Hindawi
Journal of Advanced Transportation
Volume 2019, Article ID 7496017, 13 pages
https://doi.org/10.1155/2019/7496017

*Research Article*

# Sensitivity and Performance Evaluation of Multiple-Model State Estimation Algorithms for Autonomous Vehicle Functions

**Olivér Törő**,[1] **Tamás Bécsi**,[1] **Szilárd Aradi**,[1] **and Péter Gáspár**[2]

[1] *Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, Stoczek u.2., H-1111 Budapest, Hungary*
[2] *Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u. 13-17, H-1111 Budapest, Hungary*

Correspondence should be addressed to Tamás Bécsi; becsi.tamas@mail.bme.hu

Received 15 December 2018; Revised 14 March 2019; Accepted 17 March 2019; Published 4 April 2019

Academic Editor: Yair Wiseman

Robust object tracking and maneuver estimation methods play significant role in the design of advanced driver assistant systems and self-driving cars. As an input to situation understanding and awareness, the performance of such algorithms influences the overall effectiveness of motion planning and plays high role in safety. The paper examines the suitability of different probabilistic state estimation methods, namely, the Extended Kalman Filter (EKF) and the more general Particle Filter (PF) with the addition of the Interacting Multiple Model (IMM) approach. These algorithms are not capable of predicting motion for long term in road traffic conditions, though their robustness and model classification capability are essential for the overall system. The performance is evaluated in road traffic scenarios where the tracked object imitates the motion characteristics of a road vehicle and is observed from a stationary sensor. The measurements are generated according to standard automotive radar models. The analysis conducted along two aspects emphasizes the different performance and scaling properties of the examined state estimation algorithms. The presented evaluation framework serves as a customizable method to test and develop advanced autonomous functions.

## 1. Introduction

Nowadays, highly automated driver assistance systems and autonomous vehicles are in focus of attention and pose many different challenges that need to be solved. The architecture of the motion planning has multiple layers from route planning, through behavioral and tactical planning to local control [1]. The perception subsystem consists of vehicle and environment sensors, data fusion and tracking layers, behavioral classification and prediction tasks [2]. These layers need to work together to fulfill the original task of the vehicle, namely, to reach its destination with the possible minimization of journey time though with respect to traffic rules, passenger comfort, and, most importantly, to safety. The local and short term decisions require the autonomous vehicle to have some ability to reason about the future motion of surrounding vehicles [3]. This leads to the problem of behavior prediction, where the ego vehicle needs to predict the possible future trajectories of the surrounding traffic participants, such as vehicles or pedestrians [4]. Classic object tracking algorithms are not suitable for mid-term motion prediction because they cannot consider the interaction of participants, though their robustness is essential for the proper input generation for these algorithms. The classification ability of multiple model (MM) systems can also enhance the efficiency of the prediction [5].

A maneuvering vehicle can be effectively tracked by using a multimodel state estimator [6]. The interacting multiple model (IMM) estimator is an approximate solution of the general multimodel problem [7, 8], which is computationally tractable with linear scaling in the number of the considered models. The IMM estimator was originally proposed with Kalman filters [9]; however, it can be realized by particle filters [10, 11] or fused with random set filtering methods. In [12, 13] IMM estimation, realized as a particle filter, was presented which is able to work in a cooperative road traffic situation. A random finite set based particle filtering in the IMM framework for cooperative road traffic application was presented in [14]. The choice of coordinate system for maneuvering target tracking affects the structure of the estimation algorithm and the achieved performance [15, 16]. Plenty of research deals with the problem that which object tracking, or in more general, probabilistic state estimation algorithm is suitable for these purposes. One approach is to examine to what extent the process and measurement noise affect the position estimates. The authors of [17] concluded that if the effect of the process noise to measurement noise ratio is above 0.5, the IMM offers better estimates. While the study in [18] states that the performance differences are highly dependent on the maneuvers of the target. This solution assumed a so-called perfect IMM (PIMM), which is a lower bound error estimation, since it never fails to choose the appropriate model. Naturally the PIMM is an ideal model and was generated artificially. Using nonideal IMM, our study examines the effects of the hyperparameter tuning of the IMM, in a road vehicle-like environment, where the model for the sensor imitates the capabilities and performance of radar sensing.

The paper is organized as follows. Section 2 introduces the probabilistic state estimation problem, and the possible solutions, such as Kalman filter, Extended Kalman Filter, Particle Filtering, and IMM methods. Section 3 gives an overview on the scenario and models used for the evaluation of the algorithms, and, finally, Section 4 evaluates the performance of the above-mentioned methods.

## 2. Problem Statement

In probabilistic estimation the state is considered as a random variable and its distribution is approximated. The approximation can be performed with various precision; however, the underlying mathematical structure, the Bayes theorem, can give a unified description of the different methods. The general problem is to estimate the state $\mathbf{x}$ given measurements $\mathbf{z}$ and the model of the considered dynamic system. The model incorporates the time evolution of the system, which is the motion model and the measurement model, which accounts for how we get information from the system through sensors. Formally the estimation at timestep $k$ involves the evaluation of the formula

$$p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k}\right) = \frac{g_k\left(\mathbf{z}_k \mid \mathbf{x}_k\right) p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\right)}{\int g_k\left(\mathbf{z}_k \mid \mathbf{x}_k\right) p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\right) \mathrm{d}\mathbf{x}_k} \quad (1)$$

where the conditional probability $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ is the posterior density of the state in question. The likelihood $g_k(\mathbf{z}_k \mid \mathbf{x}_k)$ can be constructed in the knowledge of the measurement model. The denominator equals the marginal Probability density function (PDF) $p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})$ which is called the evidence and serves as a normalizing factor. The prior term $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ can be expressed with the Chapman-Kolmogorov equation

$$\begin{aligned} & p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\right) \\ & \quad = \int \varphi_{k|k-1}\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}\right) \mathrm{d}\mathbf{x}_{k-1} \end{aligned} \quad (2)$$

where $\varphi_{k|k-1}(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ is the state transition density constructed from the motion model. The estimation can be carried out in a recursive manner; however, there are no general analytic solution because of the involved integrals. In case of a linear system model and additive Gaussian noise (1) can be shown to reduce to the Kalman filter [19].

```
Initialize variables: $\mathbf{x}_0$, $\mathbf{P}_0$
for $k = 1$ to $N_k$ do
  Predict
    $\mathbf{F}_k = (\partial f(\mathbf{x})/\partial \mathbf{x})|_{\mathbf{x}_{k-1|k-1}}$ {Compute Jacobian}
    $\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1})$ {Predict state}
    $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$ {Predict covariance}
  Update
    $\mathbf{H}_k = (\partial h(\mathbf{x})/\partial \mathbf{x})|_{\mathbf{x}_{k|k-1}}$ {Compute Jacobian}
    $\mathbf{r}_k = \mathbf{z}_k - h(\mathbf{x}_{k|k-1})$ {Residual}
    $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$ {Residual covariance}
    $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$ {Kalman gain}
    $\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{r}_k$ {Update state}
    $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$ {Update covariance}
end for
```

Algorithm 1: Extended Kalman filter.

Beside the fact that the Kalman filter is the optimal minimum mean squared error filter for linear-Gaussian systems, its usage is limited by the presuppositions of the model. Most real-life systems show nonlinearities and the Gaussian presumption cannot always be held. An approximate method is the Extended Kalman Filter (EKF) which, to handle nonlinearities, linearizes the system equations by first order Taylor expansion around the current state.

A general discrete time state space system model with additive noise has the form

$$\mathbf{x}_{k+1} = f_k\left(\mathbf{x}_k\right) + \mathbf{w}_k \quad (3)$$

$$\mathbf{z}_k = h_k\left(\mathbf{x}_k\right) + \mathbf{v}_k \quad (4)$$

where $f$ is the state transition function, $h$ is the measurement function, and $\mathbf{w}$ and $\mathbf{v}$ are noise vectors. If the nonlinearities in $f$ and $h$ are not too strong, the linearization approximates the functions well and the EKF gives good results; convergence however cannot be guaranteed. The structure of the EKF is shown in Algorithm 1.

A more general approach to handle nonlinearities or even non-Gaussian PDFs is to use the particle filter (PF), which is an umbrella term. It corresponds to a family of Monte Carlo sampling based sequential algorithms. It is a numerical method that approximates a function, in our context the posterior PDF by particles. The particles are weighted samples, drawn from the distribution; hence, the higher the particle number, the more accurate the approximation. The PDF of a state vector $\mathbf{x}_k$ is approximated by $n_p$ particles as

$$p\left(\mathbf{x}\right) \approx \sum_{n=1}^{n_p} w^{(n)} \delta_{\mathbf{x}^{(n)}}\left(\mathbf{x}\right), \quad (5)$$

where $\mathbf{x}^{(n)}$ is the $n$-th sample, $w^{(n)}$ is its weight, and $\delta_{\mathbf{x}^{(n)}}$ stands for the Dirac measure centred at $\mathbf{x}^{(n)}$.

```
Initialize particles: $\mathbf{x}_0^{(i)}, i = 1 \ldots N_p$
for $k = 1$ to $N_k$ do
    Predict
        $\mathbf{x}_{k|k-1}^{(i)} \sim \varphi(\mathbf{x} \mid \mathbf{x}_{k-1}^{(i)})$ {Draw particles from transition density}
Update
        $w_k^{(i)} = g_k(\mathbf{z}_k \mid \mathbf{x}_{k|k-1}^{(i)})$ {Compute particle likelihoods}
        $\widehat{\mathbf{x}}_k = \sum w_k^{(i)} \mathbf{x}_{k|k-1}^{(i)}$ {Output point estimate as weighted sum}
        $w_k^{(i)} = w_k^{(i)} / \sum w_k^{(i)}$ {Compute normalized weights}
        Draw $\mathbf{x}_k^{(i)}$ from $\{\mathbf{x}_{k|k-1}^{(i)}\}_{i=1}^{Np}$ with probability $w_k^{(i)}$ {Resample particles}
end for
```

Algorithm 2: Bootstrap particle filter.

The particle filter can be considered a Bayesian-type filter where the PDFs are numerically handled. At the predict stage samples are drawn from a pool, called the importance density. The choice of the importance density is crucial. An obvious choice is to use the state transitional density whose method is referred to as the bootstrap particle filter. The update stage involves the computation of the likelihoods for every particle which is realized by the evaluation of the PDF associated with the measurement model. The normed likelihood values give the updated particle weights and the point estimate for the current timestep can be obtained by the weighted sum of the particle states. The main steps of the bootstrap particle filter are summarized in Algorithm 2.

*2.1. Multimodel Estimation.* The purpose of multimodel filtering is twofold. On one hand it helps giving a more precise state estimation if the correct system model is used and on the other hand it provides information on the actual mode of operation. To be effective at both at the same time, as will be seen, can be a contradictory requirement.

System model is the collective term for the process or motion model and the measurement or sensor model. The mode of operation or system mode refers to a certain kind of behavior that we identify, e.g., accelerating or turning. A given process model, if general enough, can account for multiple modes of operation. Mode change is the switching between two modes of operation. The mode history is the time sequence of the actual system modes. Mode uncertainty refers to the circumstance that we are not aware of the actual system mode.

One way of dealing with model uncertainties in an estimation problem is to use a number of plausible system model, compare their performances, and choose one result or combine several. The output of a filter associated with a certain mode is referred as the mode-conditioned estimate. There are numerous multiple model algorithms at hand [6]. The static multiple model approach has the supposition that the system does not change its behavior during the observation period and the filter selects the most likely mode and outputs a weighted estimate that is a combination of the individual filters. The dynamic multiple model estimator considers mode switching and uses mode transition probabilities that are predefined parameters. The exact solution to the dynamic multiple model problem, the Generalized Pseudo-Bayesian (GPB) estimator, has complexity that exponentially grows with time because it covers all the possible combinations of mode histories with a filter. Since the exact solution is intractable, one has to approximate and take into account only the last or the last two modes of operation resulting in the first (GPB1) and second order (GPB2) estimators. If $J$ mode of operation is considered, the GPB1 runs $J$ filters while the GPB2 runs $J^2$ filters, because of deeper memory of the algorithm. Therefore the computational requirement of the GPB1 and GPB2 filters scales are linearly and quadratically respectively in the number of modes.

*2.1.1. The Interacting Multiple Model estimator.* The interacting multiple model method, proposed in [7], is another type of approximate estimator to the dynamic multiple model problem. It uses only the output of the last step and creates a unique mix from the mode-conditioned estimates for every filter. Being a first order approximation it has a performance near the GPB2 but computationally only intense as the GPB1 [9].

The structure of the IMM estimator is depicted in Figure 1. The inputs to the recursive algorithm at timestep $k$ are the mode-conditioned state estimates and the mixing weights as a matrix $\nu^{(i,j)}$ for $i, j = 1 \ldots J$. The state estimates consist of a mean value and a covariance matrix, describing a Gaussian distribution. At the mixing stage the input for each filter is computed as a weighted sum of Gaussians:

$$\widetilde{\mathbf{x}}_{k-1}^{(j)} = \sum_{i=1}^{J} \nu_{k-1}^{(i,j)} \mathbf{x}_{k-1}^{(i)} \tag{6}$$

$$\widetilde{P}_{k-1}^{(j)} = \sum_{i=1}^{J} \nu_{k-1}^{(i,j)} \left[ P_{k-1}^{(i)} + \left( \mathbf{x}_{k-1}^{(i)} - \widetilde{\mathbf{x}}_{k-1}^{(j)} \right) \left( \mathbf{x}_{k-1}^{(i)} - \widetilde{\mathbf{x}}_{k-1}^{(j)} \right)^{\top} \right] \tag{7}$$

These values and the measurement vector $\mathbf{z}_k$ are passed to the filters. Beside the state estimation the filters also produce the residual and the associated covariance matrices. From these a model likelihood is computed as

$$\mathscr{L}_k^{(j)} \left( \mathbf{z}_k \right) = \mathscr{N} \left( \mathbf{r}_k; 0, S_k^{(j)} \right). \tag{8}$$

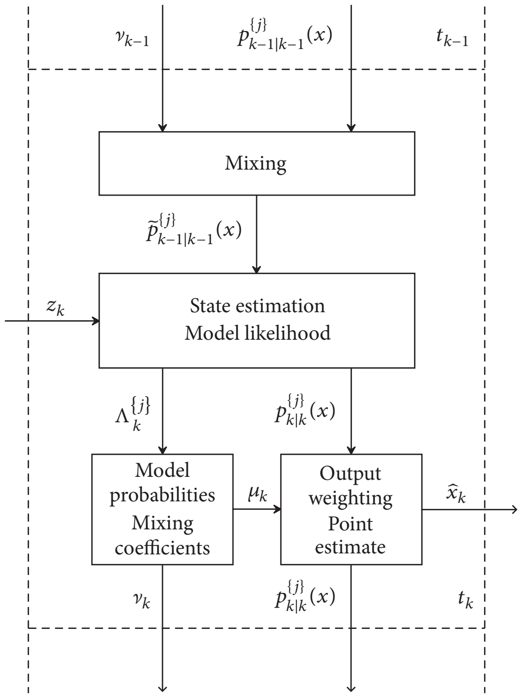


Figure 1: Structure of the interacting multiple model algorithm.

The mode probabilities are not purely the likelihoods because mode switching dynamics are implemented with the help of the mode transition probability matrix $\pi$ through which the prior values are derived:

$$\mu^{(j)}_{k|k-1} = \sum_{i=1}^{J} \pi_{ij} \mu^{(i)}_{k-1}. \tag{9}$$

The updated mode probabilities are

$$\mu^{(j)}_{k} = \frac{\mathscr{L}^{(j)}_k \mu^{(j)}_{k|k-1}}{\sum_{i=1}^{J} \mathscr{L}^{(i)}_k \mu^{(i)}_{k|k-1}}, \tag{10}$$

An overall estimate can be computed by weighting each filter output by the mode probabilities:

$$\hat{\mathbf{x}}_{k|k} = \sum_{j=1}^{J} \mu^{(j)}_k \mathbf{x}^{(j)}_{k|k}. \tag{11}$$

The associated covariance matrix is

$$P_{k|k} = \sum_{j=1}^{J} \mu^{j}_k \left[ P^{(j)}_{k|k}. + \left(\mathbf{x}^{(j)}_{k|k} - \hat{\mathbf{x}}_{k|k}\right)\left(\mathbf{x}^{(j)}_{k|k} - \hat{\mathbf{x}}_{k|k}\right)^{\top} \right]. \tag{12}$$

The mixing coefficients for the next step in the recursion is given by

$$\nu^{(i,j)}_{k-1} = \frac{\pi_{ij} \mu^{(i)}_{k-1}}{\sum_{m=1}^{J} \pi_{mj} \mu^{(m)}_{k-1}}. \tag{13}$$

*2.2. IMM Particle Filter.* Originally the IMM algorithm was working with Kalman filters [7]. The realization of the IMM with extended Kalman filters is trivial, because the output of an EKF is consistent with the KF. The same does not hold for the particle filter since its output consists of weighted samples and cannot be directly integrated into the IMM framework. One method is to distribute the particles between the mode-conditioned estimators. The more likely a mode estimate is, the more particles will belong to that estimator. We used another approach, which is closer to the original structure of the algorithm [20, 21].

Every filter has a unique Gaussian input parametrized by $\tilde{\mathbf{x}}^{(j)}_{k-1|k-1}, \tilde{P}^{(j)}_{k-1|k-1}$. The PF draws $N_p$ samples from this distribution

$$\mathbf{x}^{(j,n)}_{k-1|k-1} \sim \mathcal{N}\left(\tilde{\mathbf{x}}^{(j)}_{k-1|k-1}, \tilde{P}^{(j)}_{k-1|k-1}\right), \quad n = 1 \ldots N_p. \tag{14}$$

The weights associated with the samples have equal values: $1/N_p$. The particles are then propagated through the motion model:

$$\mathbf{x}^{(j,n)}_{k|k-1} \sim \mathcal{N}\left(F\mathbf{x}^{(j,n)}_{k-1|k-1}, \Gamma Q \Gamma^{\top}\right), \quad n = 1 \ldots N_p \tag{15}$$

The likelihood of a particle is

$$g\left(\mathbf{z}_k \mid \mathbf{x}^{(j,n)}_{k|k-1}\right) = \mathcal{N}\left(\mathbf{z}_k; h\left(\mathbf{x}^{(j,n)}_{k|k-1}\right), R\right). \tag{16}$$

A point estimate is computed from the particles by a weighted sum:

$$\hat{\mathbf{x}}^{(j)}_{k|k} = \sum_{n=1}^{N} w^{(j,n)}_{k|k} \mathbf{x}^{(j,n)}_{k|k} \tag{17}$$

with covariance

$$P^{(j)}_{k|k} = \sum_{j=1}^{N} \left(\mathbf{x}^{(j,n)}_{k|k} - \hat{\mathbf{x}}^{(j)}_{k|k}\right)\left(\mathbf{x}^{(j,n)}_{k|k} - \hat{\mathbf{x}}^{(j)}_{k|k}\right)^{\top}. \tag{18}$$

The particle filter does not produce residual covariance like the EKF. To be integrated in the IMM framework, a model likelihood has to be derived. The estimated measurement is

$$\hat{\mathbf{z}}^{(j)}_k = H\hat{\mathbf{x}}^{(j)}_{k|k}. \tag{19}$$

The model likelihood has an analogous form as in the EKF

$$\mathscr{L}^{(j)}_k\left(\mathbf{z}_k\right) = \mathcal{N}\left(\mathbf{z}_k; \hat{\mathbf{z}}_k, S^{(j)}_k\right), \tag{20}$$

Here, $S^{(j)}_k$ is

$$S^{(j)}_k = R + \sum_{n=1}^{N} \left(H\mathbf{x}^{(j,n)}_{k|k} - \hat{\mathbf{z}}^{(j)}_k\right)\left(H\mathbf{x}^{(j,n)}_{k|k} - \hat{\mathbf{z}}^{(j)}_k\right)^{\top}. \tag{21}$$

With $\mathscr{L}^{(j)}_k(\mathbf{z}_k)$ the mode probability is computed as in (10).

If particle degeneration is an issue, one will perform some form of resampling after the update step [22]. In our case, as we sample from the mixed distribution at every timestep, no particle degeneration is possible; hence, we implemented the PF without resampling. The individual particles are not preserved; only the weighted point estimate, its covariance, and the model likelihood are utilized.

## 3. Evaluation Framework

To evaluate the filter performances a simulated environment is used. The tracked object is assumed to be a road vehicle with appropriate motion characteristics. The maneuvering vehicle is observed from a stationary sensor at the origin. The measurements are generated according to standard automotive radar models.

*3.1. Models.* In this study the considered modes of operation are moving with constant velocity and turning with constant speed along a circle or a clothoid segment.

The CV and CT model both use the state vector $\mathbf{x} = [x, y, \vartheta, v, w]$ and the motion model has the form:

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k) + \mathbf{w}_k, \tag{22}$$

where $f_k$ either stands for the CV or the CT model. With the given the state vector the CV model is

$$f_k^{CV}(\mathbf{x}_k) = \begin{bmatrix} x_k + T_s v_k \cos(\vartheta_k) \\ y_k + T_s v_k \sin(\vartheta_k) \\ \vartheta_k \\ v_k \\ w_k \end{bmatrix} \tag{23}$$

which after linearization reads as

$$F_k^{CV} = \begin{bmatrix} 1 & 0 & -T_s v_k \sin(\vartheta) & T_s \cos(\vartheta) & 0 \\ 0 & 1 & T_s v_k \cos(\vartheta) & T_s \sin(\vartheta) & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

The constant turn rate (CT) model is angular velocity dependent:

$$f_k^{CT}(\mathbf{x}_k) = \begin{bmatrix} x_k + \dfrac{v_k}{w_k} \sin(\vartheta_k + w_k T_s) - \dfrac{v_k}{w_k} \sin(\vartheta_k) \\ y_k - \dfrac{v_k}{w_k} \cos(\vartheta_k + w_k T_s) + \dfrac{v_k}{w_k} \cos(\vartheta_k) \\ \vartheta_k + w_k T_s \\ v_k \\ w_k \end{bmatrix} \tag{25}$$

Linearizing (25) around $\mathbf{x}_k$ yields

$$F_k^{CT} = \begin{bmatrix} 1 & 0 & \dfrac{\partial f_x^{CT}}{\partial \vartheta_k} & \dfrac{\partial f_x^{CT}}{\partial v_k} & \dfrac{\partial f_x^{CT}}{\partial \omega_k} \\ 0 & 1 & \dfrac{\partial f_y^{CT}}{\partial \vartheta_k} & \dfrac{\partial f_y^{CT}}{\partial v_k} & \dfrac{\partial f_y^{CT}}{\partial \omega_k} \\ 0 & 0 & 1 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

where

$$\frac{\partial f_x^{CT}}{\partial \vartheta_k} = \frac{v_k}{w_k} \cos(\vartheta_k + w_k T_s) - \frac{v_k}{w_k} \cos(\vartheta_k) \tag{27}$$

$$\frac{\partial f_x^{CT}}{\partial v_k} = \frac{\sin(\vartheta_k + w_k * T_s) - \sin(\vartheta_k)}{w_k} \tag{28}$$

$$\begin{aligned} \frac{\partial f_x^{CT}}{\partial \omega_k} = \frac{v_k}{w_k^2} (&\cos(\vartheta_k + w_k T_s) + w_k T_s \sin(\vartheta_k + w_k T_s) \\ &- \cos(\vartheta_k)) \end{aligned} \tag{29}$$

$$\frac{\partial f_y^{CT}}{\partial \vartheta_k} = \frac{v_k}{w_k} \sin(\vartheta_k + w_k T_s) - \frac{v_k}{w_k} \sin(\vartheta_k) \tag{30}$$

$$\frac{\partial f_y^{CT}}{\partial v_k} = \frac{\cos(\vartheta_k) - \cos(\vartheta_k + w_k T_s)}{w} \tag{31}$$

$$\begin{aligned} \frac{\partial f_y^{CT}}{\partial \omega_k} = \frac{v_k}{w_k^2} (&\cos(\vartheta_k + w_k T_s) + w_k T_s \sin(\vartheta w_k T_s) \\ &- \cos(\vartheta_k)). \end{aligned} \tag{32}$$

It would also be possible to use a simple CV motion model with states $[x, \dot{x}, y, \dot{y}]$, system matrix

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \tag{33}$$

and perform the mixing in the IMM algorithm with state vectors of different length and components. In such case, depending on the actual components of the state vectors, one has to either augment one state vector with the missing element or transform the state vector to the base of the other one. These kinds of computations, detailed in [23, 24], are out of scope of the current study. For this reason the same state vector was used for both the CV and CT motion.

The noise acting on the system is modeled as an additive Gaussian term. For the CV and CT model it is a white noise linear acceleration and white noise linear and angular acceleration, respectively.

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k) + \Gamma w_k \tag{34}$$

where

$$\Gamma = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix}. \qquad (35)$$

and $w_k$ is a zero mean Gaussian with covariance

$$Q = \gamma \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}. \qquad (36)$$

In (36) the diagonal elements stand for the linear ($\sigma_1$) and angular ($\sigma_2$) acceleration. The factor $\gamma$ scales the noise intensity. The PDF associated with the motion model has the form

$$\varphi_k(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}; f_k(\mathbf{x}_k), \Gamma Q \Gamma^\top\right) \qquad (37)$$

The measurement vector comes from simulated radar observations with components $\mathbf{z} = [z_1, z_2]^\top$, where $z_1$ is the bearing angle and $z_2$ is the distance. The measurements originate from the nonlinear sensor model $h(\mathbf{x})$:

$$h(\mathbf{x}) = \begin{bmatrix} \operatorname{atan2}(y_k, x_k) \\ \sqrt{x_k^2 + y_k^2} \end{bmatrix}. \qquad (38)$$

In linearized form the sensor model reads as

$$H_k = \begin{bmatrix} \frac{-y_k}{d^2} & \frac{x_k}{d^2} & 0 & 0 & 0 \\ \frac{x_k}{d^2} & \frac{y_k}{d^2} & 0 & 0 & 0 \end{bmatrix} \qquad (39)$$

where $d = \sqrt{(x_k^2 + y_k^2)}$.

With the help of (38) the likelihood function is given by

$$g(\mathbf{z}_k \mid \mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; h(\mathbf{x}_k), R), \qquad (40)$$

where the covariance matrix is

$$R = \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix}. \qquad (41)$$

*3.2. System Setup.* As it was pointed out in [25], narrow beam range finders like long range radar and especially lidar have a possibility of failing to detect thin objects, for example, lampposts. An additional problem is that even if the sensor correctly detects an obstacle, a filtering algorithm might not recognize its existence. Sensors that are practically noise free lead to a measurement likelihood function that is spike-like. The support of a likelihood function of this kind, due to limited numerical precision, can be a very small region in the state space. If the estimator algorithm does not provide a prior in that small region, then the likelihood function will be zero out of other priors. At the filtering level the narrow likelihood problem can be handled by applying a Gaussian smoother. Low level sensor fusion techniques and track-before-detect approaches can help to overcome typical sensor drawbacks [26]. At the detector level, new, emerging technologies are expected to overcome this deficiency [27–29].

Table 1: Motion of the tracked object in the simulation: Trajectory1.

| | Vehicle motion | Turn rate | Duration [s] |
|---|---|---|---|
| #1 | CV | — | 15 |
| #2 | CT | −13 deg/s | 15 |
| #3 | CV | — | 20 |
| #4 | Clothoid | $\omega_1$ | 40 |
| #5 | CV | — | 10 |

Table 2: Motion of the tracked object in the simulation: Trajectory2.

| | Vehicle motion | Turn rate | Duration [s] |
|---|---|---|---|
| #1 | CV | — | 25 |
| #2 | CT | −13 deg/s | 15 |
| #3 | CV | — | 30 |
| #4 | Clothoid | $\omega_2$ | 20 |
| #5 | CV | — | 10 |

The vehicle is initialized with state vector $x_0 = [0, 0, 30, \pi/4, 0]$. The noise acting on the system is described by $\sigma_1^2 = 1\,\mathrm{m}^2\mathrm{sec}^{-4}$ $\sigma_2^2 = 0.01\,\mathrm{rad}^2\mathrm{sec}^{-4}$ $\sigma_b^2 = 0.1\,\mathrm{deg}^2$ $\sigma_d^2 = 20\,\mathrm{m}^2$. Sensor noise values are chosen according to common automotive radar devices [30].

*3.3. Scenario.* The vehicle moves along a road segment and its trajectory is segmented into straight lines and circular or clothoid arcs. Two trajectories were designed with different characteristics to help emphasize the nature of the filters. Trajectory1 (T1), as outlined in Table 1, has more turns and Trajectory2 (T2) has more straight parts, as indicated by Table 2. In the clothoid segment in both cases the angular velocity ($\omega_1$ and $\omega_2$) changes linearly with time. For T1 $\omega_1$ starts from zero and increases to 16 deg /s in 20 steps and then decreases to −12 deg /s in 20 steps. The same is true for $\omega_2$ for T2 except that 10 – 10 steps are used in this case.

The speed is constant throughout the whole simulation. The duration of the simulation is $T = 100$ sec with time step $T_s = 1$ sec. The steps for $\omega_1$ and $\omega_2$ are also $T_s$.

The filter performances were examined along two parameters, the mode transition probability matrix $\pi$ and the process noise intensity $\gamma$. The performance is also measured along two dimensions: the position error and the estimated mode.

The mode transition matrix is adopted as

$$\begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} = \begin{bmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{bmatrix}, \qquad (42)$$

where $\alpha$ is the scaling parameter. The values taken by $\alpha$ and the process noise intensity $\gamma$ are shown in Figure 2.

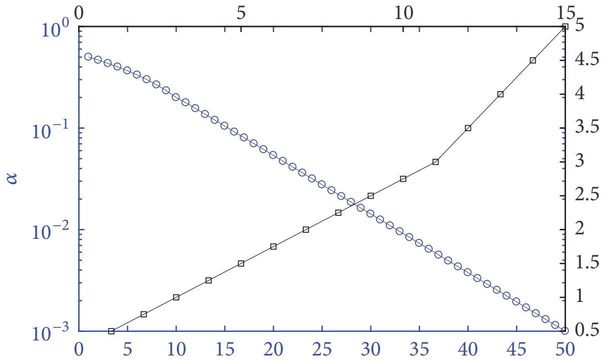


Figure 2: Parameter set of the simulation. 50 $\alpha$ values marked with circles and 15 $\gamma$ values marked with squares.

## 4. Results

The performance of the filters was measured from two aspects of interest. First, the position error is calculated as the Euclidean distance between the real and the estimated coordinates. Second, as the measure of maneuver detection, the estimated mode is regarded and two measures are defined: from which the first is the ratio of time steps when the correct mode was estimated versus the total time; and the second method is area based and indicates the robustness of the output. The area under the estimated mode probability curve is divided by the length of time that mode was active.

A ratio to measure the effect of relative motion to measurement uncertainty on the position is defined in [17] as the maneuvering index. In our case it can be quantified as

$$\lambda = \sqrt{\frac{\left\|\Gamma \mathbf{Q} \Gamma^{\top}\right\|_2}{\|\mathbf{R}\|_2}} \tag{43}$$

which takes values between 0.2 and 0.5. According to [17] an IMM is preferred to a single model KF if $\lambda$ is higher than 0.5. This limit is challenged in [18] with the conclusion that the IMM can outperform the KF if the model with the lower process noise is active for enough time.

Numerical complexity of the compared algorithms has the following order. The single model EKF has the lowest requirements. The state space is of low dimension, the involved matrix operations can be computed directly, without sophisticated numerical methods. The IMM-EKF needs to run a filter for every considered mode of operation. In addition, the IMM algorithm poses overhead mainly in terms of creating Gaussian mixtures and computing covariance matrices. The IMM-PF is the most computationally expensive. The most demanding steps are sampling, evaluation of the likelihood function, and sample covariance computation. The particle number was held constant 1000 which can be considered as a mediocre quantity. The current implementation used parallelization wherever it was possible at the particle level.

Simulations were conducted for both trajectories. The IMM is implemented with EKF and PF and for comparison single model CV/CT EKF were used. 100 Monte Carlo runs were averaged out for every parameter setting meaning 75000 runs in total. Figures 3, 7, 11, and 13 show IMM filter performance across the whole parameter range. Figures 4, 8, 12, and 14 sum up the efficiency of mode estimation and RMSE for parameter $\gamma = 2.5$ and for every $\alpha$.

Figures 5 and 9 show the results of the 100 MC averaged runs for parameters $\gamma = 2.5$ and $\alpha = 0.3$. The estimation of one particular run from 100 is presented in Figures 6 and 10.

The estimated mode at each timestep is acquired by selecting the most probable mode:

$$J_k = \arg\max_j \left(\mu_k^{(j)}\right), \tag{44}$$

A general property of the IMM filters in all cases is that the position RMSE is lower for greater $\alpha$ values while the dependence on $\gamma$ shows variety. The effect of a smaller $\alpha$ value is that the diagonal elements $(1-\alpha)$ in $\pi$ become large and mode transitions are harder to achieve; resulting in a slower multimodel filter and a lower ratio in terms of mode-correctness. In contrary with a higher $\alpha$ value mode transitions can be realized with low latency after measurements start to deviate from the expected state. The downside of using values near 0.5 is that the robustness of the mode estimation in terms of the area based measure degrades as the correct mode gets lower confidence. This effect is demonstrated in Figures 4, 8, 12, and 14.

The position RMSE of IMM-EKF exhibits negligible dependence on $\gamma$ for both T1 and T2. The accuracy of mode estimation on Figure 7 shows a connection of linear nature between the hyperparameters for T2. In case of T1 the best model estimation efficiency can be achieved with lower noise and higher $\alpha$ parameters (Figure 3).

The single model EKF with constant turn rate can be considered as an option in certain situations. If the tracked vehicle performs a great amount of maneuvering, the EKF can outperform the IMM in terms of position RMSE at the cost of providing no mode estimates. The EKF, in scenario T1, estimates the state significantly better than the IMM; on the contrary in T2 the IMM performs comparably while the EKF is worse. The particle filter implementations of the IMM have different characteristics. The position RMSE shows dependence on both hyperparameters. The mode of operation can be estimated with greater efficiency for certain parameters compared to the IMM-EKF. In scenario T1 the dependence of the RMSE on $\alpha$ is weak for lower $\gamma$ values indicating a greater freedom at the filter design process. The mode can be estimated slightly more efficiently with the IMM-PF than the IMM-EKF in both scenario T1 and T2.

## 5. Conclusion

The continuously increasing level of automation in road traffic and vehicles poses new requirements towards data processing algorithms [31]. Road vehicles with autonomous functions need the ability to sense the environment and faithfully capture and interpret the traffic situation. Algorithms for advanced driver assistance systems have to run realtime which greatly constrains the level of analysis of acquired data. This study covers performance evaluation of multimodel object tracking and maneuver identification methods of great

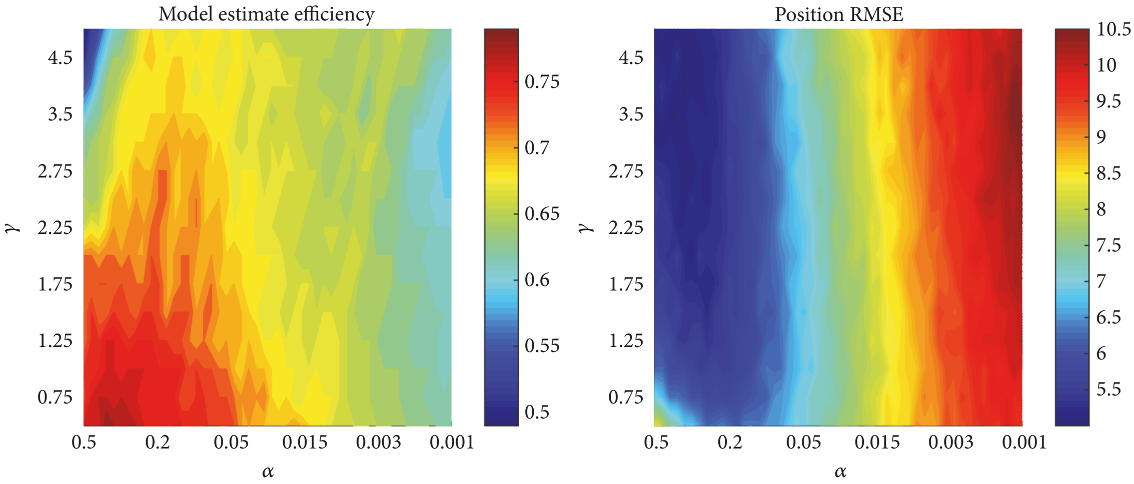


Figure 3: Contour plot of MC averaged position RMSE with respect to filter hyperparameters. Case: EKF – T1.

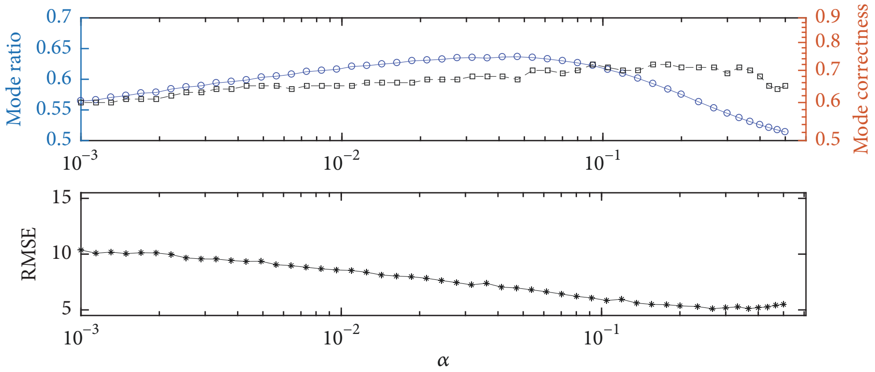


Figure 4: Top: MC averaged correctness of estimated mode of operation versus parameter $\alpha$. Squares indicate the time interval based ratio; circles stand for the area based computation. Case: EKF – T1.

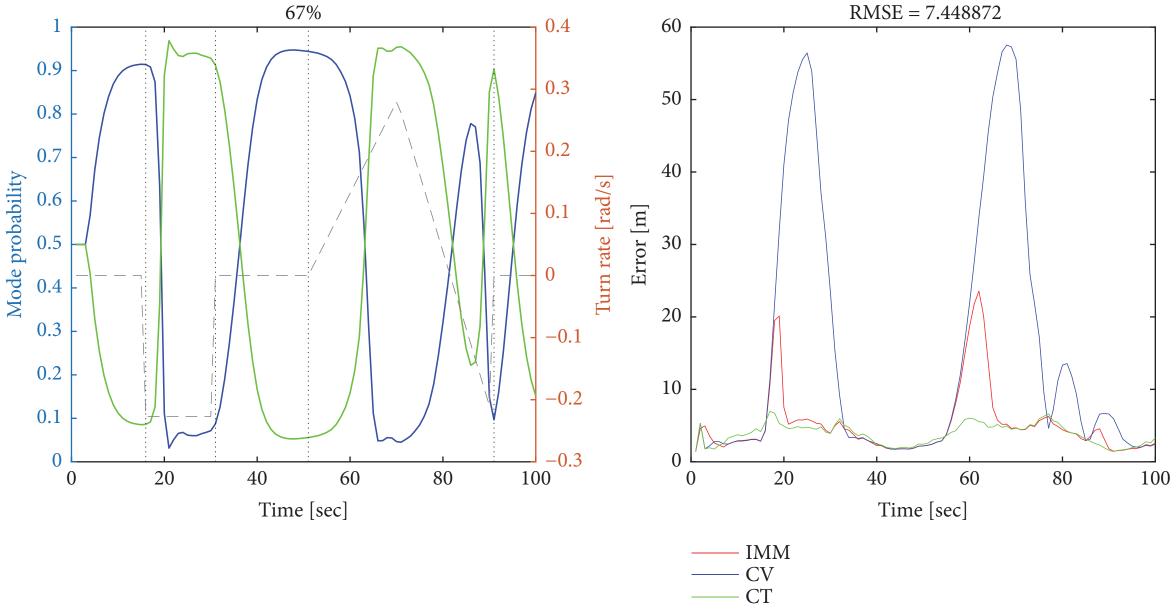


Figure 5: MC averaged result of the estimated mode probabilities (left) and the position error (right). The vertical dotted lines indicate mode changings. The dashed line is the actual angular velocity of the tracked vehicle. Case: EKF – T1.

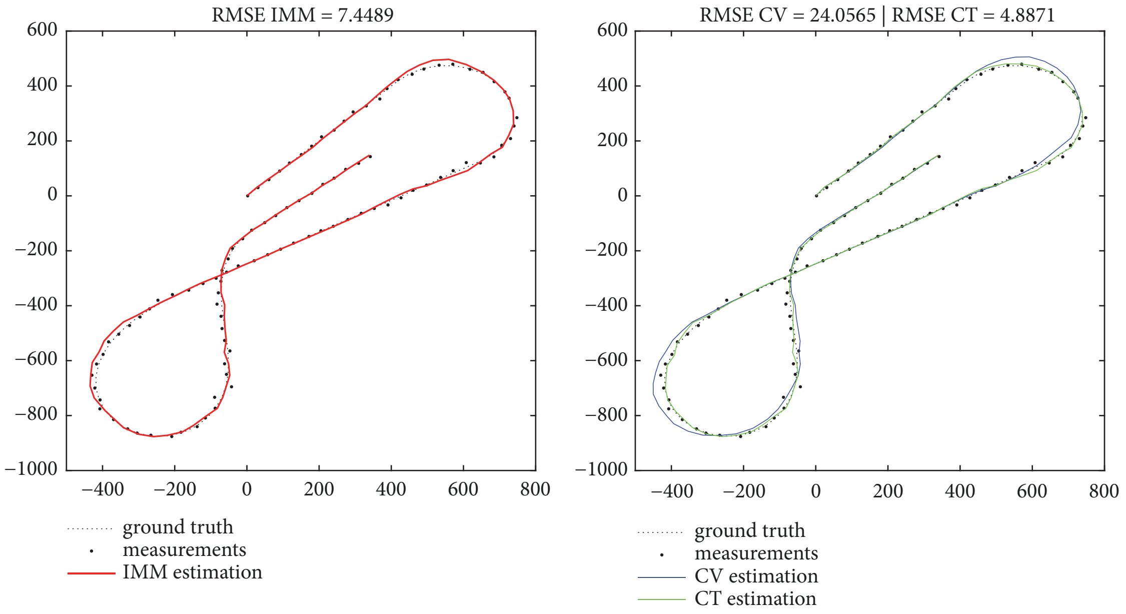


Figure 6: Result of a single MC run for the IMM-EKF (left) and CV/CT EKF. Case: EKF – T1.

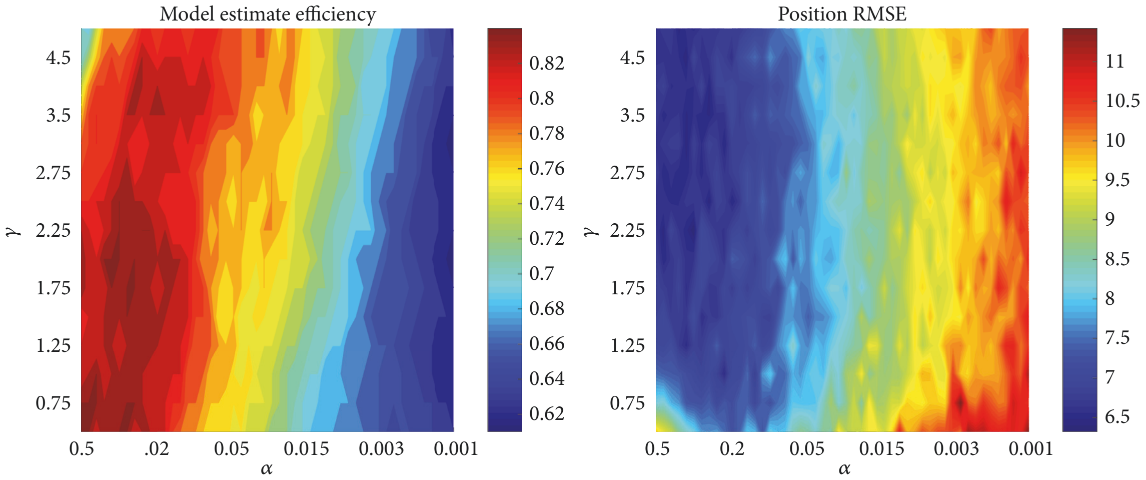


Figure 7: Contour plot of MC averaged position RMSE with respect to filter hyperparameters. Case: EKF – T2.

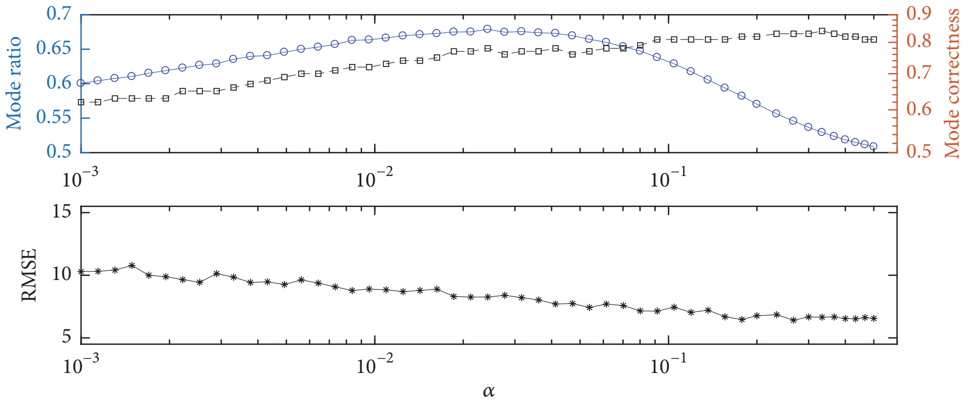


Figure 8: Top: MC averaged correctness of estimated mode of operation versus parameter $\alpha$. Squares indicate the time interval based ratio; circles stand for the area based computation. Case: EKF – T2.

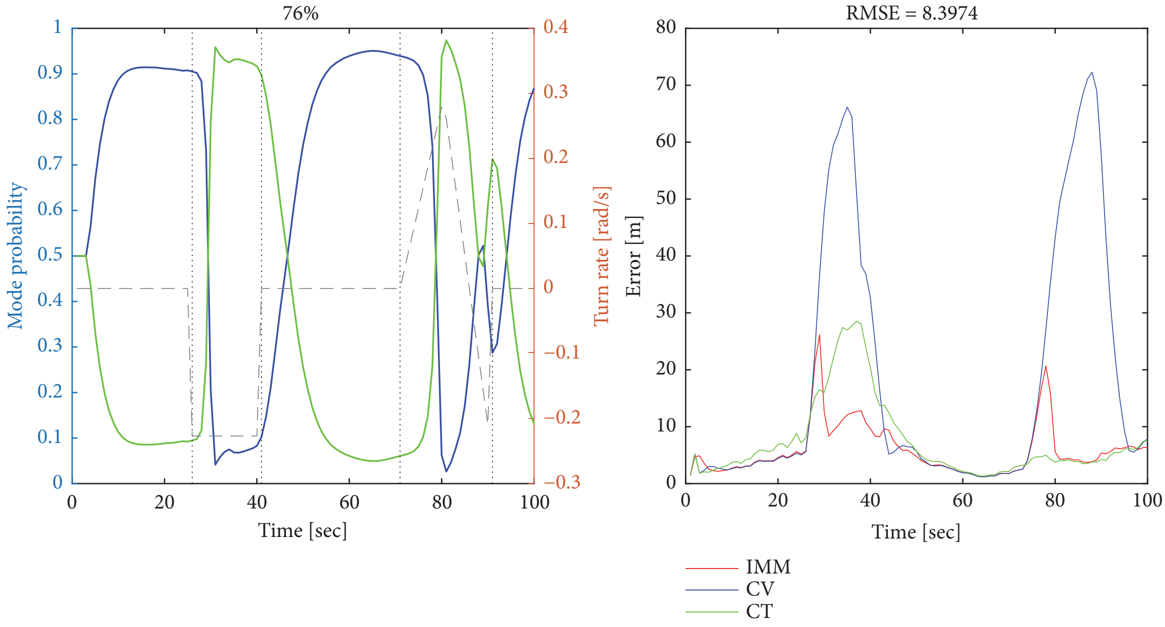


Figure 9: MC averaged result of the estimated mode probabilities (left) and the position error (right). The vertical dotted lines indicate mode changings. The dashed line is the actual angular velocity of the tracked vehicle. Case: EKF – T2.

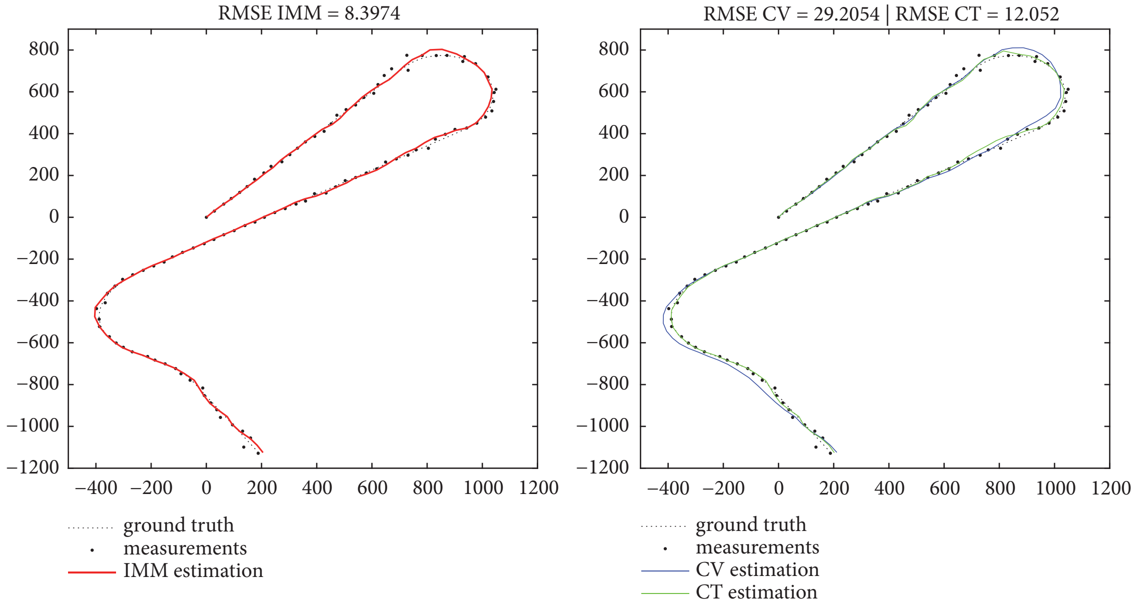


Figure 10: Result of a single MC run for the IMM-EKF (left) and CV/CT EKF. Case: EKF – T2.

importance for the automotive industry. The scenario was provided by a simulated environment where the sensor model had the characteristics of common automotive radar which performed measurements on the bearing angle and distance of the tracked object. The implemented estimators are the IMM-EKF, IMM-PF, and the single model EKF. The filters were evaluated along two aspects, the RMSE of the estimated position and the performance of the mode estimates. The varying parameters of the simulation were the mode transition probability matrix and the intensity of the process noise. The robustness of estimates of the actual mode of operation was measured by two methods. The single and multimodel filters were compared in terms of position RMSE. The IMM-KF and IMM-PF have different scaling properties regarding the simulation parameters which imply different design approaches.

It is essential to fine tune or, during operation, reconfigure the filters to help adaption to the actual conditions. The presented work can serve as a guideline creating a customized framework to analyze and help the development of

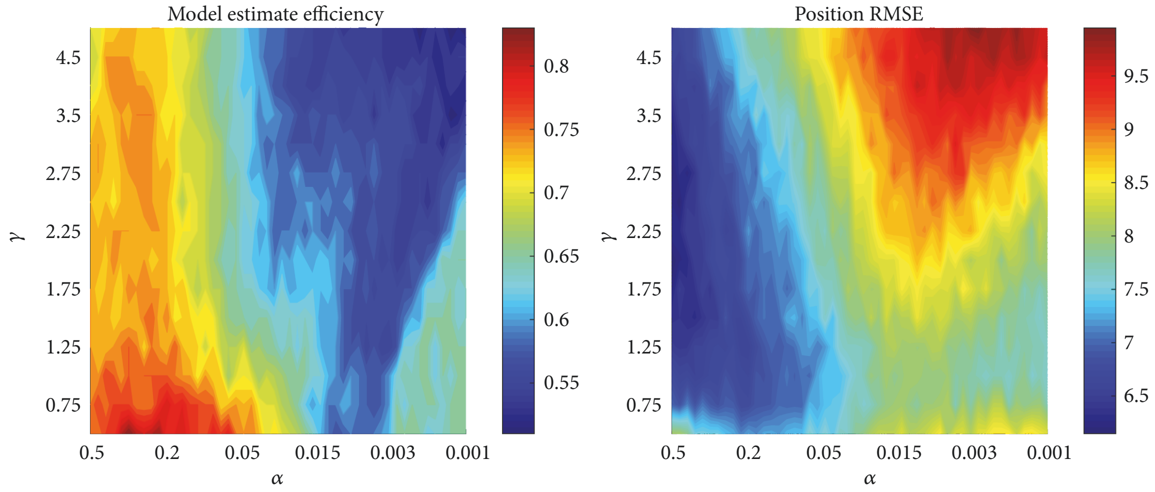


Figure 11: Contour plot of MC averaged position RMSE with respect to filter hyperparameters. Case: PF - T1.

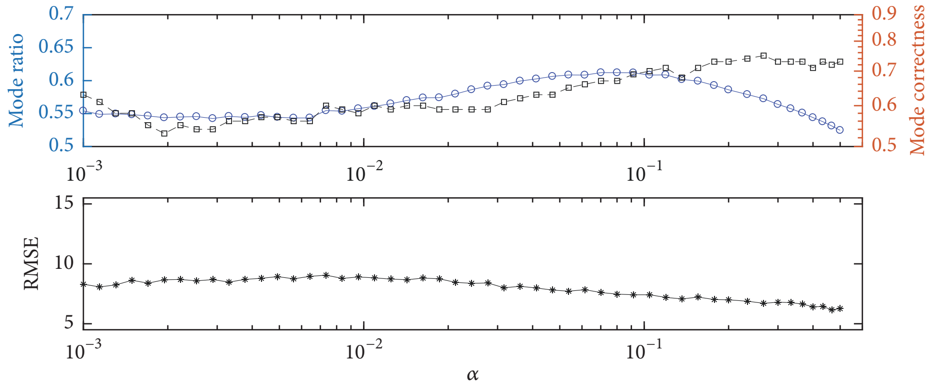


Figure 12: Top: MC averaged correctness of estimated mode of operation versus parameter $\alpha$. Squares indicate the time interval based ratio; circles stand for the area based computation. Case: PF – T1.

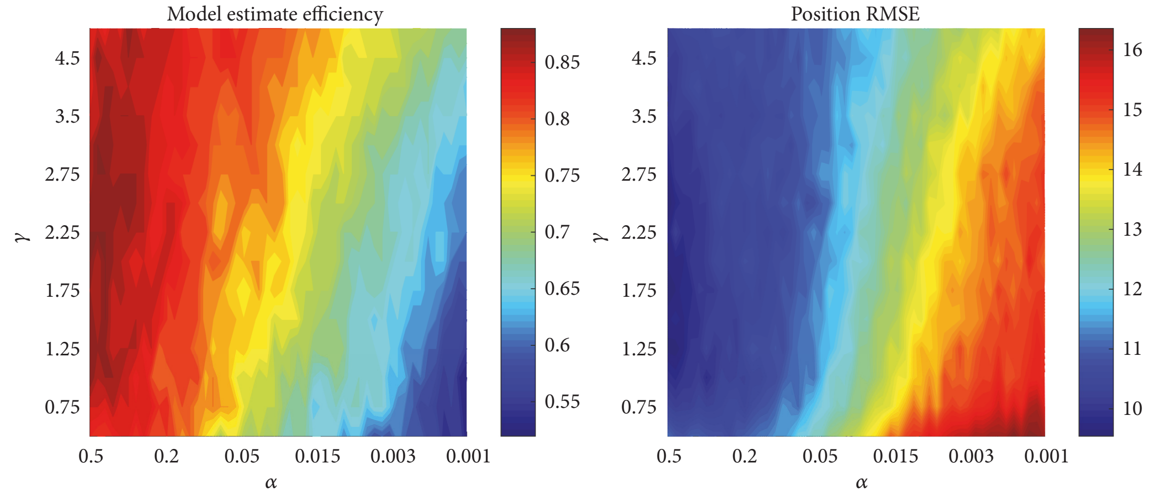


Figure 13: Contour plot of MC averaged position RMSE with respect to filter hyperparameters. Case: PF – T2.

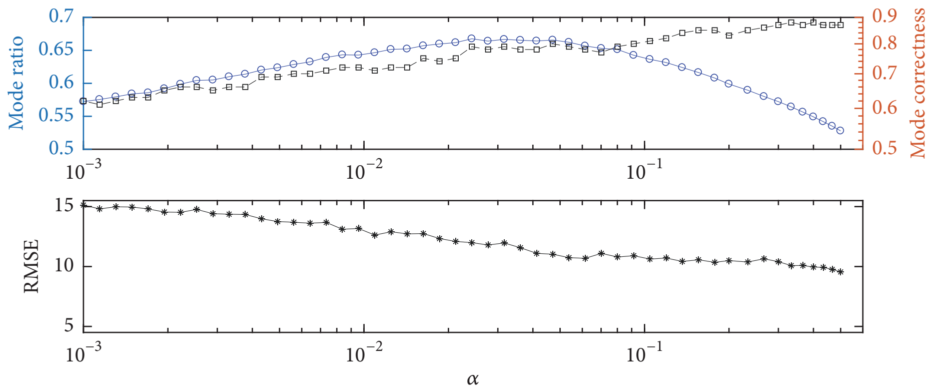


FIGURE 14: Top: MC averaged correctness of estimated mode of operation versus parameter $\alpha$. Squares indicate the time interval based ratio; circles stand for the area based computation. Case: PF – T2.

multiple-model state and maneuver estimating algorithms for advanced vehicular functions. In the knowledge of the scaling properties and behavior of the algorithms parameter ranges for the optimal performance can be identified.

## List of Notations

| | |
|---|---|
| $n_x$: | Dimension of state space |
| $n_z$: | Dimension of measurement space |
| $p(\mathbf{x}_k)$: | PDF of state vector $\mathbf{x}_k$ |
| $\mathbf{x}_k$: | State vector at time $t_k$ |
| $\mathbf{w}_k \in \mathbb{R}^{n_x}$: | Process noise vector |
| $\mathbf{z}_k$: | Measurement vector at time $t_k$ |
| $\mathbf{v}_k \in \mathbb{R}^{n_z}$: | Measurement noise vector |
| $\varphi_{k\|k-1}(\mathbf{x}_k \mid \mathbf{x}_{k-1})$: | State transition density |
| $g_k(\mathbf{z}_k \mid \mathbf{x}_k)$: | Likelihood function of $\mathbf{z}_k$ |
| $F$: | State transition matrix |
| $H$: | Measurement matrix |
| $f(\cdot)$: | Nonlinear state transition function |
| $h(\cdot)$: | Nonlinear measurement function |
| $\mathbf{x}_k^{(j)}$: | Estimated state conditioned on mode $j$ |
| $\mathbf{P}_k^{(j)}$: | Estimated covariance conditioned on mode $j$ |
| $\mathscr{L}_k^{(j)}$: | Likelihood of mode $j$ |
| $S_k^{(j)}$: | Residual covariance of mode $j$ |
| $\pi$: | Mode transition probability matrix |
| $\gamma$: | Intensity of process noise |
| $Q$: | Process noise covariance |
| $R$: | Measurement noise covariance |
| $\Gamma$: | Process noise input matrix |
| $\mu_k^{(j)}$: | Probability of mode $j$ |
| $\nu_k^{(i,j)}$: | Mixing matrix. |

## Data Availability

No data were used to support this study. The study used MATLAB simulation to evaluate the methods; the source can be requested from the corresponding author.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The research reported in this paper was supported by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKPMI/FM). The project has also been supported by the European Union, cofinanced by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001: talent management in autonomous vehicle control technologies).

## References

[1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[2] T. Tettamanti, I. Varga, and Z. Szalay, “Impacts of autonomous cars from a traffic engineering perspective,” *Periodica Polytechnica Transportation Engineering*, vol. 44, no. 4, pp. 244–250, 2016.

[3] N. Deo and M. M. Trivedi, “Multi-modal trajectory prediction of surrounding vehicles with maneuver based lSTMs,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Changshu, China, 2018.

[4] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *Robomech Journal*, vol. 1, no. 1, 2014.

[5] O. Törő, T. Bécsi, S. Aradi, and Á. Vellai, “Multimodel state estimation in road traffic using constrained filtering,” in *Proceedings of the IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, IEEE, 2018.

[6] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. Part V: multiple-model methods,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

[7] H. A. P. Blom and Y. Bar-Shalom, “Interacting multiple model algorithm for systems with Markovian switching coefficients.,” *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[8] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, “Interacting multiple model methods in target tracking: a survey,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[9] Y. Bar-Shalom, L. X. Rong, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley & Sons, 2004.

[10] Z. Messaoudi, A. Ouldali, and M. Oussalah, “Comparison of interactive multiple model particle filter and interactive multiple model unscented particle filter for tracking multiple manoeuvring targets in sensors array,” in *Proceedings of the IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS)*, pp. 1–6, IEEE, 2010.

[11] M. Zhang and W. Chen, “Variable structure multiple model particle filter for maneuvering radar target tracking,” in *Proceedings of the International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, pp. 1754–1757, IEEE, 2010.

[12] H. F. Pek and W. N. Gee, “Combining imm method with particle filters for 3d maneuvering target tracking,” in *Proceedings of the International Conference on Information Fusion*, pp. 1–8, IEEE, 2007.

[13] R. Guo, Z. Qin, X. Li, and J. Chen, “An immupf method for ground target tracking,” in *Proceedings of the International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 96–101, IEEE, 2007.

[14] O. Törö, T. Bécsi, S. Aradi, and P. Gáspár, “IMM bernoulli filter for cooperative object tracking in road traffic,” *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 355–360, 2018.

[15] M. Roth, G. Hendeby, and F. Gustafsson, “Ekf/ukf maneuvering target tracking using coordinated turn models with polar/cartesian velocity,” in *Proceedings of the 17th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2014.

[16] F. Gustafsson and A. J. Isaksson, “Best choice of coordinate system for tracking coordinated turns,” in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 3, pp. 3145–3150, IEEE, Kobe, Japan, 1996.

[17] T. Kirubarajan and Y. Bar-Shalom, “Kalman filter versus imm estimator: when do we need the latter?” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1452–1457, 2003.

[18] M. Silbert, S. Sarkani, and T. Mazzuchi, “Comparing the state estimates of a kalman filter to a perfect imm against a maneuvering target,” in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pp. 1–5, IEEE, 2011.

[19] Z. Chen, “Bayesian filtering: from kalman filters to particle filters, and beyond,” *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[20] Z. Liu and J. Wang, “Interacting multiple model gaussian particle filter,” in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pp. 270–273, IEEE, 2011.

[21] S.-C. Du, Z.-G. Shi, W. Zang, and K.-S. Chen, “Using interacting multiple model particle filter to track airborne targets hidden in blind Doppler,” *Journal of Zhejiang University SCIENCE A*, vol. 8, no. 8, pp. 1277–1282, 2007.

[22] D.-C. Chang and M.-W. Fan, “Interacting multiple model particle filtering using new particle resampling algorithm,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 3215–3219, IEEE, 2014.

[23] K. Granström, P. Willett, and Y. Bar-Shalom, “Systematic approach to IMM mixing for unequal dimension states,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pp. 2975–2986, 2015.

[24] K. Granström, P. Willett, and Y. Bar-Shalom, “IMM without a match,” in *Proceedings of the 6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 109–112, IEEE, 2015.

[25] Y. Wiseman, “Ancillary ultrasonic rangefinder for autonomous vehicles,” *International Journal of Security and Its Applications*, vol. 10, no. 5, pp. 49–58, 2018.

[26] O. Törő, T. Bécsi, S. Aradi, and P. Gáspár, “Cooperative object detection in road traffic,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 264–269, 2017.

[27] Y. Takashima, B. Hellman, J. Rodriguez et al., “MEMS-based Imaging LIDAR,” in *Light, Energy and the Environment 2018 (E2, FTS, HISE, SOLAR, SSL)*, p. ET4A.1, Optical Society of America, 2018.

[28] H. W. Yoo, N. Druml, D. Brunner et al., “MEMS-based lidar for autonomous driving,” *e & i Elektrotechnik und Informationstechnik*, vol. 135, no. 6, pp. 408–415, 2018.

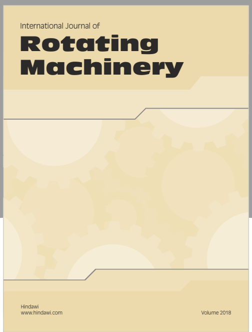
[29] J. Liu, Q. Sun, Z. Fan, and Y. Jia, “TOF Lidar development in autonomous vehicle,” in *Proceedings of the IEEE 3rd Optoelectronics Global Conference (OGC)*, pp. 185–190, IEEE, Shenzhen, China, 2018.

[30] R. H. Rasshofer, “Functional requirements of future automotive radar systems,” in *Proceedings of the 4th European Radar Conference (EURAD)*, pp. 259–262, IEEE, 2007.

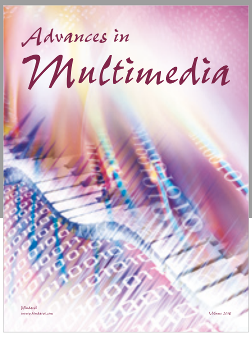
[31] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, “Multisensor data fusion: a review of the state-of-the-art,” *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

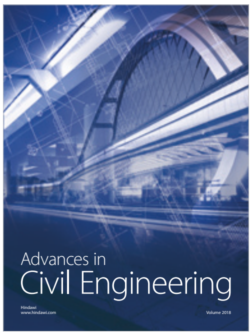
Journal of
Engineering

The Scientific World Journal

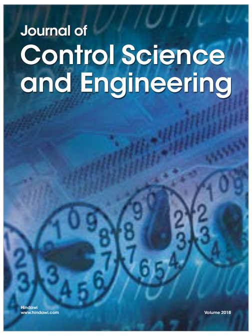

International Journal of
Rotating Machinery

Journal of
Sensors

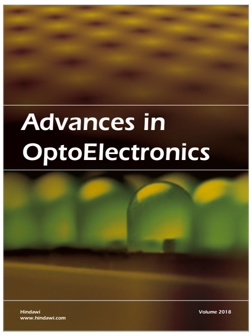

Advances in
Multimedia


Advances in
Civil Engineering

Journal of
Robotics

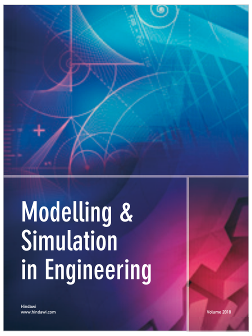

Journal of
Control Science and Engineering

Journal of
Electrical and Computer Engineering

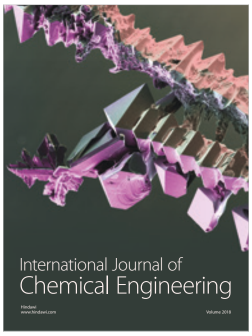

Advances in
OptoElectronics

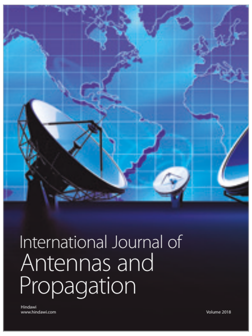
VLSI Design

International Journal of
Navigation and Observation

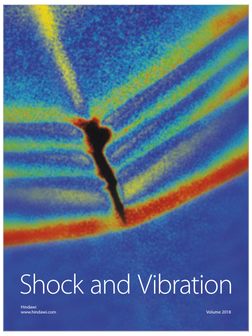

Modelling & Simulation in Engineering

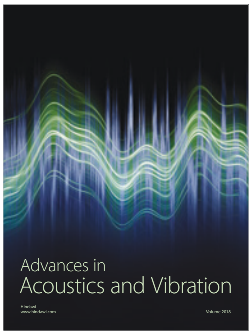
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration