Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 321967, 13 pages
doi:10.1155/2008/321967

Research Article

Variable-Mass Particle Filter for Road-Constrained Vehicle Tracking

Giorgos Kravaritis and Bernard Mulgrew

Institute for Digital Communications, The University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh EH9 3JL, UK

Correspondence should be addressed to Giorgos Kravaritis, g.kravaritis@ed.ac.uk

Received 20 July 2006; Revised 21 March 2007; Accepted 13 August 2007

Recommended by T.-H. Li

The paper studies the road-constrained vehicle tracking problem employing the multiple-model particle filtering framework. It introduces an approach which enables for a more efficient particle use within the multimodel structure of the tracker; rather than allocating the particles to the various modes of operation using fixed mode probabilities, it proposes to allocate the particles freely according to user-defined application-specific criteria. For compensating for the arbitrary allocation of the particles, the particles are assigned with masses which scale appropriately their weights. Simulation results demonstrate the improved particle efficiency of the new variable-mass approach when contrasted with the standard variable-structure multiple model particle filter in a vehicle tracking application.

1. INTRODUCTION

Vehicle tracking has drawn recently considerable attention from the scientific community, which studied it extensively in a wide range of applications including highway tracking, traffic control, navigation, accident avoidance, and joint classification and tracking [1–5]. This increasing interest was not only due to the growing importance of the problem itself but also due to its difficulty and complexity which made it ideal for comparing and benchmarking different tracking techniques. The problem is demanding since one often encounters physical constraints and obstructions, terrain-coupled vehicle motion, intense clutter returns, high false alarm rates, and closely separated slow targets that can execute abrupt turns and even stop.

Throughout the literature many different sensors have been used for the specific application, such as electro-optical and video [5, 6], infrared [7], GPS [8], high-range resolution radar [9], space-time adaptive processing radar [10], and ground moving target indicator (GMTI) radar [11–13]. In this work we use two-dimensional measurements from a static radar which measures the azimuth angle and the range of a vehicle which can move freely on and off the road. For tracking we use particle filters (PFs) which employ multiple modes of operation accounting for the different tracking subspaces and their associated dynamics. Road map information, in the form of motion constraints, is exploited for improving the estimation accuracy.

The PFs, introduced in their current form in [14] in 1993 (see report [15] for an insightful genealogical analysis of the sequential simulation-based Bayesian filtering), are powerful numerical methods which address the nonlinear/non-Gaussian Bayesian estimation problem. Based on the concepts of Monte Carlo integration and importance sampling, they employ a set of weighted samples or *particles* of the state density, which they propagate appropriately over time to calculate discrete approximations of the posterior state distribution. Textbooks [16, 17], report [15], and papers [18–20] offer a comprehensive analysis and literature review on sequential Monte Carlo methods and particle filtering.

In our application since the vehicle switches between different motion dynamics (can travel on or off a road, along a bridge, cross a junction, etc.), we use a *multiple-model* filter. The estimates in this class of filters are obtained using a mechanism that combines the outputs of the possible operating modes. Our work is based on the variable-structure multiple model particle filter (VSMMPF) [12, 17] vehicle tracker. The VSMMPF incorporated to particle filtering the *variable-structure* approach of the variable-structure interacting multiple model (VSIMM) algorithm [21, 22]. The VSIMM aimed to address a weakness of the interacting multiple model (IMM) filter [23, 24] which in certain applications exhibited a degraded performance due to the excessive "competition" among its models [25]. The VSIMM therefore proposed to use a *varying* number of active models according to the vehicle positioning on the road map approach which, indeed, enhanced the tracking accuracy. Moreover, due to the eclectic use of its active modes, it reduced the overall computational requirements. The VSMMPF demonstrated an even greater performance compared to the VSIMM since its particle filtering structure enabled it to cope better and more efficiently with the intense nonlinearity and non-Gaussianity of vehicle tracking.

The work described in this article attempts to improve the particle efficiency of the VSMMPF. Its key contribution is the use of particles with variable masses. Whereas in the VSMMPF the number of the particles allocated to its modes is proportional to *fixed* mode probabilities, in the proposed variable mass particle filter (VMPF) that number is allowed to vary according to arbitrary user-defined criteria. For compensating for the arbitrary over- or under-population of the particles to its modes, in the VMPF the particles are rescaled with appropriate scaling factors which we call masses.

The introduced vehicle tracker, adopting the variable-mass approach, is allowed to exploit information from the measurement and the difficulty of the mode dynamics to allocate its particles to the modes. The benefits thus are twofold: firstly more particles are allocated to the most probable and/or difficult modes for improving the tracking accuracy and secondly modes which are less probable and/or have easier dynamics obtain fewer particles for reducing the computational requirements. Other—more application specific—features of the proposed vehicle tracker is an on-road propagation mechanism which uses just one particle and a Kalman filter (KF) for reducing further the computational demands and a technique which enables the algorithm to deal with random road departure angles (instead of just $\pm 90^\circ$ in VSMMPF).

The structure of the paper is as follows. Section 2 establishes briefly basic principles of terrain-aided vehicle tracking and Section 3 introduces the variable-mass technique. Section 4 describes the new VMPF vehicle tracker, and Section 5 presents a simulation study which contrast the new algorithm with the VSMMPF. Finally, Section 6 summarises and presents the conclusions of this work.

2. VEHICLE TRACKING WITH ROAD MAPS

This section presents some basic concepts of vehicle tracking. A comprehensive introduction to tracking can be found in the standard textbook [26]. The notation that we use throughout the paper is bold uppercase roman letters for matrices ($\mathbf{A}$), bold lowercase roman letters for vectors ($\mathbf{a}$), uppercase roman letters for points in the space (A), and italic letters for functions and variables (A, a). The transpose of the matrix $\mathbf{A}$ is denoted as $\mathbf{A}^T$ and its inverse as $\mathbf{A}^{-1}$. In the studied scenario, a static radar monitors a ground scene

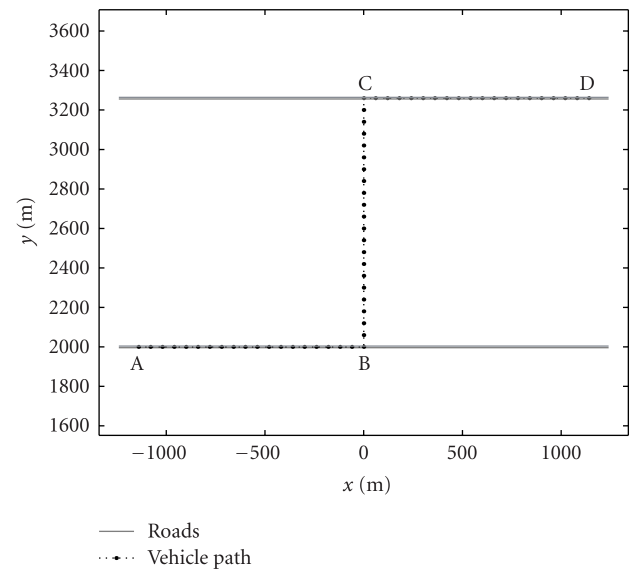

Figure 1: The road map of the simulation scenario. Although the figure presents a constant velocity ABCD path and a 90° road-departure angle, for the comparison in Section 5, the onroad velocity is perturbed with random accelerations and the departure angle varies randomly between 20–160°.

(Figure 1) in which a vehicle moves on and off the road. The vehicle moves with a nominal constant velocity, perturbed by a random Gaussian noise, and its dynamics evolve in the tracking state space according to the following equation:

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

The state vector $\mathbf{x}_k = [x_k\ y_k\ \dot{x}_k\ \dot{y}_k]^T$ consists of the vehicle's position and velocity and the noise vector $\mathbf{u}_k = [u_k^x\ u_k^y]^T$ of random accelerations, both based on the Cartesian x-y plane. We assume Gaussian system noise $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, with $\mathbf{Q}_k$ its diagonal 2×2 covariance matrix. The state transition matrix $\mathbf{F}$ and the state noise matrix $\mathbf{G}$ are

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix}, \tag{2}$$

where T is the measurement update rate.

The radar lies at the origin of the plane at point $(x, y) = (0, 0)$ and feeds the tracking algorithm with noisy measurements of the azimuth angle and range of the vehicle. The measurement equation is given next:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k. \tag{3}$$

The measurement vector $\mathbf{z}_k = [\theta_k\ r_k]^T$ consists of the vehicle azimuth angle and range in the polar plane. The nonlinear function $\mathbf{h}(\cdot)$ that maps the state—with the measurement—space is

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \arctan(y_k/x_k) \\ \sqrt{x_k^2 + y_k^2} \end{bmatrix}, \tag{4}$$

where the top element accounts for the azimuth angle of the vehicle and the bottom for its range, given its Cartesian position (x_k, y_k). The measurement noise vector $\mathbf{v}_k = [v_k^\theta \; v_k^R]^T$ models the radar's azimuth and range inaccuracy, where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ in which $\mathbf{R}_k$ is the diagonal 2×2 noise covariance matrix.

Generally in vehicle tracking we assume that some features on the ground scene of interest force locally the vehicle to move under specific patterns. Some of the features (like bridges and lakes [27]) impose hard constraints on the vehicle movement, whereas other (roads in our study) impose soft constraints. The objective in this class of problems is to incorporate efficiently a-priori knowledge of these features into the tracking algorithm.

In this work we assume that a vehicle travels on a terrain with known road structure, having the ability to move on and off the road. The roads impose probabilistic constraints on the movement of the vehicle which implies that when the vehicle is on the road the uncertainty for its state is larger along the road than orthogonal to it. We model this by setting the variance of the process noise along the road, $\sigma\{u_k^\alpha\}^2$, larger than the variance orthogonal to it $\sigma\{u_k^o\}^2$. The direction of the on-road noise depends on the direction of the road. Therefore the associated process noise covariance $\mathbf{Q}_k$ is rotated using the following relation:

$$\mathbf{Q}_{\text{on},k}(\psi) = \mathbf{\Omega}_\psi \begin{bmatrix} \sigma\{u_k^o\}^2 & 0 \\ 0 & \sigma\{u_k^\alpha\}^2 \end{bmatrix} \mathbf{\Omega}_\psi^T, \quad (5)$$

where $\mathbf{\Omega}_\psi$ is the rotational transformation matrix and ψ is the angle of the road measured clockwise from the y-axis:

$$\mathbf{\Omega}_\psi = \begin{bmatrix} -\cos\psi & \sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \quad (6)$$

For off-road motion since the vehicle travels unconstrained, we use the same process noise variances for both x- and y-axes, $\sigma\{u_k^x\}^2 = \sigma\{u_k^y\}^2$; the covariance thus becomes

$$\mathbf{Q}_{\text{off},k} = \begin{bmatrix} \sigma\{u_k^x\}^2 & 0 \\ 0 & \sigma\{u_k^y\}^2 \end{bmatrix}. \quad (7)$$

For notational purposes we define $\mathcal{R}_s$ as the set of the roads r on the ground scene of interest. For off-road motion we use the convention $r = 0$. Consider that both VSMMPF and VMPF vehicle trackers employ *nominally* N_f particles $\{\mathbf{x}_k^i\}_{i=1}^{N_f}$. In contrast to the VSMMPF which always uses N_f particles, the VMPF uses a varying number of particles which is smaller or equal to N_f. In both algorithms each particle is associated with a mode M_k^i according to the following:

$$M_k^i = \begin{cases} r & \text{if particle } \mathbf{x}_k^i \text{ is on the road } r, \text{ where } r \in \mathcal{R}_s, \\ 0 & \text{if particle } \mathbf{x}_k^i \text{ is off-road.} \end{cases} \quad (8)$$

For instance, if in the simulation scenario the vehicle can move freely among three roads ($\mathcal{R}_s = \{1, 2, 3\}$) and can also travel off-road, each particle $\mathbf{x}_k^i$ will be assigned with one of the possible modes: $M_k^i = 1, 2, 3,$ *or* 0. For further analysis and examples of this modal approach and a description of the VSMMPF algorithm, please refer to [12, 17]. Next we introduce and discuss the variable-mass particle allocation principle.

3. VARIABLE-MASS TECHNIQUE

This section introduces the variable-mass mechanism and discusses its strengths and benefits.

3.1. The proposed approach

In this part we first summarise the VSMMPF logic for allocating the particles to the multiple modes and then introduce the VMPF approach. Consider an n_m-mode particle filter which at time $k-1$ has $N_{\alpha,k-1}$ particles at mode α. At k each particle can either continue on the same mode or switch to another. Let the known a priori probability switching[1] from mode α to mode β be $p_{\alpha\to\beta} \in \mathbb{R}[0,1]$,[2] where $\alpha, \beta \in \mathbb{N}[1, n_m]$; $\mathbb{R}$ and $\mathbb{N}$ are, respectively, the sets of the real and natural numbers. According to the VSMMPF, the number of the transferred particles to a mode is proportional to the fixed prior mode probability:

$$N_{\alpha\to\beta,k} = \left| \left\{ \nu_i < p_{\alpha\to\beta} : \{\nu_i : \nu_i \sim \mathcal{U}(0,1)\}_{i=1}^{N_{\alpha,k-1}} \right\} \right|, \quad (9)$$

where $N_{\alpha\to\beta,k}$ is the number of the particles that are transferred from mode α to mode β at k and $\mathcal{U}(\cdot,\cdot)$ stands for the uniform distribution. For a large number of particles, we have

$$\lim_{N_{\alpha,k-1}\to\infty} N_{\alpha\to\beta,k}|_{N_{\alpha,k-1}} = p_{\alpha\to\beta} \cdot N_{\alpha,k-1}, \quad (10)$$

which indicates that on average we get

$$\overline{N}_{\alpha\to\beta,k} = p_{\alpha\to\beta} \cdot N_{\alpha,k-1}. \quad (11)$$

Furthermore, for the VSMMPF it holds that

$$\sum_{\beta=1}^{n_m} N_{\alpha\to\beta,k} = N_{\alpha,k-1}, \quad \forall \alpha, \quad (12)$$

which implies that the overall number of its particles remains constant.

Consider again the n_m-mode particle filter defined previously. In the VMPF, we can change the number of the particles according to an *arbitrary* defined probabilistic parameter, $\gamma_{\alpha\to\beta,k} \in \mathbb{R}[0,1]$, which we call *gamma* metric:

$$N'_{\alpha\to\beta,k} = \gamma_{\alpha\to\beta,k} \cdot N_{\alpha,k-1}, \quad (13)$$

[1] A *switch* from mode α to β refers to a change of the particle propagation model from the one of mode α to β.

[2] The case $\beta = \alpha$ refers to continuation on the same mode.

where $N'_{\alpha\to\beta,k}$ is the transferred number of particles from mode α to β at k. For $\gamma_{\alpha\to\beta,k}$, it holds

$$\sum_{\beta=1}^{n_m} \gamma_{\alpha\to\beta,k} = 1, \quad \forall \alpha, k. \tag{14}$$

We define $m_{\alpha\to\beta,k}$ as the *mass* of the particles that are transferred from mode α to β at k:

$$m_{\alpha\to\beta,k} = \frac{p_{\alpha\to\beta}}{\gamma_{\alpha\to\beta,k}} = p_{\alpha\to\beta}\cdot\frac{N_{\alpha,k-1}}{N'_{\alpha\to\beta,k}}. \tag{15}$$

The masses are used to rescale the weights of the particles, so as the arbitrary particle allocation not to bias the final estimates (if the weights were left unscaled, then the state estimate would be biased towards the modes which the *gamma* metric "favoured").

In contrast to the VSMMPF, see (12), the total number of the VMPF particles is allowed to vary:

$$\sum_{\beta=1}^{n_m} N'_{\alpha\to\beta,k} \neq N_{\alpha,k-1}, \quad \forall \alpha. \tag{16}$$

A stepwise algorithm for the variable-mass technique for a general multimodel particle filter is given in the appendix.

3.2. Justification

Equation (13) is the key to the proposed particle allocation scheme, which (a) enables the particles to be allocated to their modes more deterministically than within the VSMMPF, and (b) allows the proportion of the allocated particles to vary with time k. With this features the algorithm can *precisely* and *freely* allocate the number of its particles to the different modes at each k. The assignment of the particles with appropriate masses keeps the estimates unbiased from the arbitrary particle allocation.

Essentially, the variable-mass mechanism introduces another degree of freedom to the estimation process, by employing particle triples consisting of {state, weight, mass}. The extra degree of freedom, the mass, enables the estimator to exploit *indirectly* additional information, which is expected to increase the efficiency of the particles, affecting both the estimation accuracy and the computational load of the tracker. This additional information might concern, for instance, the estimation difficulty of particular subspaces of the estimation space. The algorithm, thus, can use fewer particles in a mode which has relatively simple and linear state prediction dynamics. In contrast it can use more particles when the mode dynamics are more difficult due to intense model nonlinearities and/or multimodalities of the posterior-state probability density function (pdf). The extra information can also concern directly the measurements. For instance, if a measurement indicates that a mode is highly unlikely (i.e., its particles will be most probably assigned with negligible weights), the algorithm can allocate fewer particles to it and more to the more likely modes, so as totally the particles to be assigned with bigger weights and thus contribute more to the state estimation process.

Overall, the proposed approach can be described as an eclectic *spatial* enhancement or degradation of the *resolution* of the discrete approximation of the posterior-state pdf, $p(\mathbf{x}_k, \mathbf{z}_k)$. This manipulation of the resolution, or else of the particles' density, is allowed since the variable masses rescale appropriately the particles' weights for debiasing the final estimate. It is characterised as "spatial" since it alters the particle density only on specific areas, in contrast to "universal" which would imply simply the change of the total number of particles N_f.

4. VARIABLE-MASS PARTICLE FILTER

We begin this section by outlining the features of the vehicle-tracking VMPF and then we describe in detail how the specific algorithm works.

4.1. Features of the vehicle tracker

The VMPF employs the varying mass technique for propagating its on-road particles on and off the road. Specifically for these particles, the tracker uses as the *gamma* metric an approximation of the *posterior*-mode probabilities, obtained by fusing the fixed *prior* mode probabilities with the varying modes' *likelihoods* conditioned on the current measurement. As described before, the varying masses that the algorithm uses, compensate for the resulting over- or under-population of its modes. The fact that in contrast to the VSMMPF, the VMPF is not "blind" to the measurements when allocating its on-road particles to their corresponding modes results in a more efficient particle use, which translates consequently to a performance improvement. For the off-road particles, both algorithms use a similar propagation mechanisms.

Another feature of the new vehicle tracker is that it employs just one particle on the road. This is because the on-road dynamics are easier to estimate due to the soft constraints that the roads themselves impose [28]. Following the varying-mass logic, the mass of that on-road particle is proportional to the posterior probability of the on-road mode. Compared to the VSMMP, the fact that the variable mass approach allows the tracker to use just one particle for this mode, results in significant computational gains when the vehicle travels on the road.

For the prediction of the on-road particle the VMPF employs a Kalman filter. For running the KF, it converts the 2D polar radar measurements to 1D Cartesian pseudomeasurements (approximated as Gaussian) that lie in the middle of the road. The KF operates in a reduced-dimension 2D state-space along the middle of the road and feeds the tracker with estimates of the mean and covariance of the on-road states. These estimates are transformed and placed into the original 4D tracking state-space to finally form the on-road particle. The estimated on-road probability distribution from the KF is used also in the prediction step, to *draw* particles randomly and propagate them off the road. The number of these departing particles is determined from the posterior road-exit mode probabilities.

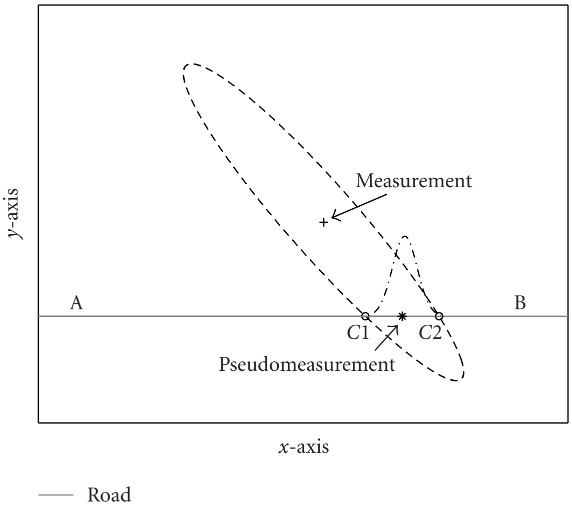

Figure 2: The skewed ellipse (dashed line) around the measurement $\mathbf{z}_k^c$ is a vertical section of the measurement pdf. The pseudomeasurement, $\hat{z}_{\mathrm{on},k}$, is set on the mode of the distribution resulting from the cross-section of line AB (the middle of the road) with the measurement pdf and is fit with a one-dimensional Gaussian pdf (dot-dashed line, rotated 90° for illustration).

4.2. The algorithm

For the sake of clarity, we do not consider a junction or bridge prediction model as in [12] and focus just on an environment with a vehicle travelling on and off nonintersecting roads. The VMPF consists of a prediction, an update, and a resampling step, which we describe next.

4.2.1. Prediction step

In the prediction step, the algorithm predicts the particles one step ahead according to their mode dynamics. First we describe the prediction phase for the road particles and then for the off-road particles.

Prediction of the on-road particles

This phase consist of the prediction of the on-road particles which either continue on the road or depart from it. We employ one particle for modelling the on-road motion. For the on-road prediction, we first generate an on-road pseudomeasurement $\hat{z}_{\mathrm{on},k}$ with its associated variance and then apply a KF. We consider Figure 2 assuming that line AB lies in the middle of the road. For clarity and simplicity in our analysis, the roads are set parallel to the x-axis.

At time instant k, we receive a radar measurement $\mathbf{z}_k = [\theta_k \; r_k]^T$ which we transform to the Cartesian plane to obtain $\mathbf{z}_k^c$:

$$\mathbf{z}_k^c = h^{-1}(\mathbf{z}_k) = \begin{bmatrix} r_k \cdot \cos\theta_k \\ r_k \cdot \sin\theta_k \end{bmatrix}. \tag{17}$$

The skewed ellipse around $\mathbf{z}_k^c$ at Figure 2, is the n_σth standard deviation $(\hat{\sigma}_{z,k})$ confidence interval of the measurement noise, after being transformed to the Cartesian plane using function $\mathbf{h}^{-1}(\cdot)$ from (17). $C1 = (x_{C1}, y_{C1})$ and $C2 = (x_{C2}, y_{C2})$ are the cross-section points of the interval and the middle of the road. The value of n_σ is chosen arbitrary (usually 3-4) since later (18) cancels it out.

The assumption here is that the cross section of line AB and the 2D skewed-Gaussian measurement noise pdf can be approximated as a 1D Gaussian pdf along AB. Therefore, since we are also using a *linear* constant velocity vehicle model, we track on-road on a reduced state-space (along AB) with a 2D Kalman filter. The tracking space of the KF consists of the vehicle's position $x_{\mathrm{on},k}$ and velocity $\dot{x}_{\mathrm{on},k}$ *just* along the middle of the road. This is because an attempt to track any possible on-road movement orthogonal to the road will have negligible significance; especially since the roads seem to have *zero* width when the radar is far.

For computing the pseudomeasurement $\hat{z}_{\mathrm{on},k}$ on AB we find the point within the segment $C1C2$ which maximises the measurement likelihood (i.e., the statistical *mode*) and fit to it a Gaussian pdf. The standard deviation of the pdf can be approximated numerically as

$$\hat{\sigma}_{z,\mathrm{on},k} = \frac{|x_{C1} - x_{C2}|}{n_\sigma}. \tag{18}$$

Using $\hat{z}_{\mathrm{on},k}$, we predict the on-road particle $\mathbf{x}_{\mathrm{on},k-1}^{2D}$ one step ahead with the following set of KF equations:

$$\begin{aligned} \mathbf{x}_{\mathrm{on},k}^{2D-} &= \mathbf{F}_{\mathrm{on}} \cdot x_{\mathrm{on},k-1}^{2D}, \\ \mathbf{P}_{\mathrm{on},k}^{-} &= \mathbf{F}_{\mathrm{on}} \cdot \mathbf{P}_{\mathrm{on},k-1} \cdot \mathbf{F}_{\mathrm{on}}^T + \mathbf{G}_{\mathrm{on}} \cdot Q_{\mathrm{on}} \cdot \mathbf{G}_{\mathrm{on}}^T, \\ \mathbf{K}_k &= \mathbf{P}_{\mathrm{on},k}^{-} \cdot \mathbf{H}_{\mathrm{on}} \cdot \left[\mathbf{H}_{\mathrm{on}} \cdot \mathbf{P}_{\mathrm{on},k}^{-} \cdot \mathbf{H}_{\mathrm{on}}^T + \hat{R}_{\mathrm{on},k}\right]^{-1}, \\ \mathbf{x}_{\mathrm{on},k}^{2D} &= \mathbf{x}_{\mathrm{on},k}^{2D-} + \mathbf{K}_k \cdot \left[\hat{z}_{\mathrm{on},k} - \mathbf{H}_{\mathrm{on}} \cdot \mathbf{x}_{\mathrm{on},k}^{2D-}\right], \\ \mathbf{P}_{\mathrm{on},k} &= \left[\mathbf{I} - \mathbf{K}_k \cdot \mathbf{H}_{\mathrm{on}}\right] \cdot \mathbf{P}_{\mathrm{on},k}^{-}, \end{aligned} \tag{19}$$

where

$$\mathbf{F}_{\mathrm{on}} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \; \mathbf{G}_{\mathrm{on}} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \; Q_{\mathrm{on}} = \sigma_\alpha^2, \; \mathbf{H}_{\mathrm{on}} = [0 \; 1]. \tag{20}$$

$\hat{R}_{\mathrm{on},k} = (\hat{\sigma}_{z,\mathrm{on},k})^2$ is the variance of $\hat{z}_{\mathrm{on},k}$ and $\mathbf{x}_{\mathrm{on},k}^{2D} = [x_{\mathrm{on},k} \; \dot{x}_{\mathrm{on},k}]^T$ is the truncated 2D version of the on-road particle. We augment then the $\mathbf{x}_{\mathrm{on},k}^{2D}$ and place it into the original 4D state-space:

$$\mathbf{x}_{\mathrm{on},k}^{\star} = \begin{bmatrix} x_{\mathrm{on},k} \\ y_{\mathrm{on},k} \\ \dot{x}_{\mathrm{on},k} \\ 0 \end{bmatrix}, \tag{21}$$

where $y_{\mathrm{on},k}$ is the y-axis value of the middle of the road.

Next we compute the likelihood of the vehicle continuing on the road or departing from it. For that, we employ

n_ϕ road prediction *submodes*[3] $M^j_{\phi,k}$, for the following set of propagation angles:

$$\{\phi^j\}_{j=1}^{n_\phi} = \{\phi^1, \dots, \phi^{n_\phi}\}, \tag{22}$$

where ϕ^j is the departure angle of the particles of the jth submode, measured anti-clockwise from the road. As a convention, we always set $\phi^1 = 0°$ accounting for the on-road propagation. The nominal positions $\mathbf{x}^{j-}_{\phi,k}$ of the road-prediction submodes $M^j_{\phi,k}$ are given by the following relation:

$$\mathbf{x}^{j-}_{\phi,k} = \begin{bmatrix} x_{\mathrm{on},k-1} + (x_{\mathrm{on},k} - x_{\mathrm{on},k-1}) \cdot \cos\phi^j \\ y_{\mathrm{on},k-1} + (x_{\mathrm{on},k} - x_{\mathrm{on},k-1}) \cdot \sin\phi^j \\ \dot{x}_{\mathrm{on},k-1} \cdot \cos\phi^j \\ \dot{x}_{\mathrm{on},k-1} \sin\phi^j \end{bmatrix}, \tag{23}$$

where $j \in \{1 \dots n_\phi\}$. According to (23), the $\mathbf{x}^{j-}_{\phi,k}$ are calculated by propagating from $k-1$ to k the position of the on-road particle and rotating it according to the corresponding angle ϕ^j. The probability of each submode is then computed by transforming each $\mathbf{x}^{j-}_{\phi,k}$ to the measurement space and computing its likelihood according to the measurement $\mathbf{z}_k$ and its covariance $\mathbf{R}_k$:

$$\widetilde{p}^j_{\phi,k} = p\left(M^j_{\phi,k} \mid \mathbf{z}_k\right) = \mathcal{N}\left(\mathbf{h}\left(\mathbf{x}^{j-}_{\phi,k}\right), \mathbf{R}_k\right), \tag{24}$$

where $\mathbf{h}(\cdot)$ is defined in (4). The normalised probabilities are

$$p^j_{\phi,k} = \frac{\widetilde{p}^j_{\phi,k}}{\sum_{\zeta=1}^{n_\phi} \widetilde{p}^\zeta_{\phi,k}}. \tag{25}$$

We then use a weighted sum of the *varying* $p^j_{\phi,k}$ and the *fixed* prior probability $\overline{p}$:

$$\widetilde{p}^j_k = \begin{cases} w_p \cdot \overline{p} + (1 - w_p) \cdot p^j_{\phi,k}, & j = 1 \text{ (on-road)}, \\ w_p \cdot \dfrac{(1-\overline{p})}{(n_\phi - 1)} + (1 - w_p) \cdot p^j_{\phi,k}, & j \neq 1 \text{ (on-road)}, \end{cases} \tag{26}$$

where $0 \leq w_p \leq 1$ is a user defined parameter. A value of w_p closer to 1 weights more the prior $\overline{p}$ whereas closer to 0 more the measurement-dependent $p^j_{\phi,k}$. The final normalised submode probability is given by

$$p^j_k = \frac{\widetilde{p}^j_k}{\sum_{\zeta=1}^{n_\phi} \widetilde{p}^\zeta_k}. \tag{27}$$

We use p^j_k as the *gamma* metric from (13) to calculate the number of the particles $N^j_{\phi,k}$ that we will allocate to each submode $M^j_{\phi,k}$:

$$N^j_{\phi,k} = p^j_k \cdot N_{\mathrm{on},k-1}, \tag{28}$$

where $N_{\mathrm{on},k-1}$ is the *nominal* number of the on-road particles at $k-1$ (as we will see later the resampling step spawns temporally $N_{\mathrm{on},k}$ on-road particles, which are later discarded). As described before, for the on-road submode ($j = 1$), irrespectively of (28), we are always employing one particle ($N^j_{\phi,k}|_{j=1} = 1$). Next, according to p^j_k, we predict a number of particles off the road. First, we generate the particles required by sampling the on-road state pdf ($\mathbf{P}_{\mathrm{on},k-1}$), derived from the KF at the previous time instant:

$$\{\mathbf{x}^i_{\mathrm{off},k}\}_{i=1}^{N^o_{\mathrm{off},k}} = \left\{\left[x^i_{\mathrm{off},k} \ \dot{x}^i_{\mathrm{off},k}\right]^T\right\}_{i=1}^{N^o_{\mathrm{off},k}} \sim \mathcal{N}(\mathbf{x}_{\mathrm{on},k-1}, \mathbf{P}_{\mathrm{on},k-1}), \tag{29}$$

where $N^o_{\mathrm{off},k} = \sum_{j=2}^{n_\phi} N^j_{\phi,k}$ The new-born particles $\{\mathbf{x}^i_{\mathrm{off},k}\}_{i=1}^{N^o_{\mathrm{off},k}}$ which initially lie on the road are propagated off the road according to the mode departure angles $\{\phi^j\}_{j=1}^{n_\phi}$, using the relation below:

$$\mathbf{x}^{ij}_{\mathrm{off},k} = \begin{bmatrix} \dfrac{x^i_{\mathrm{off},k} \cdot \tan\phi^j - x_{\mathrm{on},k-1} \cdot \tan(\phi^j/2)}{\tan\phi^j - \tan(\phi^j/2)} \\ \dfrac{\tan\phi^j \cdot (x^i_{\mathrm{off},k} \cdot \tan\phi^j - x_{\mathrm{on},k-1} \cdot \tan(\phi^j/2))}{(\tan\phi^j - \tan(\phi^j/2))} \\ -x^i_{\mathrm{off},k} \cdot \tan\phi^j + y_{\mathrm{on},k-1} \\ \dot{x}^i_{\mathrm{off},k} \cdot \cos\phi^j \\ \dot{x}^i_{\mathrm{off},k} \cdot \sin\phi^j \end{bmatrix}. \tag{30}$$

Finally, we partition the resulting particles to the ones that lie right (clockwise), $\{\mathbf{x}^{\zeta\star,R}_{\mathrm{off},k}\}_{\zeta=1}^{N^R_{\mathrm{off},k}}$, and left (anti-clockwise), $\{\mathbf{x}^{\zeta\star,L}_{\mathrm{off},k}\}_{\zeta=1}^{N^L_{\mathrm{off},k}}$, from the road. For them it holds

$$\left\{\left\{\mathbf{x}^{\zeta\star,R}_{\mathrm{off},k}\right\}_{\zeta=1}^{N^R_{\mathrm{off},k}}, \left\{\mathbf{x}^{\zeta\star,L}_{\mathrm{off},k}\right\}_{\zeta=1}^{N^L_{\mathrm{off},k}}\right\} = \left\{\left\{\mathbf{x}^{ij}_{\mathrm{off},k}\right\}_{i=1}^{N^j_{\phi,k}}\right\}_{j=2}^{n_\phi}, \tag{31}$$

where $N^R_{\mathrm{off},k} + N^L_{\mathrm{off},k} = N^o_{\mathrm{off},k}$.

Prediction of the off-road particles

We continue with the second phase and we predict the particles which were off-road at $k-1$ (i.e., $M^i_{k-1} = 0$) following the

[3] If a particle which at $k-1$ is lying on the road, r (i.e., $M^i_{k-1} = r$) is to be propagated with the VSMMPF, there are two possibilities: either to continue on the same road ($M^i_k = M^i_{k-1} = r$) or to depart from it ($M^i_k = 0$). For the latter case, the VSMMPF just uses the mode-transition probability $p_{r\to 0}$. The particular version of the VMPF that we study here accounts for $n_\phi - 1$ (since $\phi^1 = 0°$) different road exit angles. Thus, in contrast to the VSMMPF, rather than using one mode-transition probability for road departure, the $p_{r\to 0}$, the VMPF employs $n_\phi - 1$, the $p_{r\to M^2_{\phi,k}}, p_{r\to M^3_{\phi,k}}, \dots, p_{r\to M^{n_\phi}_{\phi,k}}$, or for convenience $\{p^j_k\}_{j=2}^{n_\phi}$. This is why we prefer to use the term *sub*mode for the $M^j_{\phi,k}$—since all the $\{p^j_k\}_{j=2}^{n_\phi}$ are *sub*cases of $p_{r\to 0}$. Regarding the case that the particle stays on the road, the probability p^1_k is equivalent to $p_{r\to r}$. Therefore, note that there is not any *qualitative* difference between the terms "mode" and "submode" in this article, and the specific terminology is used just for the sake of consistency.

off-road prediction scheme of the VSMMPF. Consider that we have $N_{\text{off},k}$ such particles. We preliminary propagate every particle with equation

$$\mathbf{x}_{\text{off},k}^{i-} = \mathbf{F}\mathbf{x}_{k-1}^{i}. \tag{32}$$

We introduce then the following binary function:

$$c(\mathbf{x}_{k-1}^{i}, r) = \begin{cases} 1 & \text{if } \mathbf{x}_{k-1}^{i} \longrightarrow \mathbf{x}_{\text{off},k}^{i-} \text{ crosses road } r \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

The mode transition probabilities ($p_{M_{k-1}^{i} \to M_{k}^{i}}$) are given by

$$p_{0\to r}(\mathbf{x}_{k-1}^{i}) = \begin{cases} p^{\star} & \text{if } c(\mathbf{x}_{k-1}^{i}, r) = 1, \\ 0 & \text{if } c(\mathbf{x}_{k-1}^{i}, r) = 0, \\ & \quad d(\mathbf{x}_{\text{off},k}^{i-}, r) > \tau, \\ \dfrac{p^{\star}\left(\tau - d(\mathbf{x}_{\text{off},k}^{i-}, r)\right)}{\tau} & \text{otherwise,} \end{cases} \tag{34}$$

where $p^{\star}$ is the user-defined probability that the vehicle enters a road when crossing, $d(\mathbf{x}_{\text{off},k}^{i-}, r)$ is the shortest distance from particle $\mathbf{x}_{\text{off},k}^{i-}$ to the road r, and τ is a user defined threshold according to the acceleration capabilities of the vehicle. The probability that the particle will remain off-road is

$$p_{0\to 0}(\mathbf{x}_{k-1}^{i}) = 1 - p_{0\to r}(\mathbf{x}_{k-1}^{i}). \tag{35}$$

The mode $M_{k}^{i\star}$ is randomly drawn according to the associated transition probabilities:

$$P\{M_{k}^{i\star} = r\} = p_{M_{k-1}^{i} \to r}\big|_{r\in\{0,R_s\}}. \tag{36}$$

If $M_{k}^{i\star} = 0$, the mode implies that the particle stays off the road and therefore we propagate it simply by using the state transition equation with a random noise sample $\mathbf{u}_{k-1}^{i}$:

$$\mathbf{x}_{\text{off},k}^{i\star} = \mathbf{F}\mathbf{x}_{k-1}^{i} + \mathbf{G}\mathbf{u}_{k-1}^{i}. \tag{37}$$

If $M_{k}^{i\star} \neq 0$, the particle is positioned at the shortest point on the road and its velocity is rotated, using the rotation matrix (6), randomly towards one road direction. All predicted particles from this phase are denoted as $\{\mathbf{x}_{\text{off},k}^{i\star}\}_{i=1}^{N_{\text{off},k}}$.

The resulting set of the particles from the prediction step finally becomes

$$\{\mathbf{x}_{k}^{i\star}\}_{i=1}^{N_{v,k}} = \left\{\mathbf{x}_{\text{on},k}^{\star}, \{\mathbf{x}_{\text{off},k}^{\zeta\star,R}\}_{\zeta=1}^{N_{\text{off},k}^{R}}, \{\mathbf{x}_{\text{off},k}^{\zeta\star,L}\}_{\zeta=1}^{N_{\text{off},k}^{L}}, \{\mathbf{x}_{\text{off},k}^{\zeta\star}\}_{\zeta=1}^{N_{\text{off},k}}\right\}, \tag{38}$$

where $N_{v,k}$ stands for the total number of particles that the VMPF uses at the specific time instant k:

$$N_{v,k} = 1 + N_{\text{off}+,k} + N_{\text{off},k}. \tag{39}$$

4.2.2. *Update step*

At the beginning of the update step we weight each particle in the VSMMPF fashion:

$$\widetilde{w}_{k}^{i} = p(\mathbf{z}_k \mid \mathbf{x}_{k}^{i\star}) = \mathcal{N}(\mathbf{h}(\mathbf{x}_{k}^{i\star}), \mathbf{R}_k), \tag{40}$$

and we normalise its weight:

$$w_{k}^{i} = \frac{\widetilde{w}_{k}^{i}}{\sum_{j=1}^{N_{v,k}} \widetilde{w}_{k}^{j}}, \tag{41}$$

where in analogy with (31) we obtain

$$\left\{w_{k}^{i}\right\}_{i=1}^{N_{v,k}} = \left\{w_{\text{on},k}, \left\{w_{\text{off},k}^{\zeta,R}\right\}_{\zeta=1}^{N_{\text{off},k}^{R}}, \left\{w_{\text{off},k}^{\zeta,L}\right\}_{\zeta=1}^{N_{\text{off},k}^{L}}, \left\{w_{\text{off},k}^{\zeta}\right\}_{\zeta=1}^{N_{\text{off},k}}\right\}. \tag{42}$$

At this point we calculate the particles' masses. Just for illustration, we present once more the relation (15) which we use to compute the masses:

$$m_{\alpha\to\beta,k} = p_{\alpha\to\beta} \cdot \frac{N_{\alpha,k-1}}{N'_{\alpha\to\beta,k}}. \tag{43}$$

The particles obtain a mass according to the subset in which they belong. The mass of the on-road particle is

$$m_{\text{on},k} = \overline{p} \cdot \frac{N_{\text{on},k-1}}{1}, \tag{44}$$

since at $k-1$ we nominally had $N_{\text{on},k-1}$ particles on road, $\overline{p}$ was the probability for the particles to remain on-road and the current mode uses one particle.

The masses of the particles that were predicted departing from the road are

$$\begin{aligned} m_{\text{off},k}^{R} &= \frac{1-\overline{p}}{2} \cdot \frac{N_{\text{on},k-1}}{N_{\text{off},k}^{R}}, \\ m_{\text{off},k}^{L} &= \frac{1-\overline{p}}{2} \cdot \frac{N_{\text{on},k-1}}{N_{\text{off},k}^{L}}. \end{aligned} \tag{45}$$

Using the same logic as before, we had previously $N_{\text{on},k-1}$ particles on the road, $(1-\overline{p})/2$ was the probability for the particles to exit either right of left the road and $N_{\text{off},k}^{R}$ and $N_{\text{off},k}^{L}$ was their respective number.

For the particles that were off-road at $k-1$, using a varying-mass analogy, we argue that their prediction was within a single mode and consequently are set with unitary masses:

$$m_{\text{off},k} = 1 \cdot \frac{N_{\text{off},k-1}}{N_{\text{off},k}} = 1. \tag{46}$$

We derive then the *scaled* weights of the particles by multiply them with their corresponding masses:

$$\begin{aligned} \widetilde{w}'_{\text{on},k} &= m_{\text{on},k} \cdot w_{\text{on},k}, \\ \left\{\widetilde{w}'^{i,R}_{\text{off},k}\right\}_{i=1}^{N_{\text{off},k}^{R}} &= m_{\text{off},k}^{R} \cdot \left\{w_{\text{off},k}^{i,R}\right\}_{i=1}^{N_{\text{off},k}^{R}}, \\ \left\{\widetilde{w}'^{i,L}_{\text{off},k}\right\}_{i=1}^{N_{\text{off},k}^{L}} &= m_{\text{off},k}^{L} \cdot \left\{w_{\text{off},k}^{i,L}\right\}_{i=1}^{N_{\text{off},k}^{L}}, \\ \left\{\widetilde{w}_{\text{off},k}^{i}\right\}_{i=1}^{N_{\text{off},k}} &= m_{\text{off},k} \cdot \left\{w_{\text{off},k}^{i}\right\}_{i=1}^{N_{\text{off},k}}, \end{aligned} \tag{47}$$

which are subsequently normalised to sum to 1:

$$w_k'^i = \frac{\widetilde{w}_k'^i}{\sum_{j=1}^{N_{v,k}} \widetilde{w}_k'^j}, \tag{48}$$

where

$$\begin{aligned}\left\{\widetilde{w}_k'^i\right\}_{i=1}^{N_{v,k}} = \Big\{\widetilde{w}'_{\text{on},k}, &\left\{\widetilde{w}_{\text{off},k}'^{\zeta,R}\right\}_{\zeta=1}^{N_{\text{off},k}^R},\\ &\left\{\widetilde{w}_{\text{off},k}'^{\zeta,L}\right\}_{\zeta=1}^{N_{\text{off},k}^L}, \left\{\widetilde{w}_{\text{off},k}'^{\zeta}\right\}_{\zeta=1}^{N_{\text{off},k}}\Big\}.\end{aligned} \tag{49}$$

The state estimate at k is finally given by the weighted sum of the particles:

$$\widehat{\mathbf{x}}_k = \sum_{i=1}^{N_{v,k}} w_k'^i \mathbf{x}_k^{i\star}. \tag{50}$$

4.2.3. Resampling step

The next step is to resample the weighted particle set to discard particles with small weights. The order of the particles and their weights should remain unaltered as in (31) and (42). We use the systematic resampling algorithm (see Algorithm 1), modified accordingly for the VMPF (see above for the pseudo-code). Its characteristic now is that it treats the on-road particle as the *parent* of multiple particles with the same states, with multiplicity proportional to the on-road mass $m_{\text{on},k}$. For this reason, we use the unscaled versions of the weights as computed in (41). After resampling, the size of the resulted resampled particle set $\{\mathbf{x}_k^i\}_{i=1}^{N_f}$ is increased from $N_{v,k}$ to N_f and all particles obtain equal weights and masses.

The final step of VMPF is to re-estimate the states of the on-road particle, accounting for particles that might have entered the road. Let us assume that after resampling $N_{\text{on},k}$ particles lie on the road $\{\mathbf{x}_{\text{on},k}^{i,r}\}_{i=1}^{N_{\text{on},k}}$. Since these post-resampling particles have equal weights, the characterisation of the on-road posterior pdf is given just by their density. For computing the final posterior on-road particle, $\mathbf{x}_{\text{on},k}$, under the assumption of Gaussianity, we simply calculate the mean state of $\{\mathbf{x}_{\text{on},k}^{i,r}\}_{i=1}^{N_{\text{on},k}}$:

$$\mathbf{x}_{\text{on},k} = \frac{1}{N_{\text{on},k}} \cdot \sum_{i=1}^{N_{\text{on},k}} \mathbf{x}_{\text{on},k}^{i,r}. \tag{51}$$

Only the $\mathbf{x}_{\text{on},k}$ is forwarded to the next time step $k+1$ while the set $\{\mathbf{x}_{\text{on},k}^{i,r}\}_{i=1}^{N_{\text{on},k}}$ is discarded.

5. SIMULATION RESULTS

In this section we study the performance of the tracking algorithms using the road structure of Figure 1. For a fair comparison we use the same parameters as in [12, 22]. The vehicle is moving along points A, B, C and D. It moves on-road along segments AB and CD and off-road along BC. In the Monte Carlo (MC) runs that we perform, we vary the angle of departure φ randomly uniformly between $20^\circ < \varphi < 160^\circ$.

Set nominal number of on-road particles:
 $N_{\text{on},k}^{\text{res}} = N_f - N_{v,k} + 1$
Initialise the cumulative density function
 (cdf) of the weights: $c_{1=w_k^1}$
for $i = 2 : N_{\text{on},k}^{\text{res}}$ **do**
Construct cdf: $c_i = c_{i-1} + c_1$
end for
for $i = (N_{\text{on},k}^{\text{res}} + 1) : N_f$ **do**
Construct cdf: $c_i = c_{i-1} + w_k^{(i-N_{\text{on},k}^{\text{res}}+1)}$
end for
Start at the bottom of the cdf: $i = 1$
Draw a starting point: $u_1 \sim \mathcal{U}(0, c_{N_f}/N_f)$
for $j = 1 : N_f$ **do**
Move along the cdf:
 $u_j = u_1 + (c_{N_f}/N_f)\cdot(j-1)$
while $u_j > c_j$ **do**
$i = i + 1$
end while
if $i < N_{\text{on},k}^{\text{res}} + 1$ **then**
Assign sample: $x_k^j = x_k^{i\star}$
else
Assign sample: $x_k^j = x_k^{(i-N_{\text{on},k}^{\text{res}}+1)\star}$
end if
end for

Algorithm 1: VMPF resampling.

The total simulation steps are 60 (20 for each segment) and the radar update rate is $T = 5$ seconds. The width of the road is 8 m.

The nominal velocity of the vehicle is 12 m/s which on-road is perturbed along its direction by random accelerations with standard deviation $\sigma_a = 0.6\,\text{m/s}^2$. The radar has angular accuracy 0.5° and range resolution 20 m. The standard deviation of the process noise is $\sigma_x = \sigma_y = 0.6\,\text{m/s}^2$ (off-road) and $\sigma_o = 0.0001\,\text{m/s}^2$ (orthogonal to the road). We set the mode probabilities $\overline{p} = p^* = 0.98$ and the threshold $\tau = 18.75$ For the VMPF we set $w_p = 0.5$, in (26), weighting thus equally the prior and the measurement-dependent mode probabilities. A smaller w_p value would improve the transition from on- to off-road and worsen the on-road performance; for a larger value the opposite would hold.

We use a VSMMPF, one VMPF with $n_\phi = 3$ (which we call VMPF$_{3\phi}$) and one VMPF with $n_\phi = 7$:

$$\begin{aligned}\{\phi^j\}_{j=1}^3 &= \{0^\circ, 90^\circ, 270^\circ\}, \{\phi^j\}_{j=1}^7\\ &= \{0^\circ, 45^\circ, 90^\circ, 125^\circ, 225^\circ, 270^\circ, 315^\circ\}.\end{aligned} \tag{52}$$

The performance gains of the VMPF$_{3\phi}$ come solely from its varying-mass structure, whereas from the VMPF come as well from the more departure angles it considers. For our analysis we vary the nominal number of the particles of the trackers: N_f = 10, 25, 50, 75, 100, 250, 500, 1000. For every N_f we perform 3000 MC runs and we measure the on- and off-road root mean square (RMS) position error, the maximum value of the position error overshoot when the vehicle departs from the road, the number of the particles

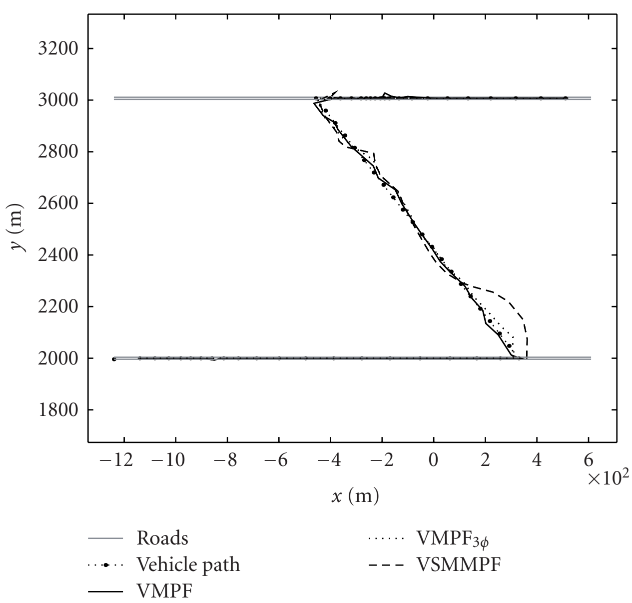

FIGURE 3: The true vehicle track and the estimates of the trackers for a representative example in which the road-departure angle is 128° and $N_f = 50$.

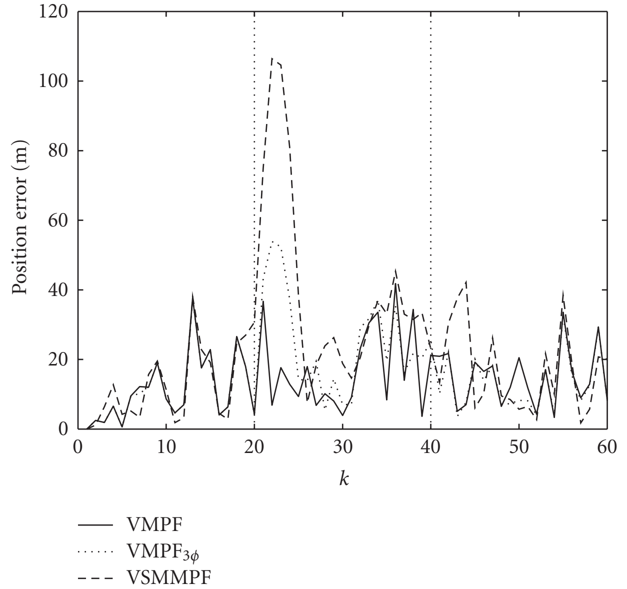

FIGURE 4: Comparison of the position error of the algorithms for the above example. The horizontal dotted lines indicate the off-road interval.

that VMPF uses, and the on-road CPU time. All algorithms were initialised by randomly seeding particles about the true states.

Figures 3 and 4 present, respectively, the vehicle tracks and the RMS position error of the three trackers, in a representative *example* in which $N_f = 50$ and $\varphi = 128°$. For the particular run, when the vehicle was on the road, both VMPF and VMPF$_{3\phi}$ employed about half of the particles that the VSMMPF used. From the figures we observe that although all algorithms attained a similar performance on-road, when the vehicle departed from the road, the transient response of the VSMMPF was considerably slower and less accurate.

Figure 5 shows the on-road RMS position error of the filters over the nominal number of the particles N_f after the MC analysis. The VMPF demonstrates better performance than the VSMMPF for $N_f < 138$, while for bigger values it converges to a slightly sub-optimal (1.1% for $N_f = 1000$) RMSE. Compared to the VMPF$_{3\phi}$, the VMPF has smaller RMSE for $N_f < 90$ because it uses more road-exit submodes and thus more particles. For $N_f > 90$, the on-road VMPF$_{3\phi}$ performance is better, because the fact that it considers just $\pm 90°$ road-exit turns, as N_f increases, makes it more robust to measurement noise. The VMPF$_{3\phi}$ improvement of the performance over the VSMMPF for $N_f > 83$ is due to the on-road Kalman filtering propagation mechanism.

From Figures 6 and 7 we witness that the off-road transient response of the VMPF during road segment BC is overall superior. We remind here that when the vehicle is off-road, the estimation schemes for both VMPF and VSMMPF converge to the same unconstrained sequential importance resampling particle filter. The difference in performance that we observe is the result of the different mechanisms for propagating off the road the on-road vehicle. From Figure 7 we see that even when $N_f = 1000$, the VMPF has 36% smaller

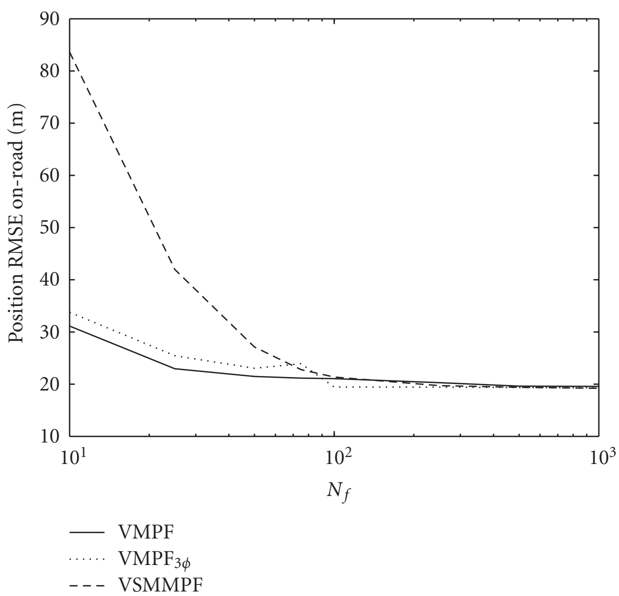

FIGURE 5: Comparison of the RMS position error when the vehicle is on-road, over the nominal number of particles N_f.

overshoot than the VSMMPF. Once more, the VMPF$_{3\phi}$ performance shows us which amount of performance improvement comes just from the varying-mass particles technique.

Figure 8 shows the percentage of the particles that the VMPF and VMPF$_{3\phi}$ use over the nominal number of particles N_f. When the vehicle is on-road, the algorithms use, respectively, about 33%–41% and 19%–29% of the N_f. When the vehicle exits the road, they rapidly increase their number of particles until reaching N_f. For continuing our analysis, we define as the particle efficiency f of VMPF over

Table 1: Particle efficiency: the ratio of the number of the VSMMPF particles to the VMPF particles for a given performance. We focus on the RMS position error, when the vehicle is on-road and off-road, and on the RMS transient overshoot, when the vehicle departs from the road.

RMSE on-road				
RMSE (m)	19.58	23.43	27.29	31.14
VSMMPF number of particles-N_f	339.77	70.62	49.62	41.46
VMPF average number of particles	337.51	8.62	5.26	4.10
Particle efficiency f	1.01	8.19	9.43	10.11
RMSE on-road				
RMSE (m)	35.55	51.66	67.77	83.88
VSMMPF number of particles-N_f	1000.00	188.19	91.18	63.62
VMPF average number of particles	240.93	33.72	16.26	9.54
Particle efficiency f	4.15	5.58	5.61	6.67
RMSE transient overshoot				
RMSE (m)	54.20	67.61	81.01	94.42
VSMMPF number of particles-N_f	1000.00	369.23	201.82	130.30
VMPF average number of particles	72.64	25.13	14.86	9.54
Particle efficiency f	13.77	14.69	13.58	13.66

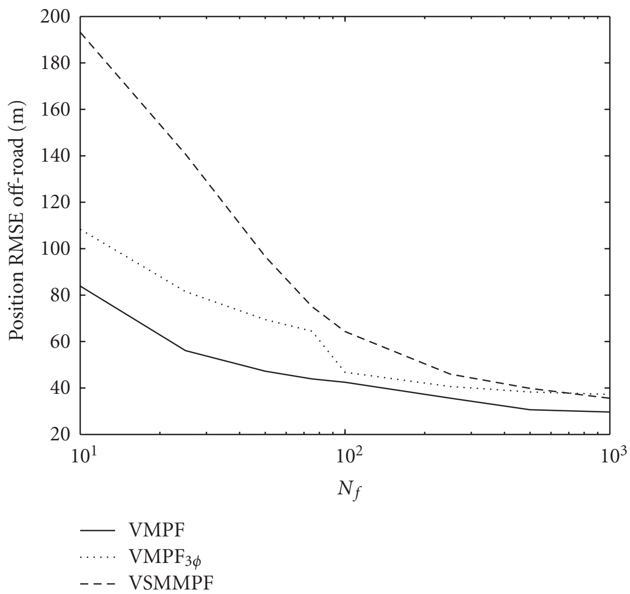

Figure 6: Comparison of the RMS position error when the vehicle is off-road, over the nominal number of particles N_f.

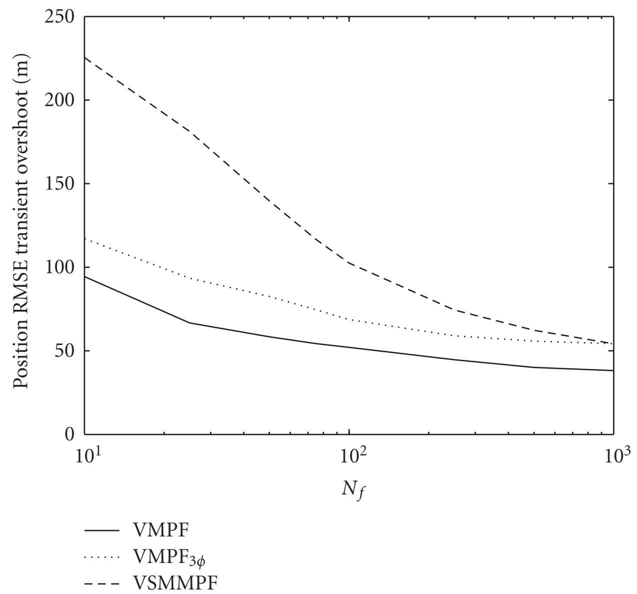

Figure 7: Comparison of the RMS position error overshoot when the vehicle departs from the road, over the nominal number of particles N_f.

VSMMPF as *the ratio of the number of the VSMMPF particles to the VMPF particles for a given performance*. For example $f(20) = 2$ for on-road RMSE indicates that the VSMMPF employs 2 times more particles than the VMPF, when both attain a 20 m on-road RMSE. Using Figures 5, 6, 7, and 8, we calculate f for the various performance metrics. The results are presented at Table 1 and demonstrate the efficiency of the proposed algorithm. In the studied scenario, the VSMMPF uses up to 14.69 times more particles than the VMPF for achieving the same performance, in the RMSE ranges within which f could be calculated.

Finally, Figure 9 compares the on-road CPU time of the algorithms (run on a Linux platform with an Intel Xeon 3 GHz processor and a 1 GB DDR2 memory). For $N_f < 40$, the VMPF trades off its on-road performance superiority compared to the VSMMPF with computing power. For larger values of N_f, the VMPF is computationally cheaper and has a CPU time linearly related to the N_f. On the road, depending on the N_f, $\text{VMPF}_{3\phi}$ requires 6%–23% less CPU time than the VMPF, while using on average almost half of the particles (Figure 8). Off the road all algorithms had the same computational demands. On the robustness of the algorithms, we observe poor performance of the VSMMPF for $N_f = 10$ and 25, where it resulted, respectively, in 40.5% and 9.1% diverged runs (resp., 8.1 and 3.7 times more than the VMPF). Nevertheless, for bigger—and more realistic—values of N_f,

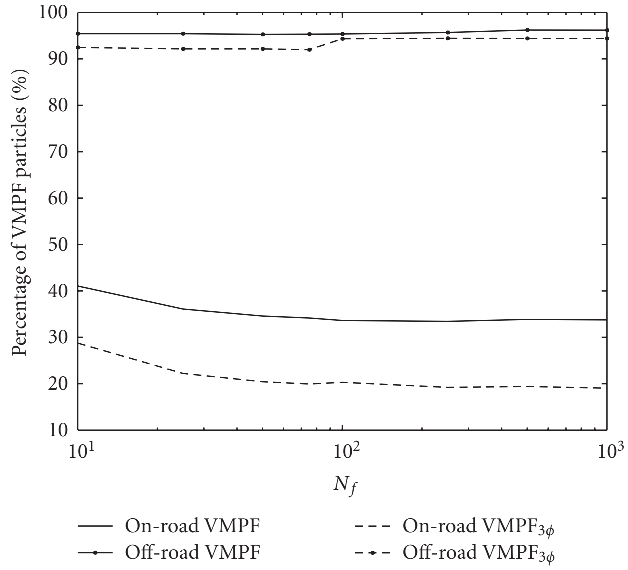

FIGURE 8: The percentage of the particles of the VMPF and VMPF$_{3\phi}$ to the particles of the VSMMPF (when the vehicle is on- and off-road), over the nominal number of particles N_f.

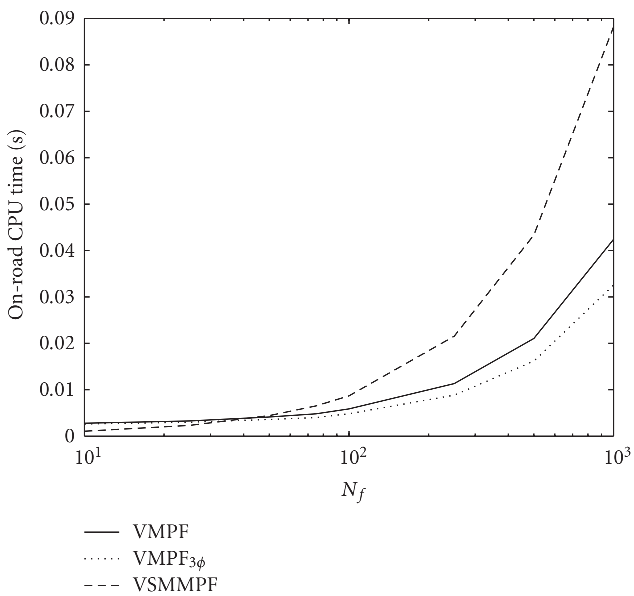

FIGURE 9: Comparison of the CPU time when the vehicle is on-road, over the nominal number of particles N_f.

Compute, where defined, the gamma metric:
for $\alpha = 1 : n_m$
  for $\beta = 1 : n_m$
    if $\alpha \rightarrow \beta$ is defined
      Compute: $\gamma_{\alpha\rightarrow\beta,k}$
    end
  end
end
Calculate the number of the particles of each mode:
for $\alpha = 1 : n_m$
  for $\beta = 1 : n_m$
    if $\alpha \rightarrow \beta$ is defined
      $N'_{\alpha\rightarrow\beta,k} = \gamma_{\alpha\rightarrow\beta,k} \cdot N_{\alpha,k-1}$
    end
  end
end
Propagate the particles according to their mode dynamics: $\{\mathbf{x}^i_{k-1}\}^{N_f}_{i=1} \rightarrow \{\mathbf{x}^{i-}_k\}^{N_f}_{i=1}$
Compute the unscaled weights:
for i $= 1 : N_f$
  $\tilde{w}^i_k = \mathcal{N}(\mathbf{h}(\mathbf{x}^{i-}_k), \mathbf{R}_k)$
end
for i $= 1 : N_f$
  $w^i_k = \tilde{w}^i_k / \sum^{N_f}_{j=1} \tilde{w}^j_k$
end
Calculate the relevant masses:
for $\alpha = 1{:}n_m$
  for $\beta = 1{:}n_m$
    if $\alpha \rightarrow \beta$ is defined
      $m_{\alpha\rightarrow\beta,k} = p_{\alpha\rightarrow\beta} \cdot N_{\alpha,k-1} / N'_{\alpha\rightarrow\beta,k}$
    end
  end
end
Scale the weights with the masses
to obtain $\{w'^i_k\}^{N_f}_{i=1}$
Derive the state estimate: $\hat{\mathbf{x}}_k = \sum^{N_f}_{i=1} w'^i_k \mathbf{x}^{i-}_k$
Resample the particles:
$[\{\mathbf{x}^i_k\}^{N_f}_{i=1}] = \text{RESAMPLE}\,[\{\mathbf{x}^{i-}_k\}^{N_f}_{i=1}, \{w^i_k\}^{N_f}_{i=1}]$.

ALGORITHM 2

both algorithm did demonstrate a robust performance. An algorithm was considered to be diverged if at any point its position error exceeded 600 m. All the simulation results presented in this section were calculated just from the converged runs.

6. CONCLUSIONS

This work introduced the variable-mass particle filter and used the terrain-aided tracking problem for comparing it with the variable-structure multimodel particle filter. Both algorithms have generic multimodel particle filtering structures which differ on their mode-switching and particle allocation mechanisms. For switching between its modes, the VSMMPF uses a fixed prior mode probability, while the VMPF employs an adaptive scheme involving varying posterior measurement-dependent mode probabilities and variable mass particles. For the studied vehicle tracking problem, the VMPF uses furthermore a reduced-dimension Kalman filter for its on-road mode and considers more angles for road departure.

Simulation results demonstrated the improved efficiency of the VMPF, since in general the new algorithm required fewer particles than the VSMMPF for achieving the same or better estimation accuracy. The variable-mass architecture enabled the vehicle tracker to incorporate efficiently the measurement information within the particle allocation mechanism which in turn resulted in a better transitional response when the vehicle was departing from the road. Moreover, the Kalman-based technique for tracking with a single on-road particle and the mechanism to spawn from it off-road particles, reduced the on-road computational demands of the algorithm. In general, the variable-mass approach can be proven a useful component of a multi-mode particle filter, allowing for a direct exploitation of available information within the particle allocation mechanism and resulting consequently in a better characterisation of the posterior state distribution.

APPENDIX

Algorithm 2 pseudoalgorithm which accounts for a fixed number of particles N_f. The number of the particles can vary by setting, for certain mode-transitions, the $N'_{\alpha\to\beta,k}$ fixed (e.g., the vehicle tracker in the paper always uses one particle for its on-road mode).

ACKNOWLEDGMENTS

The first author would like to thank Yannis Kopsinis from IDCOM for his valuable comments and suggestions on an early draft of this paper. The research was supported by BAE SYSTEMS and SELEX S & AS.

REFERENCES

[1] F. Gustafsson, F. Gunnarsson, N. Bergman, et al., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[2] D. Pham, K. Dabia, and C. Musso, "A Kalman-particle kernel filter and its application to terrain navigation," in *Proceedings of the 6th International Conference of Information Fusion*, vol. 2, pp. 1172–1179, Cairns, Queensland, Australia, July 2003.

[3] S. Gattein and P. Vannoorenberghe, "A comparative analysis of two approaches using the road network for tracking ground targets," in *Proceedings of the 7th International Conference on Information Fusion (FUSION '04)*, vol. 1, pp. 62–69, Stockholm, Sweden, June-July 2004.

[4] L. Mihaylova and R. Boel, "A particle filter for freeway traffic estimation," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 2, pp. 2106–2111, Paradise Island, Bahamas, December 2004.

[5] R. Rad and M. Jamzad, "Real time classification and tracking of multiple vehicles in highways," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1597–1607, 2005.

[6] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image and Vision Computing*, vol. 22, no. 2, pp. 143–155, 2004.

[7] A. Lauberts, M. Karlsson, F. Näsström, and R. Aljasmi, "Ground target classification using combined radar and IR with simulated data," in *Proceedings of the 7th International Conference on Information Fusion (FUSION '04)*, vol. 2, pp. 1088–1095, Stockholm, Sweden, June-July 2004.

[8] J.-H. Kim and J.-H. Oh, "A land vehicle tracking algorithm using stand-alone GPS," *Control Engineering Practice*, vol. 8, no. 10, pp. 1189–1196, 2000.

[9] E. P. Blasch and C. Yang, "Ten methods to fuse GMTI and HRRR measurements for joint tracking and identification," in *Proceedings of the 7th International Conference on Information Fusion (FUSION '04)*, vol. 2, pp. 1006–1013, Stockholm, Sweden, June-July 2004.

[10] W. Koch and R. Klemm, "Ground target tracking with STAP radar," *IEE Proceedings Radar, Sonar & Navigation*, vol. 148, no. 3, pp. 173–185, 2001.

[11] R. Sullivan, *Microwave Radar: Imaging and Advanced Concepts*, Artech House, Boston, Mass, USA, 2000.

[12] S. Arulampalam, N. Gordon, M. Orton, and B. Ristic, "A variable structure multiple model particle filter for GMTI tracking," in *Proceedings of th 5th International Conference on Information Fusion*, vol. 2, pp. 927–934, Annapolis, Md, USA, July 2002.

[13] O. Payne and A. Marrs, "An unscented particle filter for GMTI tracking," in *Proceedings of IEEE International Aerospace Conference*, vol. 3, pp. 1869–1875, Big Sky, Mont, USA, March 2004.

[14] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[15] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Technical Report CUED/F-INFENG/TR. 310, Department of Engineering, University of Cambridge, Cambridge, UK, 1998.

[16] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, New York, NY, USA, 2001.

[17] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Norwood, Mass, USA, 2004.

[18] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[19] P. Djuric, "Monte Carlo methods for signal processing: recent advances," in *Proceedings of XII. European Signal Processing Conference (EUSIPCO '04)*, p. 2310, Vienna, Austria, September 2004.

[20] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, 2005.

[21] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with topography-based variable structure IMM estimator," in *Signal and Data Processing of Small Targets*, vol. 3373 of *Proceedings of SPIE*, pp. 222–233, Orlando, Fla, USA, April 1998.

[22] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure IMM estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 26–46, 2000.

[23] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[24] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Norwood, Mass, USA, 1993.

[25] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic Control*, vol. 41, no. 4, pp. 478–493, 1996.

[26] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, Mass, USA, 1999.

[27] M. Mallick, S. Maskell, T. Kirubarajan, and N. Gordon, "Littoral tracking using particle filter," in *Proceedings of the 5th*

International Conference on Information Fusion, vol. 2, pp. 935–942, Annapolis, Md, USA, July 2002.

[28] G. Kravaritis and B. Mulgrew, "Ground tracking using a variable structure multiple model particle filter with varying number of particles," in *Proceedings of IEEE International Radar Conference*, pp. 837–841, Arlington, Mass, USA, May 2005.