posterior covariance (10) ensures that $\mathbf{P}_k$ is minimized, confirming that the Kalman Filter provides the optimal Bayesian estimate under Gaussian assumptions.

Kalman Filter Equations

Combining the prediction and update steps, the Kalman Filter recursion is summarized as follows:

Predict the State:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{17}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1} \tag{18}$$

Compute the Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k)^{-1} \tag{19}$$

$$\tag{20}$$

Update the State Estimate:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \tag{21}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{22}$$

Intuitive Understanding

- **Prediction Step**: Uses the system model to predict the state at the next time step, accounting for uncertainty growth due to process noise.
- **Update Step**: Adjusts the prediction based on the new measurement, reducing uncertainty by incorporating information from the measurement.

The **Kalman Gain** $\mathbf{K}_k$ balances the trust between the prediction and the measurement. A higher gain places more weight on the measurement, while a lower gain trusts the prediction more.

B. Extended Kalman Filter

Linearization Motivation

In nonlinear systems, a Gaussian variable undergoing a nonlinear transformation does not remain Gaussian. Specifically, if $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$ and $\mathbf{z} = h(\mathbf{x})$ is nonlinear, then $\mathbf{z}$ does not follow a Gaussian distribution. This non-Gaussianity complicates the exact Bayesian update, as the posterior distribution cannot be represented in closed-form.

Nonlinear State and Measurement Models

State Equation (Process Model)

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{23}$$

Measurement Equation

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{24}$$

Where:

- $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector at time k.
- $f(\cdot)$ is a nonlinear state transition function.
- $\mathbf{u}_{k-1} \in \mathbb{R}^p$ is the control vector.
- $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is the process noise.
- $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement vector at time k.
- $h(\cdot)$ is a nonlinear measurement function.
- $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the measurement noise.

Linearization via Taylor Series Expansion

To approximate the nonlinear functions $f(\cdot)$ and $h(\cdot)$, the EKF linearizes them around the current estimate using the first-order Taylor series expansion:

$$f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \approx f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) + \mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) \tag{25}$$

$$h(\mathbf{x}_k) \approx h(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \tag{26}$$

Where:

- $\mathbf{F}_{k-1} = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}}$ is the Jacobian matrix of f with respect to $\mathbf{x}$.
- $\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}$ is the Jacobian matrix of h with respect to $\mathbf{x}$.

This linearization allows the EKF to apply the Kalman Filter equations by approximating the nonlinear functions as linear, thereby maintaining the Gaussian assumptions necessary for closed-form Bayesian updates.

This derivation presents the EKF from a Bayesian perspective, utilizing conditional probabilities, Gaussian assumptions, and linearization techniques.

Assumptions:

- The initial state $\mathbf{x}_0$ is Gaussian with mean $\hat{\mathbf{x}}_0$ and covariance $\mathbf{P}_0$.
- The process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$ are independent of each other and of past states and measurements.
- The relationships are nonlinear, and all noise terms are Gaussian.

Extended Kalman Filter Equations

Predict the State:

$$\mathbf{x}_k | \mathbf{z}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{27}$$

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \tag{28}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1} \tag{29}$$

Predict the Measurement:

$$\mathbf{z}_k | \mathbf{z}_{1:k-1} \sim \mathcal{N}(\hat{\mathbf{z}}_{k|k-1}, \mathbf{S}_k) \tag{30}$$

$$\hat{\mathbf{z}}_{k|k-1} = h(\hat{\mathbf{x}}_{k|k-1}) \tag{31}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k \tag{32}$$

Compute the Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1} \tag{33}$$

Update the State Estimate:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \tag{34}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{35}$$

Where:

- $\mathbf{F}_{k-1} = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}}$ is the Jacobian matrix of f with respect to $\mathbf{x}$.
- $\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}$ is the Jacobian matrix of h with respect to $\mathbf{x}$.

Intuitive Understanding

The Extended Kalman Filter adapts the Bayesian Kalman Filter framework to nonlinear systems by linearizing the state and measurement models around the current estimate. This linearization enables the EKF to maintain Gaussian distributions for the state estimates, facilitating recursive Bayesian updates despite inherent nonlinearities

C. Unscented Kalman Filter

Motivation and Unscented Transform

While the Extended Kalman Filter (EKF) linearizes nonlinear functions using first-order Taylor series expansions, the Unscented Kalman Filter (UKF) leverages the *Unscented Transform (UT)*. The UT uses a deterministic sampling approach to capture the mean and covariance of a probability distribution undergoing a nonlinear transformation more accurately than a simple linearization. Instead of linearizing the functions, the UKF *propagates* a carefully chosen set of *sigma points* through the original nonlinear functions.

Sigma Points and Weights

Given an n-dimensional state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ with covariance $\mathbf{P}_{k-1|k-1}$, the UKF first generates $2n+1$ sigma points $\boldsymbol{\chi}_{k-1}^{(i)}$ according to:

$$\boldsymbol{\chi}_{k-1}^{(0)} = \hat{\mathbf{x}}_{k-1|k-1}, \tag{36}$$

$$\boldsymbol{\chi}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} + \left(\sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}\right)_i, \quad i = 1, \ldots, n, \tag{37}$$

$$\boldsymbol{\chi}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} - \left(\sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}\right)_{i-n}, \quad i = n+1, \ldots, 2n, \tag{38}$$

where $\sqrt{\cdot}$ denotes the matrix square root (e.g., via Cholesky decomposition), and $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter with typical choices for (α, β, κ) guided by problem-specific heuristics.[1]

The corresponding *weights* for the mean ($W_m^{(i)}$) and covariance ($W_c^{(i)}$) are:

$$W_m^{(0)} = \frac{\lambda}{n+\lambda}, \quad W_c^{(0)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), \tag{39}$$

$$W_m^{(i)} = W_c^{(i)} = \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, 2n. \tag{40}$$

[1]Commonly, $\alpha = 10^{-3}$, $\beta = 2$, and $\kappa = 0$ for many applications.

Here, n is the state dimension. These weights ensure that the first two moments of the sigma points accurately represent those of the underlying Gaussian distribution.

Nonlinear State and Measurement Models

Unscented Kalman Filter Equations

Like the EKF, the UKF proceeds in two main steps: *prediction* and *update*. However, instead of linearizing about the current estimate, it propagates the sigma points directly through the nonlinear functions $f(\cdot)$ and $h(\cdot)$.

Prediction Step: **1) State Prediction via Sigma Points**

Propagate each sigma point $\boldsymbol{\chi}_{k-1}^{(i)}$ through the nonlinear state transition $f(\cdot)$:

$$\boldsymbol{\chi}_{k|k-1}^{(i)} = f\big(\boldsymbol{\chi}_{k-1}^{(i)}, \mathbf{u}_{k-1}\big), \quad i = 0, 1, \dots, 2n. \tag{41}$$

Then, compute the predicted state mean and covariance:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \, \boldsymbol{\chi}_{k|k-1}^{(i)}, \tag{42}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} W_c^{(i)} \left[\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right] \left[\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right]^\top + \mathbf{Q}_{k-1}. \tag{43}$$

Prediction Step: **2) Measurement Prediction via Sigma Points**

Propagate the *predicted* sigma points $\boldsymbol{\chi}_{k|k-1}^{(i)}$ through the measurement function $h(\cdot)$:

$$\boldsymbol{\gamma}_{k|k-1}^{(i)} = h\big(\boldsymbol{\chi}_{k|k-1}^{(i)}\big), \quad i = 0, 1, \dots, 2n. \tag{44}$$

Compute the predicted measurement mean:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \, \boldsymbol{\gamma}_{k|k-1}^{(i)}, \tag{45}$$

$$\tag{46}$$

Compute the Kalman Gain:

Update Step: **Cross-Covariance and Measurement Covariance**

Compute the cross-covariance between the state and measurement:

$$\mathbf{P}_{x,z} = \sum_{i=0}^{2n} W_c^{(i)} \left[\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right] \left[\boldsymbol{\gamma}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}\right]^\top, \tag{47}$$

$$\mathbf{S}_k = \sum_{i=0}^{2n} W_c^{(i)} \left[\boldsymbol{\gamma}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}\right] \left[\boldsymbol{\gamma}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}\right]^\top + \mathbf{R}_k. \tag{48}$$

Update the State Estimate

The Kalman gain in the UKF is then:

$$\mathbf{K}_k = \mathbf{P}_{x,z} \, \mathbf{S}_k^{-1}. \tag{49}$$

Given a new measurement $\mathbf{z}_k$, the updated mean and covariance are:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \big(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\big), \tag{50}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \, \mathbf{S}_k \, \mathbf{K}_k^\top. \tag{51}$$

Intuitive Understanding

Unlike the EKF, the UKF does not rely on Jacobians or explicit linearization. Instead, it uses sigma points that *approximately* capture the mean and covariance after passing through nonlinearities. This *unscented transform* often yields more accurate results in target tracking scenarios with significant nonlinearities, since it better preserves higher-order moments of the underlying distributions than a first-order Taylor expansion.

D. Particle Filter (Sequential Importance Sampling with Resampling - SIS-R)

Motivation for Monte Carlo Methods

Particle Filter is a sequential Monte Carlo method which enables to estimate the state of a nonlinear and/or non-Gaussian dynamic system.It achieves this by dividing the probability distribution it wants to estimate into small pieces and estimating the posterior density numerically using Monte Carlo methods. The SIS-R variant incorporates resampling to address the issue of particle degeneracy, ensuring a diverse and representative set of particles. The few assumptions made in the particle filter are how the initial distribution of the random variable we want to estimate is and how the measurement variable is distributed. In this way, we can estimate the random variable we want to estimate without adhering to the restriction that the random variable is only Gaussian distributed.It does this by selecting particles on the probability distribution of the random variable and weighting these particles according to the incoming measurement.

Assumptions:

- The initial state $\mathbf{x}_0$ is sampled from a prior distribution $p(\mathbf{x}_0)$.
- The process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$ are independent of each other and of past states and measurements.
- The relationships can be nonlinear, and noise terms can follow arbitrary distributions, not necessarily Gaussian.

Particle Filter Equations

Initialization: Initialize a set of N particles $\{\mathbf{x}_0^{(i)}\}_{i=1}^N$ by sampling from the initial distribution:

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \quad i = 1, 2, \dots, N \tag{52}$$

Assign an initial weight to each particle:

$$w_0^{(i)} = \frac{1}{N}, \quad i = 1, 2, \dots, N \tag{53}$$

Prediction Step: Propagate each particle through the state transition model:

$$\mathbf{x}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}^{(i)}, \quad i = 1, 2, \dots, N \tag{54}$$

Update Step: Update the weights of each particle based on the likelihood of the new measurement:

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot p(\mathbf{z}_k|\mathbf{x}_k^{(i)}), \quad i = 1, 2, \ldots, N \tag{55}$$

Normalize the weights:

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}, \quad i = 1, 2, \ldots, N \tag{56}$$

Resampling Step: Resample the particles based on their normalized weights to focus on regions with higher probability:

$$\{\mathbf{x}_k^{(i)}\}_{i=1}^N \leftarrow \{\mathbf{x}_k^{(i^*)}\}_{i=1}^N, \quad \text{where } i^* \sim \text{Resample}\left(\{\tilde{w}_k^{(i)}\}_{i=1}^N\right) \tag{57}$$

Effective Sample Size and Resampling Condition:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} \left(\tilde{w}_k^{(i)}\right)^2} \tag{58}$$

$$\text{If } N_{\text{eff}} < \frac{N}{2}, \text{ then resample} \tag{59}$$

After resampling, reset the weights:

$$\tilde{w}_k^{(i)} = \frac{1}{N}, \quad i = 1, 2, \ldots, N \tag{60}$$

Posterior Mean and Covariance Calculations:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N} \tilde{w}_k^{(i)} \mathbf{x}_k^{(i)} \tag{61}$$

$$\mathbf{P}_k = \sum_{i=1}^{N} \tilde{w}_k^{(i)} \left(\mathbf{x}_k^{(i)} - \hat{\mathbf{x}}_k\right) \left(\mathbf{x}_k^{(i)} - \hat{\mathbf{x}}_k\right)^\top \tag{62}$$

Summary of the Particle Filter (SIS-R) Equations

Initialization:

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \quad i = 1, 2, \ldots, N \tag{63}$$

$$w_0^{(i)} = \frac{1}{N}, \quad i = 1, 2, \ldots, N \tag{64}$$

For each time step $k = 1, 2, \ldots$:

$$\text{Prediction:} \quad \mathbf{x}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}^{(i)} \tag{65}$$

$$\text{Update:} \quad w_k^{(i)} = w_{k-1}^{(i)} \cdot p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) \tag{66}$$

$$\text{Normalization:} \quad \tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}} \tag{67}$$

$$\text{Resampling:} \quad \mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i^*)}, \quad i^* \sim \text{Resample}\left(\{\tilde{w}_k^{(i)}\}_{i=1}^N\right) \tag{68}$$

$$\text{Reset Weights:} \quad \tilde{w}_k^{(i)} = \frac{1}{N}, \quad i = 1, 2, \ldots, N \tag{69}$$

Intuitive Understanding

The Particle Filter maintains a set of weighted particles to approximate the state distribution. Initially, a diverse set of particles is drawn from the prior. Each time step, these particles are propagated through the system model (prediction), then their weights are updated according to how well they match the new measurement (update). A subsequent resampling step eliminates low-weight particles and replicates high-weight ones, concentrating computational effort where the state likelihood is highest.

E. Interacting Multiple Model (IMM) Algorithm

Motivation

In many target tracking applications, the system dynamics can switch between different modes (e.g., straight motion, coordinated turn, acceleration, etc.). A single dynamic model may not be sufficient to accurately track the target's state across these different regimes. The *Interacting Multiple Model (IMM)* filter addresses this challenge by running multiple filters in parallel, each corresponding to a distinct motion model, and then probabilistically combining (or *mixing*) their estimates.

Framework and Assumptions

Multiple Models:

- Suppose there are r models, $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_r\}$.
- Each model $\mathcal{M}_j$ has its own state transition function $f_j(\cdot)$, possibly its own process noise covariance $\mathbf{Q}_j$, and its own measurement function $h_j(\cdot)$ or measurement noise covariance $\mathbf{R}_j$.
- The state space dimension is (for simplicity) common across models, denoted n.

Markov Switching:

- The mode (model) at each time step is a Markov chain with a known transition probability matrix:

$$\mathbf{\Pi} = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1r} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{r1} & \pi_{r2} & \cdots & \pi_{rr} \end{bmatrix}, \tag{70}$$

  where $\pi_{ij} = P(\mathcal{M}_j \text{ at time } k \mid \mathcal{M}_i \text{ at time } k-1)$.
- The initial probabilities of each model being active are given by $\mu_{j,0}$, with $\sum_{j=1}^{r} \mu_{j,0} = 1$.

IMM Filtering Steps

At each time step k, the IMM executes four main steps: *interaction/mixing*, *model-specific prediction and update*, *model probability update*, and *combination*.

1) Interaction (Mixing): Before predicting each model's state, the IMM computes a *mixed* initial condition for each model by combining the state estimates from all other models, weighted by their probabilities of transitioning to the current model.

- Let $\hat{\mathbf{x}}_{k-1|k-1}^{(j)}$ and $\mathbf{P}_{k-1|k-1}^{(j)}$ be the estimated mean and covariance from filter j at time $k-1$.

- Let $\mu_{k-1}^{(j)}$ be the probability that model j was active at time $k-1$.

Mixing probabilities:

$$\mu_{k-1}^{(i|j)} = \frac{\pi_{ij}\,\mu_{k-1}^{(i)}}{\sum_{\ell=1}^{r} \pi_{\ell j}\,\mu_{k-1}^{(\ell)}}, \quad \text{for } i,j \in \{1,\dots,r\}. \tag{71}$$

These $\mu_{k-1}^{(i|j)}$ give the conditional probability that the filter was in model i given it is now in model j.

Mixed initial mean and covariance for model j:

$$\bar{\mathbf{x}}_{k-1}^{(j)} = \sum_{i=1}^{r} \mu_{k-1}^{(i|j)}\,\hat{\mathbf{x}}_{k-1|k-1}^{(i)}, \tag{72}$$

$$\bar{\mathbf{P}}_{k-1}^{(j)} = \sum_{i=1}^{r} \mu_{k-1}^{(i|j)}\Big[\mathbf{P}_{k-1|k-1}^{(i)} + \big(\hat{\mathbf{x}}_{k-1|k-1}^{(i)} - \bar{\mathbf{x}}_{k-1}^{(j)}\big) \times \big(\hat{\mathbf{x}}_{k-1|k-1}^{(i)} - \bar{\mathbf{x}}_{k-1}^{(j)}\big)^{\top}\Big]. \tag{73}$$

2) Model-Specific Prediction and Update: Using the *mixed* initial conditions $(\bar{\mathbf{x}}_{k-1}^{(j)}, \bar{\mathbf{P}}_{k-1}^{(j)})$, each model j performs its own prediction and update steps *as if it were a standard filter* (EKF, UKF, or another variant). For example, if model j uses an EKF:

$$\begin{aligned}
\text{Predict:}\quad & \hat{\mathbf{x}}_{k|k-1}^{(j)} = f_j\big(\bar{\mathbf{x}}_{k-1}^{(j)}, \mathbf{u}_{k-1}\big), && (74)\\
& \mathbf{P}_{k|k-1}^{(j)} = \mathbf{F}_j\,\bar{\mathbf{P}}_{k-1}^{(j)}\,\mathbf{F}_j^{\top} + \mathbf{Q}_j, && (75)\\
\text{Update:}\quad & \hat{\mathbf{z}}_{k|k-1}^{(j)} = h_j\big(\hat{\mathbf{x}}_{k|k-1}^{(j)}\big), && (76)\\
& \mathbf{S}_k^{(j)} = \mathbf{H}_j\,\mathbf{P}_{k|k-1}^{(j)}\,\mathbf{H}_j^{\top} + \mathbf{R}_j, && (77)\\
& \mathbf{K}_k^{(j)} = \mathbf{P}_{k|k-1}^{(j)}\,\mathbf{H}_j^{\top}\big(\mathbf{S}_k^{(j)}\big)^{-1}, && (78)\\
& \hat{\mathbf{x}}_{k|k}^{(j)} = \hat{\mathbf{x}}_{k|k-1}^{(j)} + \mathbf{K}_k^{(j)}\big(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}^{(j)}\big), && (79)\\
& \mathbf{P}_{k|k}^{(j)} = \big(\mathbf{I} - \mathbf{K}_k^{(j)}\,\mathbf{H}_j\big)\,\mathbf{P}_{k|k-1}^{(j)}. && (80)
\end{aligned}$$

(Or similarly, if model j employs a UKF, then the corresponding UKF prediction and update would be applied.)

3) Model Probability Update: Next, each filter's updated likelihood is used to re-weight the model probabilities:

$$\Lambda_k^{(j)} = p\big(\mathbf{z}_k \mid \mathcal{M}_j, \mathbf{z}_{1:k-1}\big) \;=\; \mathcal{N}\Big(\mathbf{z}_k \;\Big|\; \hat{\mathbf{z}}_{k|k-1}^{(j)}, \mathbf{S}_k^{(j)}\Big), \tag{81}$$

$$\tilde{\mu}_k^{(j)} = \Lambda_k^{(j)} \sum_{i=1}^{r} \pi_{ij}\,\mu_{k-1}^{(i)}, \tag{82}$$

$$\mu_k^{(j)} = \frac{\tilde{\mu}_k^{(j)}}{\sum_{\ell=1}^{r} \tilde{\mu}_k^{(\ell)}}. \tag{83}$$

Here, $\Lambda_k^{(j)}$ is the likelihood of the new measurement under model j, and $\mu_k^{(j)}$ is the new normalized probability that model j is active at time k.

4) Combination: Finally, the IMM outputs a single fused estimate of the state by combining each model's updated estimate:

$$\hat{\mathbf{x}}_{k|k} = \sum_{j=1}^{r} \mu_k^{(j)}\,\hat{\mathbf{x}}_{k|k}^{(j)}, \tag{84}$$

$$\mathbf{P}_{k|k} = \sum_{j=1}^{r} \mu_k^{(j)}\Big[\mathbf{P}_{k|k}^{(j)} + \big(\hat{\mathbf{x}}_{k|k}^{(j)} - \hat{\mathbf{x}}_{k|k}\big)\big(\hat{\mathbf{x}}_{k|k}^{(j)} - \hat{\mathbf{x}}_{k|k}\big)^{\top}\Big]. \tag{85}$$

Intuitive Understanding

The IMM algorithm maintains a *mixture* of different motion models, each producing its own state estimate. By appropriately mixing the models based on their transition probabilities and the measurement likelihoods, the IMM can adapt to changes in the target's motion mode. This approach is especially powerful in *maneuvering target tracking*, where a target's motion may switch unpredictably among a finite set of dynamic regimes.