

Maneuvering road object tracking with the IMM algorithm and an imaging radar sensor

William Sitch

Robotics and Autonomous Systems

Graduate Program

Stanford School of Engineering

Stanford, USA

wsitch@stanford.edu

Abstract— This report summarizes how an interacting multiple model (IMM) algorithm uses measurements from an imaging radar sensor to estimate the dynamic state of a maneuvering road object with unknown control inputs. The IMM algorithm mixes the outputs from Kalman (KF) and unscented Kalman (UKF) filters running constant velocity (CV), constant acceleration (CA), and coordinated turn (CT) motion models. Simulations are used to validate IMM filter theory and determine appropriate model parameters for real-world testing. A prototype imaging radar sensor was used to capture measurements from a controlled scenario; using these data, the tuned IMM filter is shown to accurately track vehicle state.

Keywords—*Interacting multiple model, Kalman filter, nonlinear filtering, constant velocity motion model, constant acceleration motion model, coordinated turn rate motion model, imaging radar sensor, radar measurements, radar tracking, object tracking*

I. INTRODUCTION

Target tracking systems for objects (primarily vehicles, cyclists, and pedestrians) maneuvering on the road are a critically important component of developing autonomous self-driving vehicles. Accurate estimation of other road users' position and velocity is necessary for safe maneuver planning [1]. Radar is a suitable long-range sensor for this application [2].

A. Objective

The project objective is to explain and demonstrate how the interacting multiple model (IMM) filter algorithm can be used with an imaging radar sensor to track maneuvering road objects.

The implementation steps taken were:

1. Simulate various scenarios, test filter theory, determine appropriate model parameters for real-world testing
2. Validate algorithms with real radar measurements

II. BACKGROUND

A. State Estimation

State estimation is commonly performed with Bayesian filters [1], where for an object described by state x at time k , the state transition function predicts the next value of the state

$$x_{k+1} = f(x_k, u_k, v_k)$$

where u_k is the control input and v_k is the process noise that describes the uncertainty of the model. The measurement function relates sensor measurements z_k and associated noise w_k with the state as follows:

$$z_k = h(x_k, w_k)$$

Assuming the state can be described by a Markov chain, where the next state is a function only of the previous state and measurements, the state can be estimated by the recursive prediction and update steps of the Bayes filter.

The Kalman filter (KF) [3] leverages the ease-of-use and low computational complexity of manipulating Gaussians for probability estimation when the transition and measurement functions are linear. State estimates are described by Gaussian distributions with mean and covariance values, and process and measurement noises are assumed to be zero-mean Gaussian white noise with covariance matrices given by Q and R .

Nonlinear transition and measurement functions are incompatible with the KF, and several different derivative filters are commonly used. The extended Kalman filter (EKF) leverages Taylor-series expansions to locally linearize nonlinear functions $f(x)$ and $h(x)$, but only first-order terms are considered [4]. Jacobians of the state transition function and measurement function are used; they describe the rate of change of the new state (at time $k + 1$) with respect to the old state (at time k) and have the form (for example for state transition function f):

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \dots & \frac{\partial f}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

The Jacobians can be challenging to calculate for complex systems, and the limitations of a first-order approximation can produce errors if the system changes rapidly near the point of state estimation [5]. The unscented Kalman filter (UKF) addresses these shortcomings by using a weighted average of multiple points that approximate the nonlinear model [6]. Sigma points are selected around the state estimate mean, arranged as a function of the state covariance. The sigma points are transformed by the nonlinear function, and the resulting points are used to calculate the next state estimated mean and covariance. This algorithm typically has higher accuracy than the EKF but is computationally less efficient [4].

B. Target Tracking

Target tracking is an application of state estimation applied to maneuvering (actively controlled) objects. The state transition function is called the motion model, and it describes the motion exhibited by an object as a function of the current state and control inputs.

For tracking applications, however, the object's motion model and control inputs are unknown. The object could be a car or truck with two-wheel Ackermann steering, a motorcycle with single-

wheel bicycle steering, or a robot with nonholonomic differential two-wheel drive steering. The tracker also doesn't know the object's control inputs – whether it will continue at a constant speed, accelerate, or turn.

Motion models that can be applied to any type of maneuvering object track movement independent of the steering mechanism and object bearing: these states of these models are the object's position, velocity, acceleration, and turning rate.

A single motion model must describe any motion that the object will exhibit, which is generally unacceptable for tracking applications [7]. Consider, for example, the error associated with trying to predict the position of a vehicle accelerating through a corner with a model designed for tracking straight-line movement at a constant speed.

C. Motion Models

The following motion models have been validated with real-world measurement data [8] and are commonly used for tracking purposes [9, 8, 10, 7]. The motion models are described as a transition matrix with associated process noise covariance matrix, which is applied to the state transition equation as:

$$x_{k+1} = Fx_k + v_k, \text{ where } v_k = \text{GWN}(0, Q)$$

1) Constant Velocity

The constant velocity (CV) motion model is appropriate for objects traveling in a straight line (not necessarily aligned with the Cartesian coordinates) at a constant velocity. The model estimation states are $[x \dot{x} y \dot{y}]$, where x refers to the longitudinal position and y refers to the lateral position, and \dot{x} and \dot{y} refer to velocities in the respective directions. The state transition matrix is:

$$F_{CV} = \begin{bmatrix} F'_{CV} & 0_{2 \times 2} \\ 0_{2 \times 2} & F'_{CV} \end{bmatrix}, F'_{CV} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}$$

where δt is the amount of time since the last measurement was processed. The process noise covariance matrix is

$$Q_{CV} = q \begin{bmatrix} Q'_{CV} & 0_{2 \times 2} \\ 0_{2 \times 2} & Q'_{CV} \end{bmatrix}, Q'_{CV} = \begin{bmatrix} \frac{\delta t^4}{4} & \frac{\delta t^3}{2} \\ \frac{\delta t^3}{2} & \delta t^2 \end{bmatrix}$$

where q is a model tuning factor.

2) Constant Acceleration

The constant acceleration (CA) motion model is appropriate for objects accelerating in a straight line (not necessarily aligned with the Cartesian coordinates). The model estimation states are $[x \dot{x} \ddot{x} y \dot{y} \ddot{y}]$, where \ddot{x} and \ddot{y} refer to acceleration in the respective directions.

The state transition matrix is:

$$F_{CA} = \begin{bmatrix} F'_{CA} & 0_{3 \times 3} \\ 0_{3 \times 3} & F'_{CA} \end{bmatrix}, F'_{CA} = \begin{bmatrix} 1 & \delta t & \frac{\delta t^2}{2} \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix}$$

and the process noise covariance matrix is

$$Q_{CA} = q \begin{bmatrix} Q'_{CA} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q'_{CA} \end{bmatrix}, Q'_{CA} = \begin{bmatrix} \frac{\delta t^5}{20} & \frac{\delta t^4}{8} & \frac{\delta t^3}{6} \\ \frac{\delta t^4}{8} & \frac{\delta t^3}{3} & \frac{\delta t^2}{2} \\ \frac{\delta t^3}{6} & \frac{\delta t^2}{2} & \delta t \end{bmatrix}$$

3) Coordinated Turn

The coordinated turn (CT) model is appropriate for objects moving at a constant velocity and turning at a constant rate. The model estimation states are $[x \dot{x} y \dot{y} \omega]$, where ω refers to the turn rate.

The state transition matrix is:

$$F_{CT} = \begin{bmatrix} 1 & \frac{\sin(\omega \delta t)}{\omega} & 0 & -\frac{1 - \cos(\omega \delta t)}{\omega} & 0 \\ 0 & \cos(\omega \delta t) & 0 & -\sin(\omega \delta t) & 0 \\ 0 & \frac{1 - \cos(\omega \delta t)}{\omega} & 1 & \frac{\sin(\omega \delta t)}{\omega} & 0 \\ 0 & \sin(\omega \delta t) & 0 & \cos(\omega \delta t) & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and the process noise covariance matrix is

$$Q_{CT} = q \begin{bmatrix} Q'_{CT} & 0_{2 \times 2} & 0 \\ 0_{2 \times 2} & Q'_{CT} & 0 \\ 0 & 0 & \sigma_{werr}^2 \end{bmatrix}, Q'_{CT} = \begin{bmatrix} \frac{\delta t^4}{4} & \frac{\delta t^3}{2} \\ \frac{\delta t^3}{2} & \delta t^2 \end{bmatrix}$$

where σ_{werr}^2 is the initial uncertainty for the turn rate.

D. Interacting Multiple Model Algorithm

The IMM algorithm overcomes the single-model limitation of the various Kalman filters by combining state estimations from several different filters with different motion models. The IMM determines the probability each model tracks the target's motion, and then builds a state estimate through a weighted average of the model outputs [9]. The blended state estimate is used by all models on the next iteration, which is the "interacting" component of the

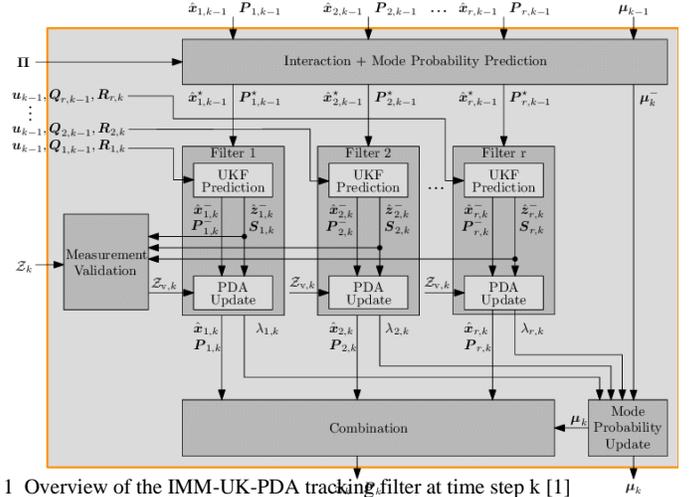


Fig. 1 Overview of the IMM-UK-PDA tracking filter at time step k [1]

IMM. This mixing strategy accounts for the unknown control inputs and time-varying dynamics of the target [1, 9, 11]. It has been proven to perform better than any single model algorithm in complex tracking problems [12].

An IMM algorithm using UKF prediction is shown in Fig. 1, described in the sections below, and detailed more thoroughly in [1, 9, 13].

1) Interaction + Mode Probability Prediction

The posterior (mean $\hat{x}_{j,k-1}$ and covariance $P_{j,k-1}$) state of each of the r filters (where $i, j \in \{1, 2, \dots, r\}$) from the previous time step $k-1$ interact with each other as weighted mixing functions:

$$\hat{x}_{j,k-1}^* = \sum_{i=1}^r \mu_{i|j,k} \hat{x}_{i,k-1}$$

$$P_{j,k-1}^* = \sum_{i=1}^r \mu_{i|j,k} \left[P_{i,k-1} + (\hat{x}_{i,k-1} - \hat{x}_{j,k-1}^*)(\hat{x}_{i,k-1} - \hat{x}_{j,k-1}^*)^T \right]$$

where $\mu_{i|j,k}$ is not the mean (yes, this is confusing terminology, but is consistent in literature) but rather describes the model

$$\hat{x}_k = \sum_{j=1}^r \mu_{j,k} \hat{x}_{j,k}$$

$$P_k = \sum_{j=1}^r \mu_{j,k} \left[P_{j,k} + (\hat{x}_{j,k} - \hat{x}_k)(\hat{x}_{j,k} - \hat{x}_k)^T \right]$$

Table 7. A comparison of the commonly employed sensors in self-driving cars; camera, LiDAR, and radar, based on technical characteristics and other external factors. The “✓” symbol indicates that the sensor operates competently under the specific factor. The “~” symbol indicates that the sensor performs reasonably well under the specific factor. The “✗” symbol indicates that the sensor does not operate well under the specific factor relative to the other sensors.

Factors	Camera	LiDAR	Radar	Fusion
Range	~	~	✓	✓
Resolution	✓	~	✗	✓
Distance Accuracy	~	✓	✓	✓
Velocity	~	✗	✓	✓
Color Perception, e.g., traffic lights	✓	✗	✗	✓
Object Detection	~	✓	✓	✓
Object Classification	✓	~	✗	✓
Lane Detection	✓	✗	✗	✓
Obstacle Edge Detection	✓	✓	✗	✓
Illumination Conditions	✗	✓	✓	✓
Weather Conditions	✗	~	✓	✓

Fig. 2 Feature comparison of sensors used in self-driving cars

probability prediction, which is the weight by which the estimate from the previous time step of filter i is applied to the new filter j at time k , given by:

$$\mu_{i|j,k} = \frac{\Pi_{ij} \mu_{i,k-1}}{\sum_{i=1}^r \Pi_{ij} \mu_{i,k-1}}$$

where $\mu_{i,k-1}$ is the probability that model i best represents recent measurements, and Π_{ij} is a pre-determined transition probability matrix that defines how frequently we expect models to transition to one another:

$$\Pi = \begin{bmatrix} p_{11} & \cdots & p_{r1} \\ \vdots & \ddots & \vdots \\ p_{1r} & \cdots & p_{rr} \end{bmatrix}$$

where p_{ij} is the probability that motion described by model i will be followed by motion described by model j .

2) Filtering

Each filter in the IMM starts an estimate iteration with the mixed input state and makes predictions and updates independently. Measurements used for the update step are validated to be close enough to the state estimation mean as calculated by the Mahalanobis distance, which uses the innovation covariance matrix $S_{j,k}$ to determine how many standard deviations away the measurement is from the estimated mean.

3) Model Probability Update

The model probabilities for each filter, which reflect how well the measurements fit the model, are updated as follows:

$$\mu_{i,k} = \frac{N(v_{i,k}, 0, S_{i,k}) \mu_{i,k-1}}{\sum_{j=1}^r N(v_{j,k}, 0, S_{j,k}) \mu_{j,k-1}}$$

where $v_{i,k}$ is the measurement residual (the difference between the measurement and the updated state estimation), and $S_{i,k}$ is the covariance.

4) Combination

The mean and covariance output from the IMM filter is a weighted combination of the individual filter outputs as follows:

III. MEASUREMENT SENSORS

Various sensors are used to detect and track road objects, primarily camera, lidar, and radar. Each sensing modality has advantages and disadvantages, which can be generally summarized as shown in Fig. 2 from [14].

A. Imaging Radar Sensor Technology

New “imaging” radar sensors with higher numbers of virtual transmit and receive antennas are approaching lidar-like angular resolution, but without the range, cost, and environmental limitations of lidar. This type of sensor is ideal for long-range target tracking.

This project used a prototype long-range imaging radar (LRR) sensor from Uhnder, Inc., an Austin, TX-based startup company that builds a digital code modulation (DCM) radar-on-chip (RoC) and a radar sensor designed for ADAS and self-driving vehicle applications. This LRR sensor can resolve a driver walking beside a vehicle at ranges approaching 300m.

B. Imaging Radar Sensor Output

The Uhnder imaging radar sensor transmits and receives digitally modulated signals in the 77-81GHz automotive radar band. The sensor’s on-chip signal processing pipeline analyzes the received signals and generates various outputs at a typical rate of 10Hz. The outputs include a minimally processed point cloud, filtered detections, and further filtered clusters of detections that represent objects that reflect energy within the sensor’s field of view (FoV).

The data associated with each point are range, azimuth angle, elevation angle, magnitude, signal-to-noise ratio (SNR), and target doppler information. The sensor performs a static/dynamic differentiation based on the sensor’s ego-velocity, such that each point is identified as stationary or moving in the world coordinate frame.

The volume of data from the sensor is suitable for low-level data fusion with lidar, or integration into a world model built with camera input. For the purposes of this project, however, the point cloud data are too voluminous to use. The project will focus on tracking moving objects in the sensor FoV by considering dynamic detection output data, which consists of as many as 50 points per object at ranges below 80m, and 1-5 points per object at ranges extending to 320m.

C. Simulating the Radar Sensor

During the first phase of the project, which consisted of validating the IMM filter theory with simulation, it was important to also simulate the radar sensor as a measurement system. Typical measurement models used in simulation reference the Cartesian coordinates of the simulated object and add some Gaussian white noise to emulate measurement uncertainty. The measurement noise R is typically a diagonal matrix because the noise in x and y dimensions is independent.

Radar sensors, however, measure energy reflections from a scene in spherical coordinates. The range r , azimuth angle α , and

elevation angle (which is not considered in our 2D example) measured by the sensor impact the shape of the noise, and when the resulting measurement is converted into Cartesian coordinates the noise in x and y dimensions is not independent, as shown in Fig. 3 from [15].

The measurement noise matrix is:

$$R = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix}$$

where

$$\begin{aligned} \theta &= \frac{\pi}{2} - \alpha \\ \sigma_x^2 &= \sigma_r^2 \cos^2(\theta) + r^2 \sigma_a^2 \sin^2(\theta) \\ \sigma_y^2 &= \sigma_r^2 \sin^2(\theta) + r^2 \sigma_a^2 \cos^2(\theta) \\ \sigma_{xy}^2 &= \frac{1}{2} \sin^2(2\theta) [\sigma_r^2 - r^2 \sigma_a^2] \end{aligned}$$

IV. TRACKER IMPLEMENTATION DETAILS

A. Complexity Reduction Strategy

Several assumptions and simplifications were taken to reduce the complexity of the project to ensure it could be completed during the timeline of the AA273 course. These decisions, and the technical challenges they avoid, include:

- Operating the tracker as a stationary observer; avoiding the requirement of low-latency ego-velocity measurement and state estimation for the observer position, velocity, acceleration, and turn rate, also avoiding reference plane translation
- Operating only a single radar sensor; avoiding the complexity of sensor fusion (especially challenging across different sensing modalities)
- Evaluating only one maneuvering object within the scene; avoiding the challenges associated with multiple tracked objects, including track management and joint probabilistic data association
- Evaluating objects with generalized 2D motion models; avoiding the challenges of object classification, object-specific motion models, and tracking in the third z-dimension
- Evaluating only simple scenarios without excessive ground clutter; reducing false-positive measurements
- Simplifying unequal state dimensions by augmenting with zero information; avoiding a more complicated implementation of the IMM filter (this is detailed more in Section IV.B)

B. State

The objective of the project is to track the state of a maneuvering road object, which – given we are limiting ourselves to tracking only in x and y dimensions – means we want to know the object’s position, speed, acceleration, and turning rate. However, the motion models described in Section II.C have unequal state dimensions. The CV model, for example, has state transfer equations for four state variables, while the CT model has equations for seven state variables.

[16] addresses a systematic approach to IMM mixing for unequal dimension states, and while the paper recommends an adjustment to the IMM algorithm to augment values that are unused in some models with a mixing term, the performance impact of doing so is minimal in a tracking application where noise terms are relatively large. For this reason, models are augmented with zeros for states that aren’t referenced in the model. For example, the turn rate ω is zeroed in the CV and CA models.

C. IMM Parameters

Design parameters for the IMM filter were selected to improve the performance of the filter in the simulation phase of the project. These parameters were then used in the measurement phase.

1) Transition Probability Matrix

As detailed in Section II.D.1), the transition probability matrix determines how frequently we expect models to transition to one another. Based on implementation notes from [11, 7], a range of values were trialed, and relatively infrequent model switching was found to be ideal. Some small benefit was found for staying in the CT model slightly less frequently than the CV and CA models. This project used the following matrices for two- and three-model IMM filters:

$$\Pi = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}, \text{ for CV, CA IMM filters}$$

$$\Pi = \begin{bmatrix} 0.96 & 0.02 & 0.02 \\ 0.02 & 0.96 & 0.02 \\ 0.04 & 0.4 & 0.92 \end{bmatrix}, \text{ for CV, CA, CT IMM filters}$$

2) Process Noise

The process noise of the KF and other filters used in the IMM filter determine how quickly the models react to measurements that deviate from the predicted motion. A low process noise lags a change in motion, but reduces the measurement noise present in the system. Conversely, large process noise adapts to changes more quickly, but don’t reduce the measurement noise.

To optimize the q value to provide both acceptable measurement noise reduction and lag-free reaction to changes in the object’s exhibited dynamics, Monte Carlo simulations were run on the simulated scenarios. q values between 0.1 and 10 were used for the different models. However, as noted in [11], the ability of the IMM filter to accurately track the changing dynamics of a target is primarily determined by how closely the filter motion models describe the target dynamics.

3) Measurement Validation (Gating)

Measurement gating was performed within the range of 5 Mahalanobis-adjusted standard deviations of the innovation covariance.

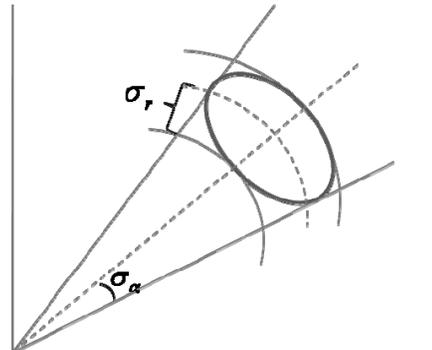


Fig. 3 Bearing noise (σ_a) and range noise (σ_r) of a radar sensor

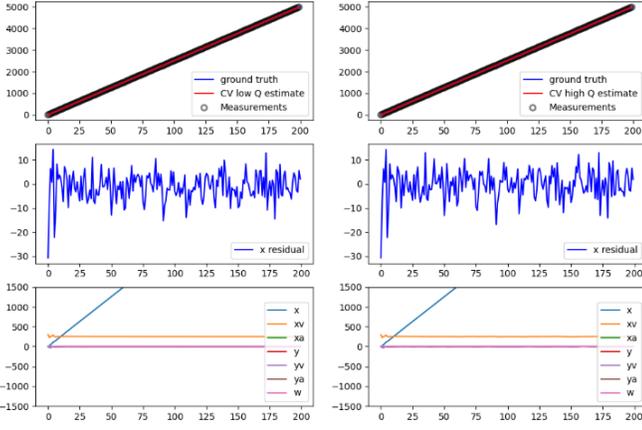


Fig. 4 KF with CV model, CV 1D simulation

D. Multiple Measurements

As detailed in Section III.B, the imaging radar sensor measures multiple detections from the same object as a function of the small angular resolution of the sensor. Each detection is an independent measurement, and the combination of all detections best describes the object's motion.

The IMM filter algorithm iterates when new measurements are received. δt is calculated as the time since the previous batch of measurements and a single filter predict step occurs. Each detection is then processed as a separate measurement with a separate filter update step. The final filter state estimate is therefore averaged across all measurements, and the IMM algorithm then further averages the output from the combinations of filters.

Fig. 5 KF with CA model, CV + CA 1D simulation

V. RESULTS

A. Simulation Results

1) KF with CV model, CV 1D simulation

The simulation was an object moving with a constant velocity in one dimension. The KF was used with the CV model. Innovation residuals show that both low-Q ($q=0.01$) and high-Q ($q=10$) models accurately predict the motion and accurately estimate the object state. Results shown in Fig. 4.

2) KF with CA model, CV + CA 1D simulation

The simulation was an object moving with varying velocity in one direction. The KF was used with the CA model. The low-Q ($q=0.01$) model was unable to react quickly enough to the object dynamics; the high-Q ($q=10$) model more accurately predicted the motion. Neither model perfectly estimated the object state. Results shown in Fig. 5.

3) KF with CA model, CV 2D simulation with U-turn

This simulation was an object moving with constant speed in a straight, u-turn, and straight motion. The KF was used with the CA model. Both low-Q ($q=0.01$) and high-Q ($q=10$) models were unable to predict the motion and lagged the measurements. Results shown in Fig. 6.

4) UKF with CT model, CV 2D simulation with U-turn

This simulation was an object moving with constant speed in a straight, u-turn, and straight motion. The UKF was used with the CT model. Both low-Q ($q=0.01$) and high-Q ($q=10$) models

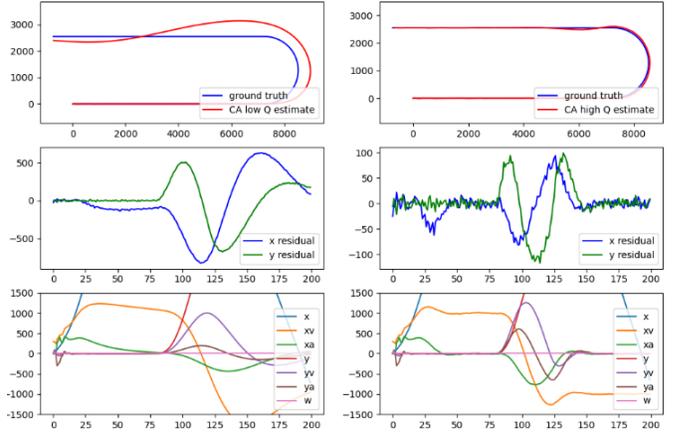


Fig. 6 KF with CA model, CV 2D simulation with U-turn

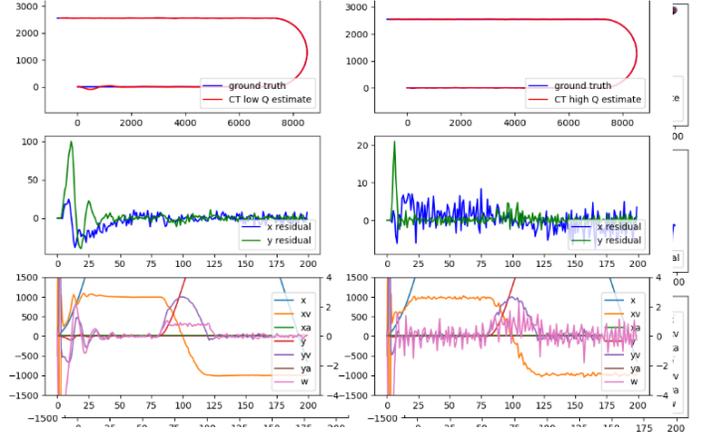


Fig. 7 UKF with CT model, CV 2D simulation with U-turn

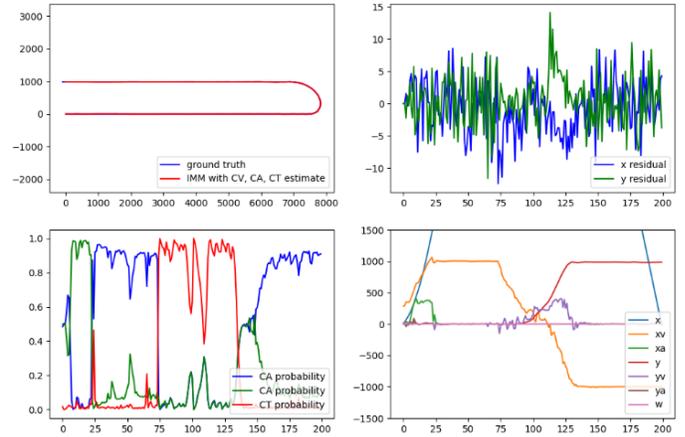


Fig. 8 IMM filter with CV, CA, and CT models, CV + CA 2D simulation with U-turn

were able to predict the motion but introduced some significant noise in the state estimates. Results shown in Fig. 7.

5) IMM filter with CV, CA, and CT models, CV + CA 2D simulation with U-turn

This simulation was the most complex scenario, it consisted of an object moving with varying velocity in a straight, u-turn, and straight motion. The IMM filter was used with a KF running the CV model ($q=0.1$), a KF running the CA model ($q=10$), and a UKF running the CT model ($q=1$). The residuals and the mode probability graph demonstrate that the model was able to accurately predict the object's motion, and the IMM model probability reflected the scenario setup (acceleration, a constant-

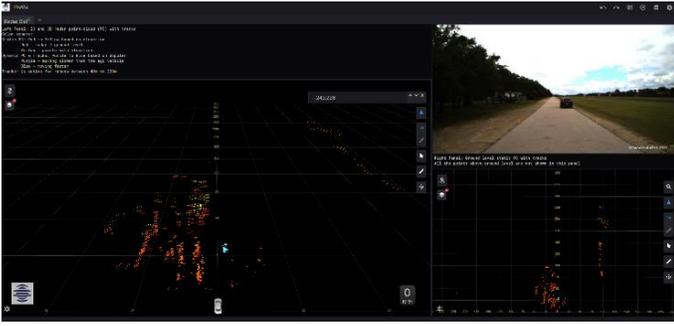


Fig. 9 Proviz rosbag visualization tool showing measurement scenario

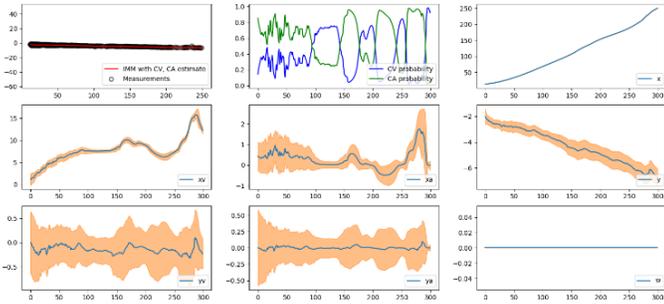


Fig. 10 IMM filter with CV and CA models, measurement data from Uhnder imaging radar sensor

speed section, a U-turn, acceleration, and a constant-speed section). Results shown in Fig. 8.

B. Measurement Results

The Uhnder prototype radar sensor was used to collect measurement data from a private airport in Austin, TX. The sensor was mounted on a stationary vehicle with an onboard compute platform that captured measurement data in a rosbag. The scenario featured two vehicles, initially located 10m and 600m from the measurement system. The vehicles accelerate towards each other and pass each other at approximately 200m from the observer, 27 seconds after starting. The data ends after 33.2 seconds when the first vehicle passes 250m and is shadowed by the second vehicle.

Fig. 9 shows the measurement data captured by the measurement system at time $t \sim 5s$. The main screen on the left is showing a significant amount of clutter returns associated with the trees on the left-hand side of the access road. These measurements were identified as being related to static objects and were not considered by the IMM. The bright cyan dots identify the moving vehicle. The camera view in the top right shows the first vehicle moving away from the observer. Approximately 500,000 radar measurements are made during the time period, of which 710 were associated with the first vehicle.

The data were extracted from the rosbag and used in an IMM with a KF running the CV model ($q=0.1$) and a KF running the CA model ($q=10$). The CT model was unnecessary given the straight-line motion of the vehicles in the scenario. Results shown in Fig. 10 demonstrate that the IMM filter was able to correctly identify the motion model appropriate for the movement of the vehicle: a period of slow acceleration for approximately 10 seconds, a period of constant speed for 6 seconds, a small acceleration and deceleration, and an acceleration when passing the other vehicle.

VI. CONCLUSIONS

This project was able to successfully demonstrate that the IMM filter with CV, CA, and CT models was able to track simulated

maneuvering road objects with simulated radar measurements. The tracking demonstrated low residual error between predicted state estimates and measurements, appropriate motion model selection for different types of simulated motion, and good matching between state estimate and simulated ground truth.

Real-world imaging radar sensor measurements were also successfully integrated into the IMM filter framework with a CV and CA motion model, and the filter tracked the movement of a maneuvering road object under controlled conditions. Ground truth data for the real-world measurements were unavailable; ad hoc observation suggests the filter performance reflects real-world behaviour.

ACKNOWLEDGMENT

W.S. thanks Uhnder, Inc. colleagues Uday Gurnani and Jonathan Preussner for assistance with gathering and processing radar data rosbags from various scenarios, and Uhnder colleagues Sunil Chomal and Aalok Acharya for advice related to tracker implementation.

VII. REFERENCES

- [1] Y. Bar-Shalom, X. Li and T. Kirubarajan, Estimation with Applications to Tracking and Navigation, New York: John Wiley & Sons, 2001.
- [2] A. Farina, G. Battistelli, L. Chisci and A. Di Lallo, "40 years of tracking for radar systems: a cross-disciplinary academic and industrial viewpoint," in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, Chiang Mai, Thailand, 2017.
- [3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME--Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35-45, 1960.
- [4] P. Abbeel, "EKF, UKF," University of California, Berkeley, [Online]. Available: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa13/slides/EKF_UKF.pdf.
- [5] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, FL, USA, 1997.
- [6] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 93, no. 3, pp. 401-422, 2004.
- [7] S. Sarkar and A. Roy, "Interacting Multiple Model (IMM) algorithm for road object tracking using automotive radar," in *11th International Radar Symposium India*, Bangalore, 2017.
- [8] R. Schubert, E. Richter and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *2008 11th International Conference on Information Fusion*, Cologne, Germany, 2008.
- [9] Y. Bar-Shalom, K. C. Chang and H. A. P. Blom, "Tracking a Maneuvering Target Using Input Estimation Versus the Interacting Multiple Model Algorithm," *IEEE Trans. Aerosp. Electronic Syst. EAS-25(2)*, pp. 296-300, 1989.
- [10] T. Kirubarajan, Y. Bar-Shalom and D. Lerro, "Bearings-Only Tracking of Maneuvering Targets Using a Batch-Recursive Estimator," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, July 2001.
- [11] A. F. Genovese, "The Interacting Multiple Model Algorithm for Accurate State Estimation of Maneuvering Targets," *Johns Hopkins APL Technical Digest*, vol. 22, no. 4, pp. 614-623, 2001.
- [12] N. Cui, L. Hong and J. Layne, "A comparison of nonlinear filtering approaches with an application to ground target tracking," *Signal Process.*, vol. 85, pp. 1469-1492, 2005.
- [13] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, pp. 1255-1321, October 2005.
- [14] D. J. Yeong, G. Velasco-Hernandez, J. Barry and J. Walsh, "Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review," *Sensors*, vol. 21, no. 2140, 2021.
- [15] P. Agarwal, D. Federman, K. Ge, T.-H. Jan, C. Jones, Y. Lou, J. Ma, S. Modi, K. Shi, V. Tan and M. Zhang, "TARGET TRACKING:

IMPLEMENTING THE KALMAN FILTER," New Jersey Governor's School in the Sciences at Drew University, New Jersey, 2005.

- [16] K. Granstrom, P. Willett and Y. Bar-Shalom, "Systematic approach to IMM mixing for unequal dimension states," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 2975-2986, October 2015.
- [17] Y.-S. Kim and K.-S. Hong, "An IMM Algorithm for Tracking Maneuvering Vehicles in an Adaptive Cruise Control Environment," *International Journal of Control, Automation, and Systems*, vol. 2, no. 3, pp. 310-318, 2004.

- [18] K. Brown, K. Driggs-Campbell and M. J. Kochenderfer, "A Taxonomy and Review of Algorithms for Modeling and Predicting Human Driver Behavior," *arXiv:2006.08832v3*, 2020.
- [19] M. Schreier, V. Willert and J. Adamy, "Compact Representation of Dynamic Driving Environments for ADAS by Parametric Free Space and Dynamic Object Maps," *IEEE Transactions on Intelligent Transportation Systems*, pp. 367-384, February 2016.