This is a repository copy of *Joint target tracking and classification with particle filtering and mixture Kalman filtering using kinematic radar information*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/82261/

Version: Submitted Version

Article:
Angelova, D. and Mihaylova, L. (2006) Joint target tracking and classification with particle filtering and mixture Kalman filtering using kinematic radar information. Digital Signal Processing , 16 (2). 180 - 204. ISSN 1051-2004

https://doi.org/10.1016/j.dsp.2005.04.007

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Joint Target Tracking and Classification with Particle Filtering and Mixture Kalman Filtering Using Kinematic Radar Information

Donka Angelova [a], Lyudmila Mihaylova [*,b],

[a] *Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences, 25A Acad. G. Bonchev St, 1113 Sofia, Bulgaria*

[b] *Department of Electrical and Electronic Engineering, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK*

Abstract

This paper considers the problem of joint maneuvering target tracking and classification. Based on recently proposed Monte Carlo techniques, a multiple model (MM) particle filter and a mixture Kalman filter (MKF) are designed for two-class identification of air targets: commercial and military aircraft. The classification task is carried out by processing radar measurements only, no class (feature) measurements are used. A speed likelihood function for each class is defined using a prior information about speed constraints. Class-dependent speed likelihoods are calculated through the state estimates of each class-dependent tracker. They are combined with the kinematic measurement likelihoods in order to improve the classification process. The two designed estimators are compared and evaluated over rather complex target scenarios. The results demonstrate the usefulness of the proposed scheme for the incorporation of additional speed information. Both filters illustrate the opportunity of the particle filtering and mixture Kalman filtering to incorporate constraints in a natural way, providing reliable tracking and correct classification. Future observations contain valuable information about the current state of the dynamic systems. In the framework of the MKF, an algorithm for delayed estimation is designed for improving the current modal state estimate. It is used as an additional, more reliable information in resolving complicated classification situations.

Key Words: Joint tracking and classification, particle filtering, mixture Kalman filtering, multiple models, maneuvering target tracking

* Corresponding author
Email addresses: donka@bas.bg (Donka Angelova), mila.mihaylova@bristol.ac.uk (Lyudmila Mihaylova).

1 Introduction

Recently there has been great interest in the problem of joint target tracking and classification. It is due to the fact that the simultaneous implementation of these two important tasks in the surveillance systems facilitates the situation assessment, resource allocation and decision-making. *Classification* (or *identification*) usually includes target allegiance determination and/or target profile assessment such as *vehicle*, *ship* or *aircraft* type. Target class information could be obtained from an *electronic support measure* (*ESM*) sensor, friend-and-foe identification system, high resolution radar or other identity sensors. It can be inferred from a tracker, using kinematic measurements only or in combination with identity sensors. On the other hand, target type knowledge applied to the tracker can improve the tracking performance by the possibility of selecting appropriate target models. Classification information can assist in correct data association and false tracks elimination in multiple target tracking systems. The notion of *joint tracking and classification (JTC)* was introduced by Challa and Pulford in [1].

Several methods such as *Bayesian* [1–4], *Dempster-Shafer* [5,6] methods and *fuzzy set* theory [7] have been applied to solve different identification problems. Comparative studies of these techniques have been published in the specialized literature [5]. The inferences for their advantages and disadvantages are usually conflicting. They are highly dependent on the particularities of the problem of interest. In addition, the combination of elements from different approaches often leads to interesting results [6]. We focus here on the Bayesian methodology because it offers a theoretically valid framework for overcoming the uncertainties of the JTC. From a Bayesian point of view the problem is reduced to reconstructing of the *joint* posterior probability density function of the target state and class over time. However, the optimal solution is infeasible in practice due to the prohibitively time-consuming computations.

Suboptimal solutions can be found in different directions. One of them is related to *multiple model* estimation algorithms relying on approximate Bayesian filtering of *hybrid* dynamic systems [2,8]. The well-known Generalized Pseudo-Bayesian and the Interacting Multiple Model (IMM) estimation algorithms are essentially a bank of Kalman filters or Extended Kalman filters, which approximate highly nonlinear functions by Taylor series expansions. A Bayesian mechanism organizes the cooperation between the individual filters and the estimation of classification probabilities. An IMM-based algorithm for JTC is proposed in [3] by fusing measurements from a suite of sensors: low resolution surveillance radar, high resolution imaging radar and an ESM sensor. An alternative approximation strategy is aimed at a direct approximation of the underlying density functions. Grid-based and Monte Carlo methods are representatives of this tendency. Challa and Pulford [1] suggest a grid-based algorithm for JTC using ESM and radar data. However, the computational efficiency of the grid-based algorithms depends on the state vector dimension. In contrast to the grid-based algorithms, the Monte Carlo algorithms

are more easily implementable for systems of high dimension.

Particle filters [9,10] are sequential Monte Carlo methods based on "particle" (sample) representation of probability densities. Multiple particles of the variables of interest are generated, each associated with a weight which characterizes the quality of a specific particle. An estimate of the variable of interest is obtained by the weighted sum of particles. In the grid-based algorithms the grid points are chosen by the designer, whereas in the particle filter the particles are randomly generated according to the model of the dynamic system and then naturally follow the state. Sequential Monte Carlo algorithms are particularly suitable for classification purposes. The highly non-linear relationships between state and class measurements and non-Gaussian noise processes can be easily processed by the particle filtering technique. Moreover, flight envelope constraints, especially useful for this task, can be incorporated into the filtering algorithm in a natural and consistent way [11].

One of the first papers where particle filtering techniques are applied to tracking and identification is [12] addressing two closely spaced objects in clutter. Later other feasible implementations are reported [4,13]. Automatic target recognition and tracking of multiple targets with particle filters is proposed in [14] by the inclusion of radar cross section measurements into the measurement vector.

The main idea of using sequential Monte Carlo filtering for JTC consists of filling the state and class space with particles and then running the filtering procedure. Since the probability of switching between classes is zero, all the particles might eventually cluster around one class [4]. This is a crucial shortcoming in some cases of classification. This could happen in many practical situations, for instance, when the classification decision is not yet been taken, and all particles might be clustered around one of the classes. Therefore, the number of particles for each class should remain constant. To overcome this drawback, Gordon, Maskell and Kirubarajan [4] suggest the following algorithm for JTC: a bank of independent filters, covering the state and feature space are run in parallel with each filter matched to a different target class. The class-conditioned independent filters stay in position "alert" and the filtering system can "change its mind" regarding the class identification if changes in the target behavior occur. An example of a successful application of this approach to littoral tracking with classification is proposed in [13].

In the present paper we develop two *sequential* Monte Carlo algorithms, a particle filter (PF) and a mixture Kalman filter (MKF). They are intended for simultaneous tracking and classification of *a maneuvering target* using kinematic measurements only, primarily in the air surveillance context. Maneuvering target tracking is implemented by the *multiple model* (MM) estimation in the particle filtering framework [15–18]. Multiple models, corresponding to different regimes of flight provide reliable tracking of fast maneuvering targets. In the particle filter the multiple models correspond to different acceleration levels. Within the mixture Kalman filtering, the maneuvering target is modelled by using a random indicator variable, called latent.

The latent variable can take values over a given set of possible values depending on the acceleration range. Given the indicator variable, a set of Kalman filters realize on-line filtering based on dynamic linear models. In such a way a multiple model structure is implemented together with Monte Carlo sampling in the space of the latent variables, instead of in the space of the state variables. The paper represents a further development and generalization of the results reported in [19,20]. Two air target classes are considered: *commercial* aircraft (slowly maneuverable, mainly straight line) and *military* aircraft (highly maneuverable turns are possible). The features of the proposed algorithms include the following:

- for each target class a separate filter is designed. These filters operate in parallel, covering the class space.
- each class-dependent filter represents a *switching* multiple model filtering procedure, covering the class-dependent kinematic state space.
- two kinds of constraints are imposed on the target kinematic parameters: on the *acceleration* and on the *speed*. Two speed likelihood functions are defined based on a prior information about the speed constraints of each class. At every filtering step, the estimated speed from each class-dependent filter is used to calculate class-dependent speed likelihoods. These speed likelihoods are combined with kinematic likelihoods in order to improve the process of classification.
- Future observations contain valuable information about the current target state [21] and a delayed sampling scheme applied to fading communication channels was shown to improve the estimation accuracy. The JTC solution can achieve better performance when such a delayed estimation of the current *maneuvering mode* state is incorporated into the MKF structure. It is used as an additional, more reliable information in resolving ambiguous classification situations.

The novelty of the present paper relies on combining the multiple model (MM) approach with the particle filtering and mixture Kalman filtering for JTC purposes, the manner of imposing the speed constraints on target behavior and the use of delayed estimation for classification. Although we consider only two classes, the generalization for more target classes is straightforward.

The remaining part of the paper is organized as follows. Section 2 summarizes the Bayesian formulation of the JTC problem according to [4,13,22]. Section 3 presents the developed MM particle filter, MKF, and delayed-pilot MKF using speed and acceleration constraints. Simulation results are given in Section 5. Finally, Section 6 contains the conclusions.

2 Bayesian joint target tracking and classification

Consider the following model describing the target dynamics and sensor measurements

$$\boldsymbol{x}_k = \boldsymbol{F}(m_k)\,\boldsymbol{x}_{k-1} + \boldsymbol{G}(m_k)\,\boldsymbol{u}_k(m_k) + \boldsymbol{B}(m_k)\,\boldsymbol{w}_k, \tag{1}$$

$$\boldsymbol{z}_k = h(m_k, \boldsymbol{x}_k) + \boldsymbol{D}(m_k)\,\boldsymbol{v}_k, \qquad k = 1, 2, \ldots, \tag{2}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is the *base (continuous) state* vector with transition matrix $\boldsymbol{F}$, $\boldsymbol{z}_k \in \mathbb{R}^{n_z}$ specifies the measurement vector with measurement function h, $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ represents a known control input and k is a discrete time. The input noise process $\boldsymbol{w}_k$ and the measurement noise $\boldsymbol{v}_k$ are *independent identically distributed* Gaussian processes having characteristics $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q})$ and $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R})$, respectively. The *modal (discrete) state* m_k, characterizing the different system modes (regimes), can take values over a finite set $\mathbb{S}$, i.e. $m_k \in \mathbb{S} \triangleq \{1,\ 2,\ \ldots,\ s\}$. We assume that m_k is evolving according to a time-homogeneous first-order Markov chain with transition probabilities

$$\pi_{ij} \triangleq Pr\left\{m_k = j \mid m_{k-1} = i\right\},\ (i, j \in \mathbb{S}) \tag{3}$$

and initial probability distribution $P_0(i) \triangleq Pr\{m_0 = i\}$ for $i \in \mathbb{S}$, such that $P_0(i) \geq 0$, and $\sum_{i=1}^{s} P_0(i) = 1$. Next we suppose that the target belongs to one of M classes $c \in \boldsymbol{C}$ where $\boldsymbol{C} = \{c_1, c_2, \ldots, c_M\}$ represents the set of the target classes. Generally, the number of the discrete states $s = s(c)$, the initial probability distribution $P_0^c(i)$ and the transition probability matrix $\boldsymbol{\pi}(c) = \left[\pi_{ij}^c\right],\ i, j \in \mathbb{S}(c)$ are different for each target class. Note that the joint state and class is time varying with respect to the state and time invariant with respect to the class [4].

Denote with $\boldsymbol{\omega}_k \triangleq \{\boldsymbol{z}_k, \boldsymbol{y}_k\}$ the set of *kinematic* $\boldsymbol{z}_k$ and *class (feature)* $\boldsymbol{y}_k$ measurements obtained at time instant k. Then $\boldsymbol{\Omega}^k = \left\{\boldsymbol{Z}^k, \boldsymbol{Y}^k\right\}$ specifies the cumulative set of kinematic $\boldsymbol{Z}^k = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\}$ and feature $\boldsymbol{Y}^k = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k\}$ measurements, available up to time k.

The *goal* of the joint tracking and classification task is to estimate simultaneously the *base state* $\boldsymbol{x}_k$, the *modal state* m_k and the *posterior classification probabilities* $P\left(c \mid \boldsymbol{\Omega}^k\right), c \in \boldsymbol{C}$ based on all available measurement information $\boldsymbol{\Omega}^k$.

If we can construct the *posterior joint state-mode-class probability density function* (pdf) $p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^k\right)$, then the posterior classification probabilities can be obtained by marginalization over $\boldsymbol{x}_k$ and m_k:

$$P\left(c \mid \boldsymbol{\Omega}^k\right) = \sum_{m_k \in \mathbb{S}(c)} \int_{\boldsymbol{x}_k \in \mathbb{R}^{n_x}} p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^k\right) d\boldsymbol{x}_k. \tag{4}$$

Suppose that we know the posterior joint state-mode-class pdf $p\left(\boldsymbol{x}_{k-1}, m_{k-1}, c \mid \boldsymbol{\Omega}^{k-1}\right)$ at time instant $k-1$. According to the Bayesian framework, $p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^k\right)$ can be computed recursively from $p\left(\boldsymbol{x}_{k-1}, m_{k-1}, c \mid \boldsymbol{\Omega}^{k-1}\right)$ in two steps – *prediction* and *measurement update* [4,13].

The predicted state-mode-class pdf $p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^{k-1}\right)$ at time k is given by the

equation

$$p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^{k-1}\right) = \tag{5}$$
$$\sum_{m_{k-1}\in\mathbb{S}(c)} \int_{\boldsymbol{x}_{k-1}\in\mathbb{R}^{n_x}} p\left(\boldsymbol{x}_k, m_k \mid \boldsymbol{x}_{k-1}, m_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right) \times p\left(\boldsymbol{x}_{k-1}, m_{k-1}, c \mid \boldsymbol{\Omega}^{k-1}\right) d\boldsymbol{x}_{k-1},$$

where the state prediction pdf $p\left(\boldsymbol{x}_k, m_k \mid \boldsymbol{x}_{k-1}, m_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right)$ is obtained from the state transition equation (1)

$$\begin{aligned} &p\left(\boldsymbol{x}_k, m_k \mid \boldsymbol{x}_{k-1}, m_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right) \propto \\ &\propto \sum_{m_k\in\mathbb{S}(c)} p\left(\boldsymbol{x}_k \mid m_k, \boldsymbol{x}_{k-1}, m_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right) \times P\left(m_k \mid \boldsymbol{x}_{k-1}, m_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right) = \\ &= \sum_{m_k\in\mathbb{S}(c)} p\left(\boldsymbol{x}_k \mid m_k, \boldsymbol{x}_{k-1}, m_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right) \sum_{m_{k-1}\in\mathbb{S}(c)} \pi^c_{m_{k-1}m_k} P\left(m_{k-1} \mid \boldsymbol{x}_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right). \end{aligned} \tag{6}$$

The form of the conditional pdf of the measurements

$$p\left(\boldsymbol{\omega}_k \mid \boldsymbol{x}_k, m_k, c\right) = \lambda_{\{\boldsymbol{x}_k, m_k, c\}}\left(\boldsymbol{\omega}_k\right) \tag{7}$$

is usually known. This is the likelihood of the joint state and feature and has a key role in the classification algorithm. In our case we do not have feature measurements $\boldsymbol{y}_k,\ k = 1, 2, \ldots$. The speed estimates from the M classes, together with speed envelope constraints, whose shapes are given in Section 3.3, form a *virtual* "feature measurement" set $\{\boldsymbol{Y}^k\}$.

When the measurement $\boldsymbol{\omega}_k$ arrives, the update step can be completed

$$p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^k\right) = \frac{\lambda_{\{\boldsymbol{x}_k, m_k, c\}}\left(\boldsymbol{\omega}_k\right) p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^{k-1}\right)}{p\left(\boldsymbol{\omega}_k \mid \boldsymbol{\Omega}^{k-1}\right)}, \tag{8}$$

where

$$p\left(\boldsymbol{\omega}_k \mid \boldsymbol{\Omega}^{k-1}\right) = \sum_{c\in\boldsymbol{C}} \sum_{m_k\in\mathbb{S}(c)} \int_{\boldsymbol{x}_k\in\mathbb{R}^{n_x}} p\left(\boldsymbol{\omega}_k \mid \boldsymbol{x}_k, m_k, c\right) p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^{k-1}\right) d\boldsymbol{x}_k.$$

The recursion (5)-(8) begins with the prior density $P\left\{\boldsymbol{x}_0, m_0, c\right\}$, assumed known, where $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}, c \in \boldsymbol{C}, m_0 \in \mathbb{S}(c)$. Using Bayes' theorem, the posterior probability of the discrete state m_k for class c is expressed by

$$\begin{aligned} &P\left(m_k \mid \boldsymbol{x}_k, c, \boldsymbol{\Omega}^k\right) = \\ &\frac{1}{\alpha_k}\lambda_{\{\boldsymbol{x}_k, m_k, c\}}\left(\boldsymbol{\omega}_k\right) \times \sum_{m_{k-1}\in\mathbb{S}(c)} \int_{\boldsymbol{x}_{k-1}\in\mathbb{R}^{n_x}} \pi^c_{m_{k-1}m_k} P\left(m_{k-1} \mid \boldsymbol{x}_{k-1}, c, \boldsymbol{\Omega}^{k-1}\right) \\ &\times p(\boldsymbol{x}_{k-1}, c|\boldsymbol{\Omega}^{k-1}) d\boldsymbol{x}_{k-1}, \end{aligned} \tag{9}$$

where α_k is a normalizing constant. Eq. (9) is substituted into (6) in order to predict the state pdf at time $k+1$.

Then the target classification probability is calculated by the equation

$$P\left(c \mid \boldsymbol{\Omega}^k\right) = \frac{\sum_{m_k \in \mathbb{S}(c)} \int_{\boldsymbol{x}_k \in \mathbb{R}^{n_x}} p\left(\boldsymbol{\omega}_k \mid \boldsymbol{x}_k, m_k, c, \boldsymbol{\Omega}^{k-1}\right) p(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^{k-1}) d\boldsymbol{x}_k}{p\left(\boldsymbol{\omega}_k \mid \boldsymbol{\Omega}^{k-1}\right\}} \tag{10}$$

that can be expressed as

$$P\left(c \mid \boldsymbol{\Omega}^k\right) = \frac{p\left(\boldsymbol{\omega}_k \mid c, \boldsymbol{\Omega}^{k-1}\right) P\left(c \mid \boldsymbol{\Omega}^{k-1}\right)}{\sum_{i=1}^{M} p\left(\boldsymbol{\omega}_k \mid c_i, \boldsymbol{\Omega}^{k-1}\right) P\left(c_i \mid \boldsymbol{\Omega}^{k-1}\right)}$$

with an initial prior target classification probability $P_0(c)$, $\sum_{c \in \boldsymbol{C}} P_0(c) = 1$.

The state estimate $\hat{\boldsymbol{x}}_k^c$ for each class c

$$\hat{\boldsymbol{x}}_k^c = \sum_{m_k \in \mathbb{S}(c)} \int_{\boldsymbol{x}_k \in \mathbb{R}^{n_x}} \boldsymbol{x}_k p\left(\boldsymbol{x}_k, m_k, c \mid \boldsymbol{\Omega}^k\right) d\boldsymbol{x}_k, \ \boldsymbol{c} \in \boldsymbol{C} \tag{11}$$

takes part in the calculation of the *combined* state estimate

$$\hat{\boldsymbol{x}}_k = \sum_{c \in \boldsymbol{C}} \hat{\boldsymbol{x}}_k^c P\left(c \mid \boldsymbol{\Omega}^k\right). \tag{12}$$

It is obvious from (5)-(12) that the estimates, needed for each class, can be calculated independently from the other classes. Therefore, the JTC task can be accomplished by the simultaneous work of M independent filters.

3 Maneuvering target tracking and classification

3.1 Maneuvering target model

The two-dimensional target dynamics are given by

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{G}\left(\boldsymbol{u}_k + \boldsymbol{w}_k\right), \ k = 1, 2, \ldots, \tag{13}$$

where the state vector $\boldsymbol{x} = (x, \dot{x}, y, \dot{y})'$ contains target positions and velocities in the horizontal (Oxy) Cartesian coordinate frame. The control input vector $\boldsymbol{u} = (a_x, a_y)'$ includes target accelerations along x and y coordinates. The process noise

$\boldsymbol{w} = (w_x, w_y)'$ models perturbations in the accelerations. The transition matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are [23]

$$\boldsymbol{F} = diag\left(\boldsymbol{F}_1, \boldsymbol{F}_1\right), \ \boldsymbol{G} = diag\left(\boldsymbol{g}_1, \boldsymbol{g}_1\right),$$

where

$$\boldsymbol{F}_1 = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \quad \text{for } \boldsymbol{g}_1 = \begin{pmatrix} \frac{T^2}{2} & T \end{pmatrix}',$$

T is the sampling interval and $\boldsymbol{B} = \boldsymbol{G}$. The target is assumed to belong to one of two classes ($M = 2$), representing either a lower speed *commercial aircraft* with limited maneuvering capability (c_1) or a highly maneuvering *military aircraft* (c_2) [1]. The flight envelope information comprises speed and acceleration constrains, characterizing each class. The speed v $= \sqrt{\dot{x}^2 + \dot{y}^2}$ of each class is limited respectively to the interval:

$$\{c_1 : \text{v} \in (100, 300)\} \ [m/s] \ \text{ and } \{c_2 : \text{v} \in (150, 650)\} \ [m/s].$$

The range of the speed overlap section is $[150, 300]$ $[m/s]$. The control inputs are restricted to the following sets of accelerations:

$$\{c_1 : \boldsymbol{u} \in (0, +2g, -2g)\} \ [m/s^2] \ \text{ and } \{c_2 : \boldsymbol{u} \in (0, +5g, -5g)\} \ [m/s^2],$$

where $g = 9.81$ $[m/s^2]$ is the acceleration due to gravity. The acceleration process $\boldsymbol{u}_k$ is a Markov chain with five states (modes) $s(c_1) = s(c_2) = 5$ [24]:

$$\begin{aligned} &1.\ a_x = 0, \quad a_y = 0, \quad 2.\ a_x = A, \quad a_y = A, \quad 3.\ a_x = A, \quad a_y = -A, \\ &4.\ a_x = -A, \ a_y = A, \quad 5.\ a_x = -A, \quad a_y = -A, \end{aligned} \tag{14}$$

where $A = 2g$ stands for class c_1 target and $A = 5g$ refers to the class c_2 (as shown on Fig. 1 (a)).

The initial probabilities of the Markov chain are selected equal for the two classes as follows: $P_0(1) = 0.6$, $P_0(i) = 0.1, \ i = 2, \ldots, 5$. The matrix $\boldsymbol{\pi}(c)$ of transition probabilities $\pi^c_{ij}, i, j \in \mathbb{S}$ is assumed of the same form for both types of targets: $p_{ij} = 0.7$ for $i = j$; $p_{1j} = 0.075$ for $j = 2, \ldots, 5$; $p_{i1} = 0.15$ for $i = 2, \ldots, 5$; $p_{ij} = 0.05$ for $j \neq i, \ i, j = 2, \ldots, 5$.

The process noise is Gaussian, $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{Q})$, $\boldsymbol{Q} = diag(\sigma^2_{wx}, \sigma^2_{wy})$, having different standard deviations for each mode and class:

$$\{c_1 : \sigma^1_w = 5.5; \ \sigma^j_w = 7.5 \ [m/s^2], j = 2, \ldots, 5\} \text{ and}$$

$$\{c_2 : \sigma^1_w = 7.5, \ \sigma^j_w = 17.5 \ [m/s^2], j = 2, \ldots, 5\},$$

where ($\sigma_w = \sigma_{wx} = \sigma_{wy}$).

3.2 Measurement model

The measurement model at time k is described by

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k, \tag{15}$$

with

$$h(\boldsymbol{x}_k) = \left(\sqrt{x_k^2 + y_k^2}, \arctan\frac{x_k}{y_k}\right)', \tag{16}$$

where the measurement vector $\boldsymbol{z} = (D,\ \beta)'$ consists of the distance D to the target and bearing β, measured by the radar. The measurement error vector $\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{R})$ is assumed to be independent from $\boldsymbol{w}$. The measurement noise covariance matrix for the particle filter has the form $\boldsymbol{R} = diag(\sigma_D^2, \sigma_\beta^2)$. The measurement function $h(\boldsymbol{x})$ is highly nonlinear and can be easily processed by the particle filter.

The MKF algorithm, however, works with linear state and measurement equations. For the purposes of the MKF design, a measurement conversion is performed from polar (D , β) to Cartesian (x , y) coordinates: $\boldsymbol{z} = (D\cos(\beta),\ D\sin(\beta))'$. Thus, the measurement equation becomes linear with a $n_z \times n_x$ measurement matrix $\boldsymbol{H}$, where all elements are zeros, except for $H_{11} = H_{23} = 1$. The components of the corresponding measurement noise covariance matrix $\boldsymbol{R}$ are

$$\begin{aligned} &R_{11} = \sigma_D^2 \sin^2(\beta) + D^2\sigma_\beta^2 \cos^2(\beta); \quad R_{22} = \sigma_D^2 \cos^2(\beta) + D^2\sigma_\beta^2 \sin^2(\beta); \\ &R_{12} = (\sigma_D^2 - D^2\sigma_\beta^2)\sin(\beta)\cos(\beta); \quad R_{21} = R_{12}. \end{aligned}$$

The following sensor parameters are selected in the simulations: $\sigma_D = 100.0\ [m]$ $\sigma_\beta = 0.15\ [deg]$. The sampling interval is $T = 5\,s$.

3.3 Speed constraints

Acceleration constraints are imposed on the filter operation by the use of an appropriate control input in the target model. The speed constraints are enforced through the speed likelihood functions. They are constructed based on the speed envelope information (3.1). Such constraints are incorporated into other approaches for decision making [25]. We define the following speed likelihood functions, respectively for each class

$$g_1\left(\mathrm{v}_k^{c_1}\right) = \begin{cases} 0.9, & \text{if } \mathrm{v}_k^{c_1} \leq 100 \quad [m/s] \\ 0.9 - \kappa_1\left(\mathrm{v}_k^{c_1} - 100\right), & \text{if } (100 < \mathrm{v}_k^{c_1} \leq 300\ [m/s]) \\ 0.05 & \text{if } \mathrm{v}_k^{c_1} > 300 \quad [m/s] \end{cases}$$

and

$$g_2\left(\mathrm{v}_k^{c_2}\right) = \begin{cases} 0.1, & \text{if } \mathrm{v}_k^{c_2} \leq 150 \quad [m/s] \\ 0.1 + \kappa_2\left(\mathrm{v}_k^{c_2} - 150\right), & \text{if } \left(150 < \mathrm{v}_k^{c_2} \leq 650\ [m/s]\right) \\ 0.95, & \text{if } \mathrm{v}_k^{c_2} > 650 \quad [m/s] \end{cases}$$

where $\kappa_1 = 0.7/200$ and $\kappa_2 = 0.85/500$. Fig. 1 (b) illustrates the evolution of the likelihoods as a function of the speed.

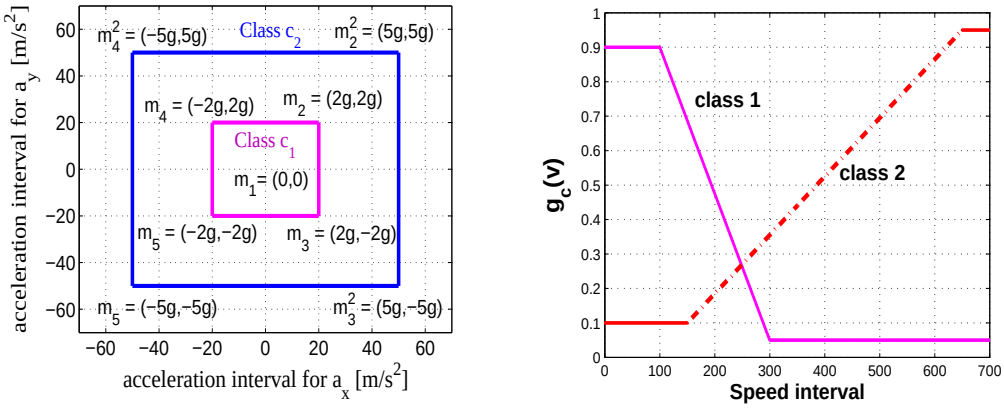

Fig. 1. (a) Mode sets for two classes (b) Speed likelihood functions

According to the problem formulation, presented in Section 2, two class-dependent filters work in parallel with N_c number of particles for each class. At time step k each filter gives a state estimate $\{\hat{\boldsymbol{x}}_k^c,\ c = 1, 2\}$. Let us assume that the estimated speed from the previous time step, $\left\{\hat{\mathrm{v}}_{k-1}^c,\ c = 1, 2\right\}$, is a kind of "feature measurement".

The likelihood $\lambda_{\{\boldsymbol{x}_k, m_k, c\}}\left(\boldsymbol{\omega}_k\right) = \lambda_{\{\boldsymbol{x}_k, m_k, c\}}\left(\{\boldsymbol{z}_k, \boldsymbol{y}_k\}\right)$ is factorized [4]

$$\lambda_{\{\boldsymbol{x}_k, m_k, c\}}\left(\{\boldsymbol{z}_k, \boldsymbol{y}_k\}\right) = f\left(\boldsymbol{z}_k | \boldsymbol{x}_k, m_k\right) g_c\left(\boldsymbol{y}_k^c\right), \tag{17}$$

where $\boldsymbol{y}_k^c = \hat{\mathrm{v}}_{k-1}^c$. Practically, the normalized speed likelihoods represent speed-based class probabilities estimated by the filters. The posterior class probabilities are modified by this additional speed information at every time step k. The inclusion of the speed likelihoods is done after some "warming-up" interval, comprising the filter initialization.

3.4 Multiple Model Particle Filter Algorithm

Within the JTC formulation problem as given in Section 2 a separate PF is designed for each target class. Assuming two classes of targets – commercial and military, we develop a bank of two independent PFs ($M = 2$). Each PF propagates a set of N_c

hybrid particles $\{\boldsymbol{x}_k^{(i)}, m_k^{(i)}\}_{i=1}^{N_c}$, containing all the necessary information about the target base state and modal state (mode). Each mode takes values from the set $\mathbb{S}(c)$. The modes evolve in time according to a Markov chain with a transition probability matrix (3). The cloud of particles for every PF allows a sequential update of the pdfs (5)-(12) by two main stages: *prediction* and *update*. During the prediction each particle is modified according to the state model, including the addition of a random noise simulating the effect of the uncertainties on the state. Then in the update stage, each particle's weight is re-evaluated based on the new sensor data. The *resampling* procedure deals with the elimination of particles with small weights and replicates the particles with higher weights.

A detailed scheme of the proposed particle filter is given in Table 1.

Table 1: A Particle Filter for Joint Target Tracking and Classification

(1) Initialization, $k = 0$.

For class $c = 1, 2, \ldots, M$ set class probabilities $P\left(c \mid \boldsymbol{\Omega}^0\right) = P_0(c)$.

* For $j = 1, 2, \ldots, N_c$,

sample $\left\{\boldsymbol{x}_0^{(j)} \sim p_0(\boldsymbol{x}_0, c),\ m_0^{(j)} \sim \{P_0^c(\ell)\}_{\ell=1}^{s(c)}\right\}$

and set initial weights $W_0^{(j)} = 1/N_c$;

* End for j

End for c

Set $k = 1$.

(2) For $c = 1, 2, \ldots, M$ (possibly in parallel) perform

* Prediction step

For $j = 1, 2, \ldots, N_c$

draw $m_k^{(j)}$ from the set $\mathbb{S}(c) = 1, 2, \ldots s(c)$ with probability

$$P\left(m_k^{(j)} = i\right) \propto \pi_{\ell\, i}^c, \qquad \text{for } \ell = m_{k-1}^{(j)},$$

draw $\boldsymbol{w}_k^{(j)} \sim \mathcal{N}(0, \boldsymbol{Q}(m_k^{(j)}, c))$,

calculate $\boldsymbol{x}_k^{(j)} = \boldsymbol{F}\boldsymbol{x}_{k-1}^{(j)} + \boldsymbol{G}\boldsymbol{u}(m_k^{(j)}, c) + \boldsymbol{G}\boldsymbol{w}_k^{(j)}$,

where $\boldsymbol{u}(m_k^{(j)}, c)$ denotes the pair of accelerations from (14) corresponding to $m_k^{(j)}$.

End for j

* Measurement processing step: on receipt of a measurement $\boldsymbol{\omega}_k = \{\boldsymbol{z}_k, y_k\}$:

For $j = 1, 2, \ldots, N_c$ evaluate the weights

$$W_k^{(j)} = W_{k-1}^{(j)} f(\boldsymbol{z}_k \mid \boldsymbol{x}_k^{(j)}) g_c\left(\boldsymbol{y}_k^c\right),$$

where $f(\boldsymbol{z}_k \mid \boldsymbol{x}_k^{(j)}) = \mathcal{N}(h(\boldsymbol{x}_k^{(j)}), \boldsymbol{R})$ and $g_c\left(\boldsymbol{y}_k^c\right) = g_c\left(\hat{\mathrm{v}}_{k-1}^c\right)$;

calculate $p\left(\boldsymbol{\omega}_k \mid c, \boldsymbol{\Omega}^{k-1}\right) = \sum_{j=1}^{N_c} W_k^{(j)}$ and set $L(c) = \sum_{j=1}^{N_c} W_k^{(j)}$

End for j

normalize the weights $W_k^{(j)} = W_k^{(j)} / \sum_{j=1}^{N_c} W_k^{(j)}$;

* <u>Compute updated state estimate and posterior mode probabilities</u>

$\hat{\boldsymbol{x}}_k^c = \sum_{j=1}^{N_c} \boldsymbol{x}_k^{(j)} W_k^{(j)}$,

$$P\left(m_k = \ell \mid c, \boldsymbol{\Omega}^k\right) = E\left\{1\left(m_k = \ell\right) \mid c, \boldsymbol{\Omega}^k\right\} \cong \sum_{j=1}^{N_c} 1(m_k^{(j)} = \ell) W_k^{(j)}, \quad \ell = 1, 2, \ldots, s(c),$$

where $1(\cdot)$ is an indicator function such that
$1(m_k = \ell) = 1$, if $m_k = \ell$ and $1(m_k = \ell) = 0$ otherwise;

* <u>Obtain a hard mode decision:</u>

$$\hat{m}_k^c = \arg\max_{\ell \in \mathbb{S}(c)} P\left(m_k = \ell \mid c, \boldsymbol{\Omega}^k\right) \cong \arg\max_{\ell \in \mathbb{S}(c)} \sum_{j=1}^{N_c} 1(m_k^{(j)} = \ell) W_k^{(j)}$$

* <u>Compute effective sample size:</u> $N_{eff}(c) = 1 / \sum_{j=1}^{N_c} \left(W_k^{(j)}\right)^2$

End for c

(3) <u>Output:</u> <u>Compute posterior class probabilities and combined output estimate</u>

$P\left(c \mid \boldsymbol{\Omega}^k\right) = \frac{L(c) P\left(c \mid \boldsymbol{\Omega}^{k-1}\right)}{\sum_{i=1}^{M} L(c_i) P\left(c_i \mid \boldsymbol{\Omega}^{k-1}\right)}$, $c = 1, 2, \ldots, M$

$\hat{\boldsymbol{x}}_k = \sum_{c=1}^{M} P\left(c \mid \boldsymbol{\Omega}^k\right) \hat{\boldsymbol{x}}_k^c$

(4) If $N_{eff}(c) < N_{thres}$, $c = 1, 2, \ldots, M$

resample with replacement N_c particles $\{\boldsymbol{x}_k^{(j)}, m_k^{(j)}\}; j = 1, \ldots, N_c$
from the set $\{\boldsymbol{x}_k^{(j)}, m_k^{(j)}\}; j = 1, \ldots, N_c$ according to the weights;
set $W_k^{(j)} = 1/N_c$, $j = 1, \ldots, N_c$

(5) <u>Set</u> $k \longleftarrow k + 1$ and go to step 2.

In step (3) the posterior class probabilities are computed after applying the Bayes rule, i.e. the posterior is equal to the product of the likelihood and the prior, divided by the evidence.

The mixture Kalman filter (MKF) [26,27] is another sequential Monte Carlo estimation technique which has been successfully applied to different problems in target tracking and digital communications (see e.g. [28,29]). It is essentially a bank of Kalman filters (KFs) run with the Monte Carlo sampling approach. The MKF is derived for state-space models in a special form, namely *conditional dynamic linear model* (CDLM), *conditional linear Gaussian model*, or *partially linear Gaussian model*:

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{F}_{\lambda_k}\boldsymbol{x}_{k-1} + \boldsymbol{G}_{\lambda_k}\boldsymbol{u}_{\lambda_k} + \boldsymbol{G}_{\lambda_k}\boldsymbol{w}_k, \\ \boldsymbol{z}_k = \boldsymbol{H}_{\lambda_k}\boldsymbol{x}_k + \boldsymbol{V}_{\lambda_k}\boldsymbol{v}_k, \end{cases} \tag{18}$$

where $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q})$ and $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}^c)$ are Gaussian distributed processes. $\{\lambda_k\}$ is a sequence of random indicator variables (called *latent*), independent of $\boldsymbol{w}_k$, $\boldsymbol{v}_k$ and the past state $\boldsymbol{x}_s$ and measurement $\boldsymbol{z}_s, s < k$. The term *conditional* justifies the characteristic of these models: they are linear for a given trajectory of the indicator λ_k. Then, the Monte Carlo sampling works in the space of *latent variables* instead of in the space of the state variables. The matrices $\boldsymbol{F}_{\lambda_k}$, $\boldsymbol{G}_{\lambda_k}$, $\boldsymbol{H}_{\lambda_k}$ and $\boldsymbol{V}_{\lambda_k}$ are known, assuming that λ_k is known. For simplicity, in the remaining part of the paper we are omitting the subscript λ_k from the matrices of (18). The indicator λ_k usually takes values from a preliminary known finite set. The MKF relies on the conditional Gaussian property and uses a marginalization operation in order to improve the efficiency of the sequential Monte Carlo estimation technique.

Let $\boldsymbol{\Lambda}_k = \{\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_k\}$ be the set of indicator variables up to time instant k. By recursively generating a set of properly weighted random samples $\{(\boldsymbol{\Lambda}_k^{(j)}, W_k^{(j)})\}_{j=1}^N$ to represent the pdf $p(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega}^k)$, the MKF approximates the state pdf $p(\boldsymbol{x}_k|\boldsymbol{\Omega}^k)$ by a random mixture of Gaussian distributions [27]

$$\sum_{j=1}^{N} W_k^{(j)} \mathcal{N}(\boldsymbol{\mu_k}^{(j)}, \boldsymbol{\Sigma_k}^{(j)}), \tag{19}$$

where $\boldsymbol{\mu_k}^{(j)} = \boldsymbol{\mu_k}(\boldsymbol{\Lambda}_k^{(j)})$ and $\boldsymbol{\Sigma_k}^{(j)} = \boldsymbol{\Sigma_k}(\boldsymbol{\Lambda}_k^{(j)})$ are obtained by a KF, designed with the system model (18). We denote by $KF_k^{(j)} = \{\boldsymbol{\mu_k}^{(j)}, \boldsymbol{\Sigma}_k^{(j)}\}$ the sufficient statistics that characterize the posterior mean and covariance matrix of the state $\boldsymbol{x_k}$, conditional on the set of accumulated observations $\boldsymbol{\Omega}^k$ and the indicator trajectory $\boldsymbol{\Lambda}_k^{(j)}$. Then based on the set of samples $\{(\boldsymbol{\Lambda}_{k-1}^{(j)}, KF_{k-1}^{(j)}, W_{k-1}^{(j)})\}_{j=1}^N$ at the previous time $(k-1)$, the MKF produces a respective set of samples $\{(\boldsymbol{\Lambda}_k^{(j)}, KF_k^{(j)}, W_k^{(j)})\}_{j=1}^N$ at the current time k. It is shown in [27] that the samples $\{(\boldsymbol{\Lambda}_k^{(j)}, KF_k^{(j)}, W_k^{(j)})\}_{j=1}^N$ are indeed properly weighted with respect to $p(\boldsymbol{\Lambda}_k|\boldsymbol{\Omega}^k)$, if the samples $\{(\boldsymbol{\Lambda}_{k-1}^{(j)}, KF_{k-1}^{(j)}, W_{k-1}^{(j)})\}_{j=1}^N$ are properly weighted at time $(k-1)$ with respect to $p(\boldsymbol{\Lambda}_{k-1}|\boldsymbol{\Omega}^{k-1})$. The MKF for JTC is described in Table 2.

Table 2: The Mixture Kalman Filter for Joint Tracking and Classification

(1) Initialization, $k = 0$
For class $c = 1, 2, \ldots, M$ set initial class probabilities $P\left(c \mid \mathbf{\Omega}^0\right) = P_0(c)$.
* For $j = 1, \ldots, N_c$,

sample $\lambda_0^{(j)} \sim \{P_0^c(\lambda)\}_{\lambda=1}^{s(c)}$, where $P_0^c(\lambda)$ are the initial indicator probabilities (for the target accelerations).

Set $KF_0^{(j)} = \{\boldsymbol{\mu}_0(\lambda_0^{(j)}), \boldsymbol{\Sigma}_0(\lambda_0^{(j)})\}$, where $\boldsymbol{\mu}_0(\lambda_0^{(j)}) = \hat{\boldsymbol{\mu}}_0$ and $\boldsymbol{\Sigma}_0(\lambda_0^{(j)}) = \boldsymbol{\Sigma}_0$ are the mean and covariance of the initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\mu}_0, \boldsymbol{\Sigma}_0)$.

Set the initial weights $W_0^{(j)} = 1/N_c$.

* end for j
End for class c
Set $k = 1$.

(2) For class $c = 1, 2, \ldots, M$ complete

* *For* $j = 1, \ldots, N_c$,

- For each $\lambda_k = i, \;\; i \in \mathbb{S}(c)$, compute
  - KF prediction step

$$
\begin{aligned}
&(\boldsymbol{\mu}_{k|k-1}^{(j)})^{(i)} = \boldsymbol{F}\boldsymbol{\mu}_{k-1|k-1}^{(j)} + \boldsymbol{G}\boldsymbol{u}(\lambda_k = i, c),\\
&(\boldsymbol{\Sigma}_{k|k-1}^{(j)})^{(i)} = \boldsymbol{F}\boldsymbol{\Sigma}_{k-1|k-1}^{(j)}\boldsymbol{F}' + \boldsymbol{G}\boldsymbol{Q}(\lambda_k = i, c)\boldsymbol{G}',\\
&(\boldsymbol{z}_{k|k-1}^{(j)})^{(i)} = \boldsymbol{H}(\boldsymbol{\mu}_{k|k-1}^{(j)})^{(i)},\\
&(\boldsymbol{S}_k^{(j)})^{(i)} = \boldsymbol{H}(\boldsymbol{\Sigma}_{k|k-1}^{(j)})^{(i)}\boldsymbol{H}' + \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}'.
\end{aligned}
$$

  Note that j is a particle index, while $i = 1, 2, \ldots, s(c)$ is an index of the Kalman filters with the different acceleration levels (14).

  - on receipt of a measurement $\boldsymbol{z}_k$ calculate the likelihood
    $\mathcal{L}_{k,i}^{(j)} = f(\boldsymbol{z}_k|\lambda_k = i, KF_{k-1}^{(j)})\; p(\lambda_k = i|\lambda_{k-1}^{(j)})$, where
    $f(\boldsymbol{z}_k|\lambda_k = i, KF_{k-1}^{(j)}) = \mathcal{N}(\; (\boldsymbol{z}_{k|k-1}^{(j)})^{(i)}, (\boldsymbol{S}_k^{(j)})^{(i)}\;)$, and
    $p(\lambda_k = i|\lambda_{k-1}^{(j)})$ is the prior transition probability of the indicator

- Sample $\lambda_k^{(j)}$ from a set $\mathbb{S}(c)$ with a probability, proportional to $\mathcal{L}_{k,i}^{(j)}, i \in \mathbb{S}(c)$; suppose that $\lambda_k^{(j)} = \ell$, where $\ell \in \mathbb{S}(c)$. Append $\lambda_k^{(j)}$ to $\boldsymbol{\Lambda}_{k-1}^{(j)}$ and obtain $\boldsymbol{\Lambda}_k^{(j)}$.

- perform the KF update step for $\lambda_k^{(j)} = \ell$:

$$\boldsymbol{K}_{k|k}^{(j)} = (\boldsymbol{\Sigma}_{k|k-1}^{(j)})^{(\ell)}(\boldsymbol{H})'[(\boldsymbol{S}_k^{(j)})^{(\ell)}]^{-1},$$

$$\boldsymbol{\mu}_{k|k}^{(j)} = (\boldsymbol{\mu}_{k|k-1}^{(j)})^{(\ell)} + \boldsymbol{K}_{k|k}^{(j)}[\boldsymbol{z}_k - (\boldsymbol{z}_{k|k-1}^{(j)})^{(\ell)}],$$
$$\boldsymbol{\Sigma}_{k|k}^{(j)} = (\boldsymbol{\Sigma}_{k|k-1}^{(j)})^{(\ell)} - \boldsymbol{K}_{k|k}^{(j)}(\boldsymbol{S}_k^{(j)})^{(\ell)}\boldsymbol{K}_{k|k}^{(j)\prime},$$

- update the importance weights

$$W_k^{(j)} = W_{k-1}^{(j)} g_c\left(\hat{\mathrm{v}}_{k-1}^c\right) \sum_{i=1}^{s(c)} \mathcal{L}_{k,i}^{(j)}$$

* *end for* j

Set $L(c) = \sum_{j=1}^{N_c} W_k^{(j)}$;

Normalize the weights $W_k^{(j)} = W_k^{(j)} / \sum_{j=1}^{N_c} W_k^{(j)}$

Compute the updated state estimate, posterior mode probabilities and hard mode decision

$$\hat{\boldsymbol{x}}_k^c = \sum_{j=1}^{N_c} \boldsymbol{\mu}_{k|k}^{(j)} W_k^{(j)}, \quad P\left(\lambda_k^c = i\right) = \sum_{j=1}^{N_c} 1(\lambda_k^{(j)} = i) W_k^{(j)}, \quad i \in \mathbb{S}(c)$$
$$\hat{\lambda}_k^c = \arg\max_{i \in \mathbb{S}(c)} P\left(\lambda_k^c = i\right)$$

Compute the effective sample size: $N_{eff}(c) = 1/\sum_{j=1}^{N_c} \left(W_k^{(j)}\right)^2$

<u>End</u> for class c

(3) <u>Output:</u> Compute the posterior class probabilities and combined output estimate (such as in the PF)

$$P\left(c \mid \boldsymbol{\Omega}^k\right) = \frac{L(c)P\left(c|\boldsymbol{\Omega}^{k-1}\right)}{\sum_{i=1}^{M} L(c_i)P\left(c_i|\boldsymbol{\Omega}^{k-1}\right)}, \quad c = 1, 2, \ldots, M$$

$$\hat{\boldsymbol{x}}_k = \sum_{c=1}^{M} P\left(c \mid \boldsymbol{\Omega}^k\right) \hat{\boldsymbol{x}}_k^c$$

(4) <u>If</u> $N_{eff}(c) < N_{thres},\ c = 1, 2, \ldots, M$

resample with replacement N_c particles for each class:

$\lambda_k^{(j)}$, $\boldsymbol{\mu}_{k|k}^{(j)}$, $\boldsymbol{\Sigma}_{k|k}^{(j)}$; $j = 1, \ldots, N_c$ according to the weights;

Set $W_k^{(j)} = 1/N_c,\ j = 1, \ldots, N_c$

(5) <u>Set</u> $k \longleftarrow k + 1$ and go to step 2.

Two MKFs are run in parallel, each of them according to the hypothesis: respectively commercial or military aircraft. In our JTC problem the *indicator* variable λ corresponds to the *mode* variable m from the previous sections. It takes values from a finite discrete set $\mathbb{S}(c) \triangleq \{1,\ 2,\ \ldots,\ s(c)\}$ and evolves according to a Markov chain with transition probabilities (3).

At every time instant k, for each particle (j), $j = 1, \ldots, N_c$, the MKF scheme runs $s(c)$ KF prediction steps, according to each $\lambda \in \mathbb{S}(c)$. The likelihood functions of the predicted states are calculated based on the received measurement $\boldsymbol{z}_k$.

They form a trial sampling distribution, according to which the new λ_k is selected. The MKF explores the predicted space in order to select the most likely value of λ. Then the KF update step is accomplished only for the selected λ_k. In the particle filtering approach this procedure is realized in some chaotic (random) sense. This fact, together with the lack of the base state sampling, make the MK filtering more precise and more computationally efficient. However, the MKF application is limited to the CDLM. For that purpose the measurement equation (2) is linearized through a coordinate conversion. It is assumed that the matrices $\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{V}$ have the same structure for two classes.

Notice that in both MM PF and MKF the updated state estimates and posterior mode probabilities are calculated before the resampling step, because resampling brings extra variation to the current samples ([30], [26], pp. 103).

3.6 Delayed-Pilot Mixture Kalman Filter

Since both types of aircraft can perform slow maneuvers, the recognition can only be achieved during the aircraft maneuvers with high speed and acceleration. In some cases it might take a rather long tracking time to distinguish the types. The estimated posterior mode probabilities can be used for a classification in complicated ambiguous situations. The reliability of the mode information can be further improved by using of a delayed mode estimation scheme. Since target maneuvers are usually modelled as a *highly correlated* acceleration process, the observations in the near future can provide a valuable information about the current mode state.

Recently, delayed estimation methods have been proposed for the purposes of mobile communications [29,21]. The idea of the delayed estimation consists in the following. Based on the posterior distribution $p(\boldsymbol{x}_k, \lambda_k \mid c, \boldsymbol{\Omega}^k)$, an instantaneous inference is made on the state $\boldsymbol{x}_k$ and the indicator variable λ_k. If we use the next measurements $(\boldsymbol{\omega}_k, \ldots, \boldsymbol{\omega}_{k+\triangle}, \triangle \geq 0)$ with the distribution $p(\boldsymbol{x}_k, \lambda_k \mid \boldsymbol{\Omega}^{k+\triangle})$, the *current* state and mode accuracy can be improved at the cost of delayed estimation at time $k+\triangle$. Consequently, the aim of delayed estimation is to generate samples and weights $\left\{\lambda_k^{(j)}, W_k^{(j)}\right\}_{j=1}^{N_c}$ from the distribution $p(\boldsymbol{\Lambda}_k \mid \boldsymbol{\Omega}^{k+\triangle})$ at time $k+\triangle$. However, the algorithms for delayed estimation are more complicated. Several schemes for delayed estimation are suggested in [29,21] in the context of mixture Kalman filtering: delayed-weight, delayed-sample sampling, delayed-pilot sampling. A highly effective delayed-sample sampling method is developed and studied in [29]. It realizes a full exploration of the space of future states of $\triangle$ steps, and generates samples of both the current state and the weights. Due to the need of marginalization over the future states, its computational complexity is exponential in terms of the delay.

A less complicated delayed-pilot sampling method is proposed in [21]. Instead of

exploring the entire space of future states, the delayed-pilot sampling generates a number of random pilot streams, each of which indicates what would happen in the future if the current state takes a particular value. The sampling distribution of the current state is then determined by the incremental importance weight associated with each pilot stream. For each class c, at each time step k, the delayed-pilot algorithm generates $s(c)$ random pilot streams of the future states of $\triangle$ steps. Each pilot stream starts with one of the possible values $\lambda_i,\ i \in \mathbb{S}$ and implements $\triangle$ MKF steps. An incremental importance weight of this pilot stream is calculated and assigned to λ_i. The aim is to utilize this information (that can forecast what would happen in the future) for generating better samples of λ. A sample of the current state is drawn proportionally to the incremental importance weight. This algorithm partially explores the space of future states. This fact explains why the pilot approximation introduces a bias, which is additionally corrected by the inverse of the pilot weights. A scheme of the delayed-pilot MKF [21], adapted to the JTC is given in Table 3. The algorithm starts with a conventional MKF, described in the previous section. The inclusion of the delayed estimation is done after the time interval of 15 scans. Only the steps that are different are described for conciseness (the other steps : initialization, output and resampling are the same as given in Table 2).

Suppose at time $(k-1)$, a set of properly weighted samples $\left\{\left(\mathbf{\Lambda}_{k-1}^{(j)}, KF_{k-1}^{(j)}, W_{k-1}^{(j)}\right)\right\}_{j=1}^{N_c}$ with respect to $p\left(\mathbf{\Lambda}_{k-1} \mid c, \mathbf{\Omega}^{k-1}\right)$ are available. Then, as the new data $\boldsymbol{\omega}_k, \ldots, \boldsymbol{\omega}_{k+\triangle}$ arrive, the following steps are implemented to obtain $\left\{\left(\mathbf{\Lambda}_{k}^{(j)}, KF_{k}^{(j)}, W_{k}^{(j)}\right)\right\}_{j=1}^{N_c}$ for each class c:

Table 3: The delayed-pilot MKF for Joint Tracking and Classification

For $j = 1, \ldots, N_c$,

- for each $\lambda_k = i,\ i = 1, 2, \ldots, s(c)$ run a Kalman filter
  $KF_{k-1}^{(j)} \rightarrow \boldsymbol{z}_k \rightarrow KF_{k,i}^{(j)} \triangleq \mathcal{N}\left[\mu_k\left(\mathbf{\Lambda}_{k-1}^{(j)}, \lambda_k = i\right), \mathbf{\Sigma}_k\left(\mathbf{\Lambda}_{k-1}^{(j)}, \lambda_k = i\right)\right]$
  compute $\mathcal{L}_{k,i}^{(j)} = f(\boldsymbol{z}_k|\mathbf{\Lambda}_{k-1}^{(j)}, \lambda_k = i, KF_{k-1}^{(j)})p(\lambda_k = i|\mathbf{\Lambda}_{k-1}^{(j)})$,

- for each $\lambda = i$ do the following:
  for $s = k+1, \ldots, k+\triangle$, repeat
  * Let $\mathbf{\Lambda}_{s-1}^{(i,j)} = \left[\mathbf{\Lambda}_{k-1}^{(j)}, \lambda_k = i, \lambda_{k+1} = \lambda_{k+1}^{(i,j)}, \ldots, \lambda_{s-1} = \lambda_{s-1}^{(i,j)}\right]$
    for each $\lambda_s = q,\ q \in \mathbb{S}(c)$, run a Kalman filter
    $KF_{s-1,i}^{(j)} \rightarrow \boldsymbol{z}_s \rightarrow KF_{s,i}^{(q,j)} \triangleq \mathcal{N}\left[\mu_s\left(\mathbf{\Lambda}_{s-1}^{(i,j)}, \lambda_s = q\right), \mathbf{\Sigma}_s\left(\mathbf{\Lambda}_{s-1}^{(i,j)}, \lambda_s = q\right)\right]$
  * compute the sampling density
    $\mathcal{L}_{s,i}^{(q,j)} = f(\boldsymbol{z}_s|\mathbf{\Lambda}_{s-1}^{(i,j)}, \lambda_s = q, KF_{s-1,i}^{(j)})p(\lambda_s = q|\mathbf{\Lambda}_{s-1}^{(i,j)})$

and draw a sample $\lambda_s^{(i,j)}$ according to the sampling density;

if $\lambda_s^{(i,j)} = \ell$, then set $KF_{s,i}^{(j)} = KF_{s,i}^{(\ell,j)}$

* compute the incremental importance weight

$\mathcal{L}_{s,i}^{(j)} = \sum_{q=1}^{s(c)} \mathcal{L}_{s,i}^{(q,j)}$

end for s

$$\rho_{k,i}^{(j)} = \mathcal{L}_{k,i}^{(j)} \prod_{s=k+1}^{k+\triangle} \mathcal{L}_{s,i}^{(j)} \ ,$$

end for each $\lambda_k = i$,

$\tilde{\rho}_{k,i}^{(j)} = \rho_{k,i}^{(j)} / \sum_{r=1}^{s(c)} \rho_{k,r}^{(j)},\ i = 1, 2, \ldots, s(c)$,

use $\tilde{\rho}_{k,i}^{(j)},\ i = 1, 2, \ldots, s(c)$ as the sampling distribution, draw a sample $\lambda_k^{(j)}$and obtain $\mathbf{\Lambda}_k^{(j)} = \left[\mathbf{\Lambda}_{k-1}^{(j)}, \lambda_k^{(j)}\right]$,

if $\lambda_k^{(j)} = \ell$, set $KF_k^{(j)} = KF_{k,\ell}^{(j)}$

- Compute the importance weights:

  if $\lambda_k^{(j)} = \ell$, $\quad W_k^{(j)} = W_{k-1}^{(j)} \frac{\mathcal{L}_{k,\ell}^{(j)}}{\tilde{\rho}_{k,\ell}^{(j)}}\, g_c\left(\hat{\mathrm{v}}_{k-1}^c\right)$,

  calculate the auxiliary weight $\quad \tilde{W}_k^{(j)} = W_k^{(j)} \prod_{s=k+1}^{k+\triangle} \mathcal{L}_{s,\ell}^{(j)}$,

end of j

use the auxiliary weight to compute the posterior mode probabilities,

- Output: compute updated state estimate, posterior class probabilities and combined output estimate according to the MKF scheme in Table 2.
- Resampling if the effective sample size is below a certain threshold.
- Set $k \longleftarrow k+1$ and go to the beginning of the scheme.

Comment: The delayed-pilot MKF algorithm for JTC generates at time $(k+\triangle)$ a weighted sample set $\left(\mathbf{\Lambda}_k^{(j)}, W_k^{(j)}\right)_{j=1}^{N_c}$, which is not properly weighted with respect to $p(\mathbf{\Lambda}_k \mid \mathbf{\Omega}_k^{k+\triangle})$. A better inference on λ_k at time $(k+\triangle)$ is done by the calculation of auxiliary weights [21]. The auxiliary weights are used both for inference on λ_k and in the resampling step. The investigation of this algorithm for JTC shows, that the scheme slightly deteriorates the posterior class probabilities in comparison with the MKF.

4 Simulation results

The performance of the implemented filters for JTC is evaluated by simulations over representative trajectories (shown on Figures 2 and 11) together with the radar location, indicated by a triangle. The target motion is generated without a process noise. The target trajectories were generated with a model ([31] Eqs. (57)–(60)) differing from the models used in the filters. The MM particle filter and the MKF using both speed and acceleration constraints with likelihood computed according to (17) are compared to filters without speed constraints, which likelihood is equal to $\lambda_{\{\boldsymbol{x}_k, m_k, c\}} = f_{\boldsymbol{x}_k}(\boldsymbol{z}_k | \boldsymbol{x}_k, m_k, c)$. The performance of the designed delayed-pilot sampling algorithm for JTC is demonstrated on a specially selected target scenario (Fig. 22).

Measures of performance. Root-Mean Squared Errors (RMSEs) [32]: on position (both coordinates combined) and speed (magnitude of the velocity vector), *average probability of correct class identification* and *average time per update* are used to evaluate the filters performance. The results presented below are based on 100 Monte Carlo runs. The PF has $N_c = 3000$ particles per class, the MKF has $N_c = 300$ particles per class, and the resampling threshold is $N_{thresh} = N_c/10$. The prior class probabilities are chosen to be equal: $P_0(1) = P_0(2) = 0.5$. The parameters of the base state vector initial distribution $\boldsymbol{x}_0 \sim \mathcal{N}(m_0, \boldsymbol{P}_0)$ in the PF are selected as follows: $\boldsymbol{P}_0 = diag(150^2\ [m],\ 20.0^2\ [m/s],\ 150^2\ [m],\ 20.0^2\ [m/s])$; m_0 contains the exact initial target states. The MKF initial parameters (mean and covariance $\hat{\boldsymbol{\mu}}_0$ $\boldsymbol{\Sigma}_0$ of the initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_0, \boldsymbol{\Sigma}_0)$) are obtained by a two-point differencing technique [23] (p. 253), and $\boldsymbol{V} = \boldsymbol{I}$. The sampling period is $T = 5\ [s]$.

Test trajectory 1. The simulated target path is shown in Figure 2. The target accomplishes two turn maneuvers with normal accelerations $2g\ [m/s^2]$ and $1g\ [m/s^2]$, respectively with a constant speed of v $= 200\,[m/s]$. Then a maneuver is performed with a longitudinal acceleration of $2g\ [m/s^2]$ in the frame of 3 scans. The speed increases up to $500\,[m/s]$. Practically, these maneuvers are typical for targets of the first class. If the speed constraints are not imposed, the target is classified as belonging to the first class as it can be seen from Figure 7. But it actually belongs to the second class which is correctly distinguished by the speed likelihoods (Figure 9) when the target speed exceeds the value of $300\,[m/s]$ (during the maneuver with longitudinal acceleration). After this maneuver, one observes switching in the class probabilities (Figure 8), which shows the importance of the speed information for classifying the target in the proper class. The class probabilities for both filters are almost identical and for this reason only the class probabilities of the PF are presented (Figures 7 and 8). Figure 10 shows the mode probabilities of the MKF for class 1. The speed RMSEs are nearly identical for the two filters as well (Figures 5 and 6), whilst the PF position RMSE is larger than those of MKF (Figures 3 and 4). The RMSEs shown are for each separate class, and the combined RMSE obtained after averaging with the class probabilities, similarly to (12).

Test trajectory 2. The target performs four turn maneuvers (Figure 11) with intensity $1g,\ 2g,\ -5g,\ 2g$. The third $5g$ turn provides an insufficient true class information, since the maneuver is of short duration, and the next $2g$ turn leads to a misclassification in the PF and MKF without speed constraints (Figure 16). The target speed of $260\ [m/s]$ provides better conditions for the probability, that the target belongs to class 2, according to the speed constraints. The estimated speed probabilities assist in the proper class identification, as we can see in Figure 17. The normalized speed likelihoods $g_c(\hat{\mathrm{v}}^c),\ c = 1, 2$, obtained by the MM particle filter and MKF are quite similar. For these reasons we present only the results from the MKF (Figure 18). According to the results from the RMSEs (Figures 12, 13, 14, 15) the peak dynamic errors of the PF are considerably larger than the respective MKF results.

The chosen target model (13) in combination with the sequential Monte Carlo technique provides an easy way of imposing acceleration constraints on the target dynamics. Air targets usually perform turn maneuvers with varying accelerations along x and y coordinates. These varying accelerations consecutively make active different models from the designed multiple model configuration, since the models have fixed accelerations along x and y directions. During the maneuver different models may have similar probabilities which makes it difficult to infer which is the most probable between them. The approach proposed here clearly distinguishes different motion segments, as can be seen from the plots of the posterior model probabilities, Figure 10 for scenario 1, and Figure 19 for scenario 2.

Figure 20 shows the clouds of the PF particles for both classes and the true target position indicated by a square. The particles corresponding to class 2 have higher likelihoods: conclusions can definitely be drawn that the target is of class 2. Figure 21 represents the MKF y-position histogram at time instant $k = 56$ (at the beginning of the $5g$ turn). The particles corresponding to class 2 are in proximity to the true target position. They quickly adapt to the changeable target dynamics.

Test trajectory 3. The trajectory is similar in shape to the previous scenario 2. The target performs four turn maneuvers with normal accelerations $-1g,\ 2g,\ -5g,\ 4g$, displayed in Figure 22(b). In scenario 2, the target speed of $260\ [m/s]$ provided better conditions for the probability that the target belongs to class 2 according to the speed constraints. Now the target speed is $245\ [m/s]$ and the speed information is insufficient to distinguish the classes. The short-durable $-5g$ turn is followed by a $4g$ turn, which is located between the acceleration constraints, bounding the classes - $2g$ and $5g$. The situation is vague from the point of view of the classification logic. It is difficult for the MKF to solve the classification task - within 100 Monte Carlo runs, there are more than 50 realizations with wrong classification. The posterior mode probabilities can help to choose between : making the decision on the class *now* or *postponing* it to the future time. The delayed-pilot sampling algorithm provides more reliable mode information. It can be seen from Figures 23 and 24. The acceleration of $-50\ [m/s^2]$ along the y-axis is processed consecutively

by the models 3 and 5 of class two filter. The $-5g$ turn is short-durable (2 scans) and the effect of the delayed estimation is comparably small. That is why the hard mode decision probabilities, obtained by the delayed-pilot MKF are a little better than the respective values of the MKF (see Figure 23(b)). The next maneuver of a $4g$ turn is longer and lasts 4 scans. The acceleration of 40 $[m/s^2]$ along the y-axis is processed consecutively by models 2 and 4 of the filter for class two. The $4g$ turn is identified by the delayed-pilot MKF with a probability of 0.88 (Figure 24(b)), while the probability of the conventional MKF is 0.35. The posterior probabilities of the last two maneuvers, provided by the delayed-pilot MKF give a consistent information for putting off the classification decision. The experiments with the delayed estimation are realized with a time delay of two steps ($\triangle = 2$).

The following inferences can be drawn from the experiments. The results obtained by the MKF approach for solving the JTC problem confirm the theoretical inferences [27], that the MKF can provide better estimation accuracy compared to the particle filter. The lower peak dynamic errors of the MKF are particularly useful for the classification task. This enables tracking and classifying objects of quite different types, with rather distinct dynamic parameters. This capability ensures some robustness of the algorithm, because the designed type parameters are not always precisely known.

The MM PF and MKF computational complexity allow for an on-line implementation. An advantage of the MKF is its reduced complexity compared to the MM PF. The MKF was investigated in the cases of three sample sizes: $N_c = 300, 200$, and 100. The relative computational time of the PF versus the MKF is presented in Table 4. The computational time for a PF with $N_c = 3000$ particles is considered as a reference time, which is compared to the MKF computational time. The experiments are performed on PC computer with AMD Athlon processor 2 GHz. Both algorithms permit parallelization at least of some parts : the MM filters corresponding to each class can definitely be run in parallel. The delayed-pilot MKF is realized with a sample size $N_c = 200$ and a delay of two steps ($\triangle = 2$). It is 3 times more computational time consuming compared to the MKF.

Table 4. Relative computational time

N_c	3000	300	200	100
PF comp. time	1			
MKF comp. time		0.6	0.38	0.185
Delayed-pilot MKF			2	

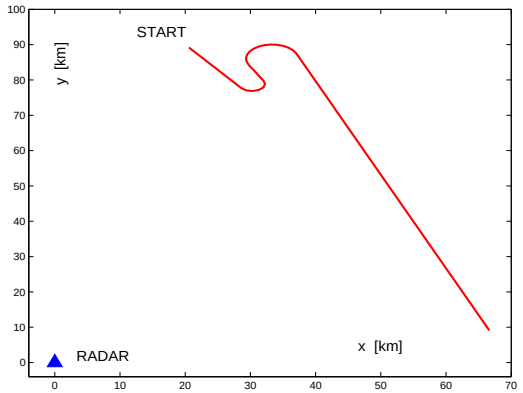

Fig. 2. Test trajectory 1

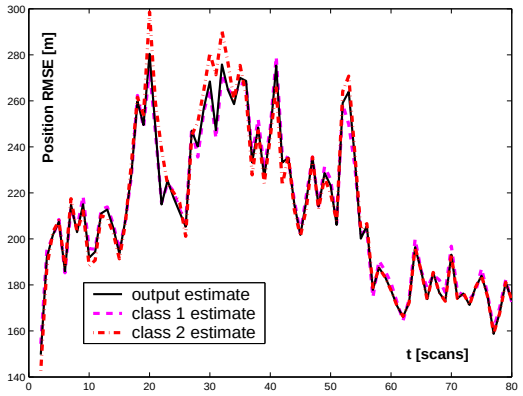

Fig. 3. PF position RMSE [m] (traj. 1)

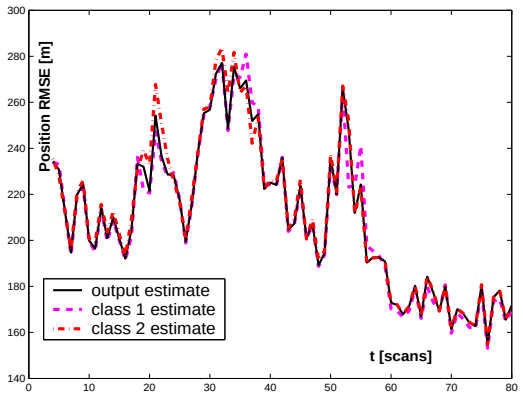

Fig. 4. MKF position RMSE [m] (traj. 1)

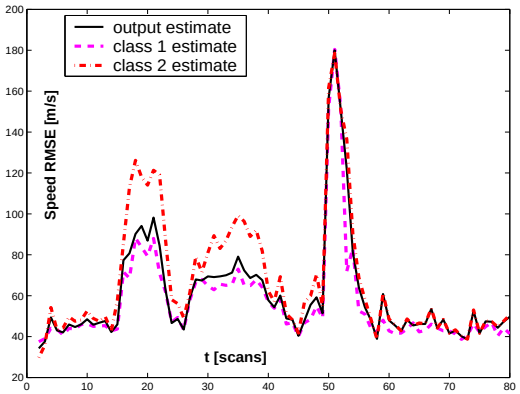

Fig. 5. PF Speed RMSE [m/s] (traj. 1)

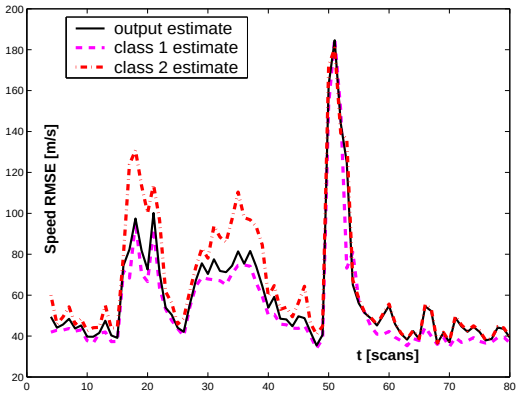

Fig. 6. MKF Speed RMSE [m/s] (traj. 1)

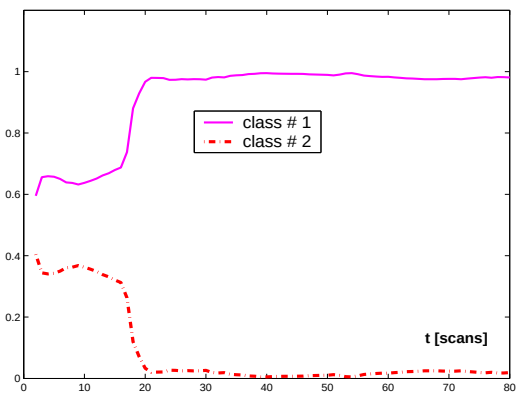

Fig. 7. PF class probabilities calculated without taking into account speed constraints (traj. 1)

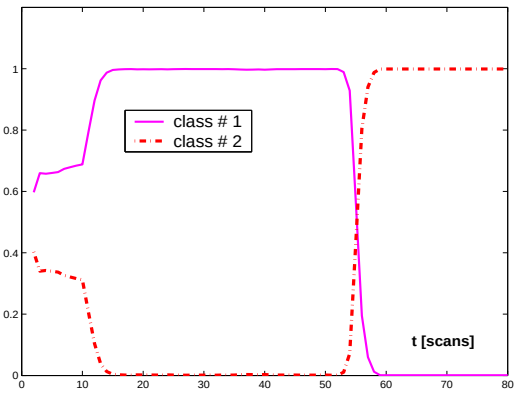

Fig. 8. PF class probabilities calculated using both speed and acceleration constraints (traj. 1)

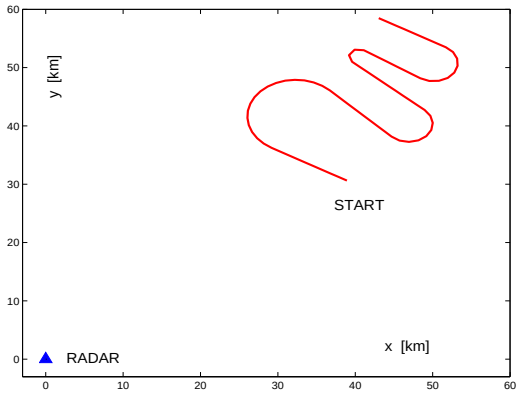

Fig. 11. Test trajectory 2

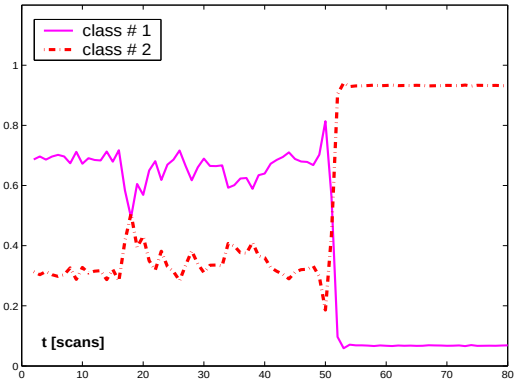

Fig. 9. PF normalized speed likelihoods (traj.1)

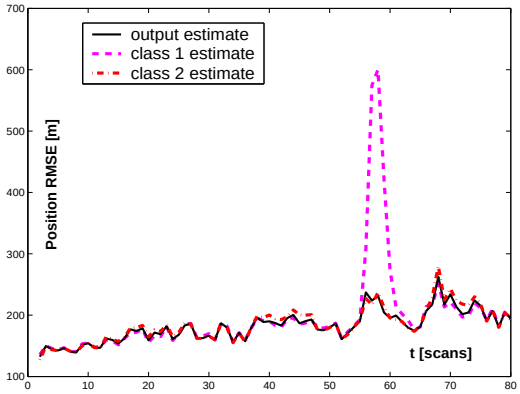

Fig. 12. PF position RMSE [m] (traj. 2)

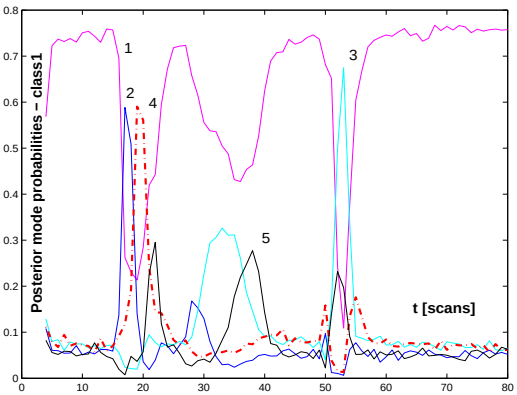

Fig. 10. MKF posterior mode probabilities - class 1 (traj. 1)

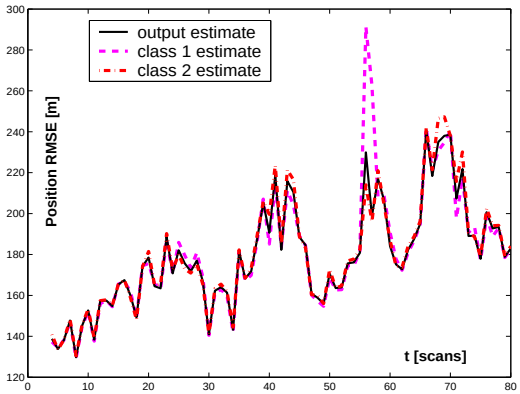

Fig. 13. MKF position RMSE [m] (traj. 2)

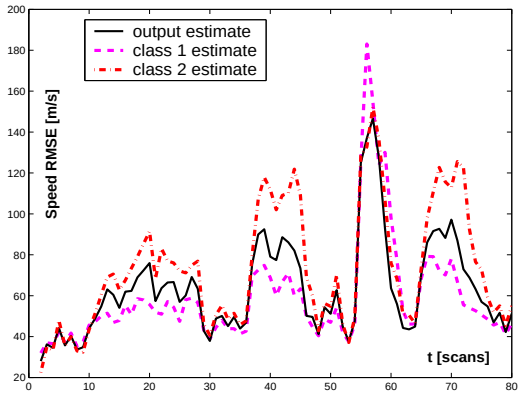

Fig. 14. PF speed RMSE [m/s] (traj. 2)

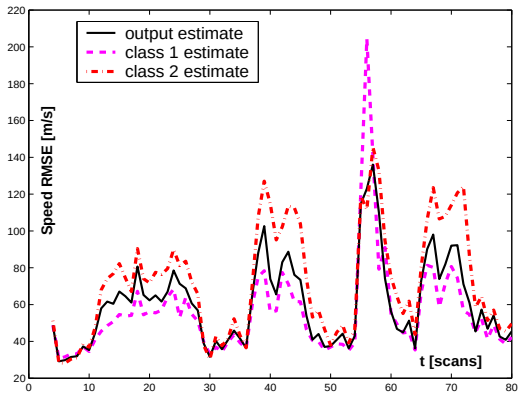

Fig. 15. MKF speed RMSE [m/s] (traj. 2)

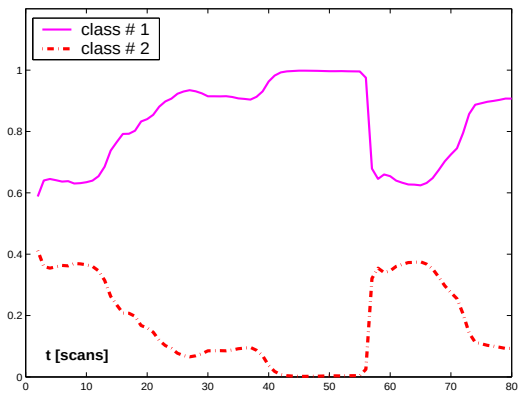

Fig. 16. Class probabilities (without speed constraints) (traj. 2)

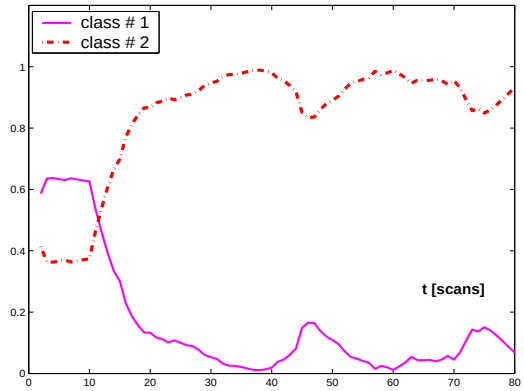

Fig. 17. Class probabilities (speed constraints) (traj. 2)

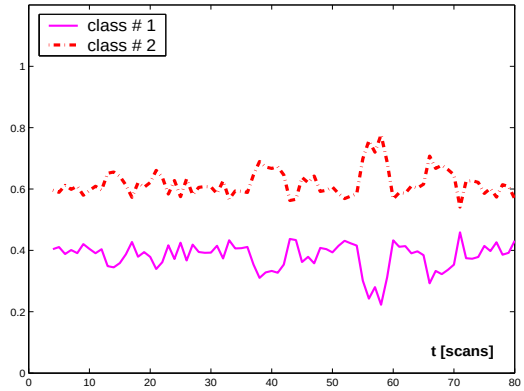

Fig. 18. Normalized speed likelihoods (traj. 2)

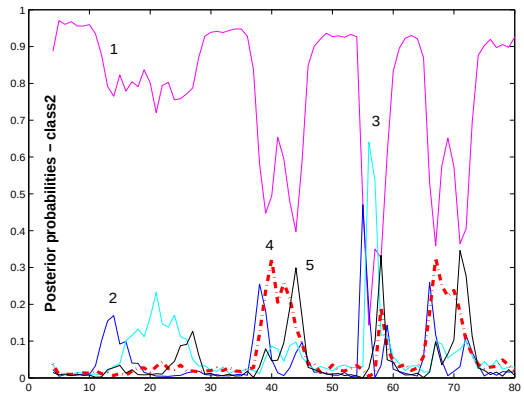

Fig. 19. MKF posterior mode probabilities - class 2 (traj. 2)

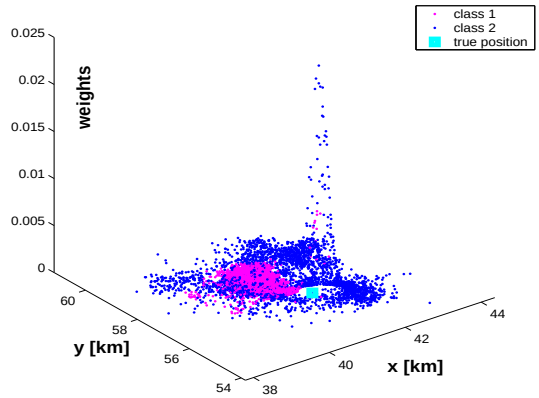

Fig. 20. PF - cloud of particles and associated weights at scan 57 - 5g turn (traj. 2)

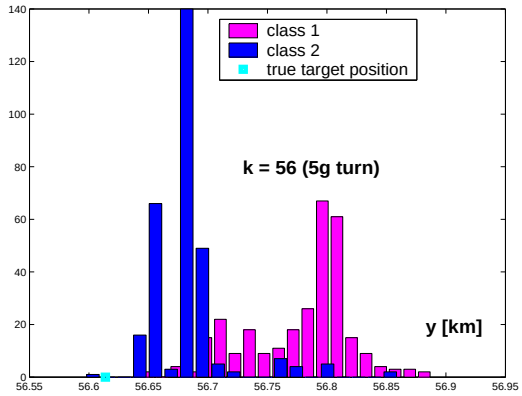

Fig. 21. MKF y-position histogram (traj. 2)

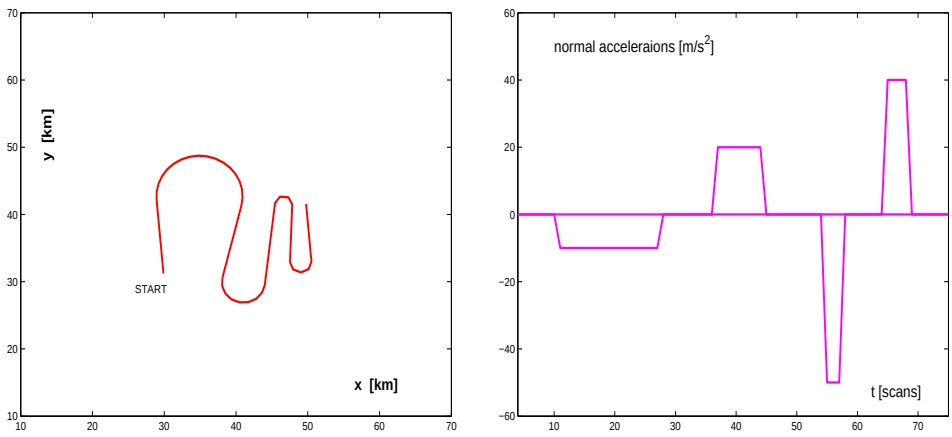

Fig. 22. Test trajectory 3

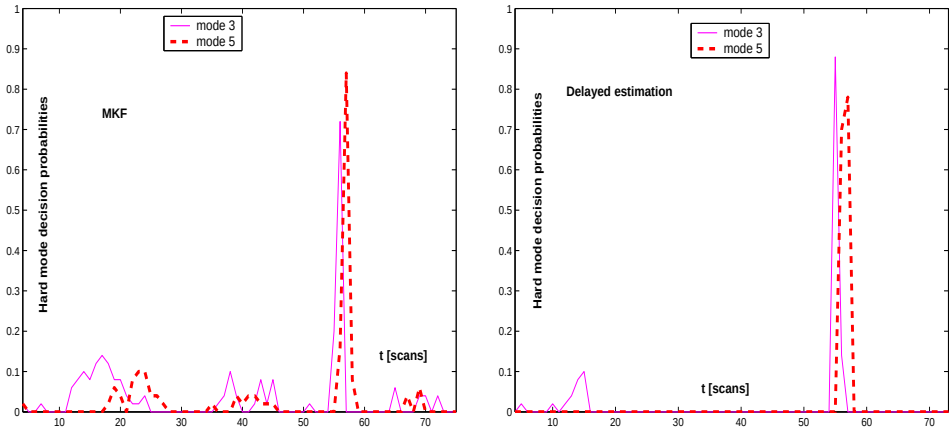

Fig. 23. Hard decision for $5g$ maneuver (modes 3 and 5) for class 2

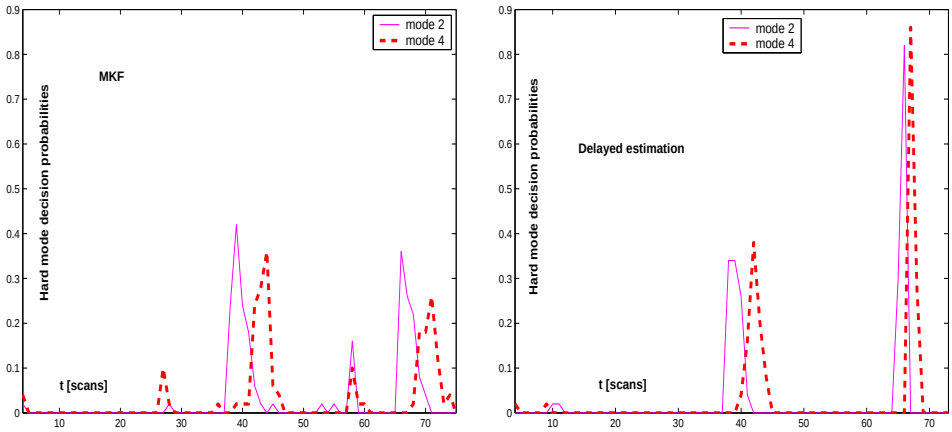

Fig. 24. Hard decision for $4g$ maneuver (modes 2 and 4) for class 2

5 Conclusions

In this paper we propose a multiple model particle filter, a mixture Kalman filter and a delayed-pilot mixture Kalman filter for the purposes of joint *maneuvering* target tracking and classification. It was shown that distinct constraints, enforced by the changeable target behavior can be easily incorporated into the Monte Carlo framework. Two air target classes are considered: *commercial* and *military* aircraft. The classification task is accomplished by processing kinematic information only, no class (feature) measurements are used. A bank of two *multiple model* class-dependent particle filters is designed and implemented in the presence of speed and acceleration constraints. The acceleration constraints for each class are imposed by using different control inputs in the target model. Speed constraints are enforced by constructing class-dependent speed likelihood functions that assist the classification process. The filters performance is analyzed by simulation over typical target trajectories in a plane. The results show reliable tracking and correct target

type classification. A generalization of the algorithms' application to the three-dimensional case is straightforward. In complicated target scenarios, the posterior mode probabilities can help in choosing from two possibilities: to make the decision on the class *now* or to *postpone* it to the future time. The posterior mode probabilities, provided by the delayed-pilot MKF offer a more reliable information for decision making.

Although the effectiveness of the developed PF and MKF algorithms is demonstrated over measurements from a single sensor their application over a network of sensors is possible and it is an open issue for future research.

Acknowledgements. The authors are thankful the reviewers which valuable comments helped for improving considerably the paper. This research is supported by the Bulgarian Foundation for Scientific Investigations under grants I-1202/02 and I-1205/02 and in part by the UK MOD Data and Information Fusion Defence Technology Center.

References

[1] S. Challa and G. Pulford, "Joint target tracking and classification using radar and ESM sensors," *IEEE Trans. on Aerosp. and Electr. Syst.*, vol. 37, no. 3, pp. 1039–1055, 2001.

[2] B. Ristic, N.Gordon, and A. Bessell, "On target classification using kinematic data," *Information Fusion*, vol. 5, pp. 15–21, 2004.

[3] A. Farina, P. Lombardo, and M. Marsella, "Joint tracking and identification algorithms for multisensor data," *IEE Proc.-Radar Sonar, Navig.*, vol. 149, no. 6, pp. 271–280, 2002.

[4] N. Gordon, S. Maskell, and T. Kirubarajan, "Efficient particle filters for joint tracking and classification," in *Proc. SPIE Signal and Data Processing of Small Targets*, vol. 4728, Orlando, USA, April 1–5 2002.

[5] H. Leung and J. Wu, "Bayesian and Dempster-Shafer target identification for radar surveillance," *IEEE Trans. Aerosp. and Electr. Syst.*, vol. 36, no. 2, pp. 432–447, 2000.

[6] P. Smets and B. Ristic, "Kalman filter and joint tracking and classification in the TBM framework," in *Proc. of the Seventh Intl. Conf. on Information Fusion*, Stockholm, Sweden, 2004, pp. 46–53.

[7] H. J. Zimmerman, *Fuzzy Set theory – and its applications*. Kluwer, Dordrecht, the Netherlands, 1996.

[8] E. Blasch and C. Yang, "Ten methods to fuse GMTI and HRRR measurements for joint tracking and identification," in *Proc. of the 7th Intl. Conf. on Multisensor Information Fusion*, Stockholm, Sweden, 2004, pp. 1006–1013.

[9] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Proc.*, vol. 50, no. 2, pp. 174–188, 2002.

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

[11] S. Challa and N. Bergman, "Target tracking incorporating flight envelope information," in *Proc. of the Third Int. Conf. Inform. Fusion*, Paris, France, July 2000.

[12] D. Salmond, D. Fisher, and N. Gordon, "Tracking and identification for closely spaced objects in clutter," in *Proc. of the Europ. Contr. Conf.*, Brussels, Belgium, 1997.

[13] M. Malick, S. Maskell, T. Kirubarajan, and N. Gordon, "Littoral tracking using particle filter," in *Proc. of the Fifth Int. Conf. Information Fusion*, USA, July 2002.

[14] S. Herman and P. Moulin, "A particle filtering appropach to FM-band passive radar tracking and automatic target recognition," in *Proc. of the IEEE Aerospace Conf.*, Big Sky, Montana, March 2002.

[15] V. Jilkov, D. Angelova, and T. Semerdjiev, "Filtering of hybrid systems: Bootstrap versus IMM," in *Proc. of the European Conf. on Circuit Theory and Design*, Budapest, Hungary, 1997, pp. 2: 873–858.

[16] S. McGinnity and G. Irwin, "Multiple model bootstrap filter for maneuvring target tracking," *IEEE Trans. on Aerospace Systems*, vol. 36, no. 3, pp. 1006–1012, Dec. 2000.

[17] Y. Boers and J. Driessen, "Interacting multiple model particle filter," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 149, no. 5, pp. 344–349, Oct. 2003.

[18] H. Driessen and Y. Boers, "An efficient particle filter for jump markov nonlinear systems," in *Proc. of the IEE Seminar Target Tracking: Algorithms and Applications*, Sussez, UK, March 23-24 2004.

[19] D. Angelova, L. Mihaylova, and T. Semerdjiev, "Monte Carlo algorithm for maneuvering target tracking and classification," in *Lecture Notes in Computer Science. ICCS Proc. Part IV, 4th Intl. Conf.*, M. Bubak and G. Dick van Albada and P. Sloot and J. Dongarra, Ed. Krakow, Poland: Springer, 2004, vol. LNCS 3039, pp. 531–539.

[20] D. Angelova and L. Mihaylova, "Sequential Monte Carlo algorithms for joint target tracking and classification using kinematic radar information," in *Proc. of the Seventh Intl. Conf. on Information Fusion*, Stockholm, Sweden, 2004, pp. 709–716.

[21] X. Wang, R. Chen, and D. Guo, "Delayed-pilot sampling for mixture kalman filter with application in fading channels," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 241–254, Feb. 2003.

[22] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. on Signal Proc.*, vol. 49, no. 3, pp. 613–624, 2001.

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

[24] A. Averbach, S. Itzikowitz, and T. Kapon, "Radar target tracking – Viterbi versus IMM," *IEEE Trans. on Aerosp. and Electr. Syst.*, vol. 27, no. 3, pp. 550–563, 1991.

[25] A. Tchamova, T. Semerdjiev, and J. Dezert, “Estimation of target behaviour tendencies using Dezert-Smarandache theory,” in *Proc. of the Sixth Intl. Conf. Inform. Fusion*, Australia, July 2003, pp. 1349–1356.

[26] J. S. 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. [Online]. Available: citeseer.nj.nec.com/article/liu98sequential.html

[27] R. Chen and J. S. Liu, “Mixture Kalman filters,” *Journal of the Royal Statistical Society B*, no. 62, pp. 493–508, 2000.

[28] Z. Chen, “Bayesian filtering: From Kalman filters to particle filters, and beyond,” *Adaptive Syst. Lab., McMaster Univ., Hamilton, ON, Canada. [Online], http://soma.crl.mcmaster.ca/˜zhechen/homepage.htm*, 2003.

[29] R. Chen, X. Wang, and J. Liu, “Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering,” *IEEE Trans. on Inform. Theory*, vol. 46, no. 6, pp. 493–508, 2000.

[30] G. Casella, “Statistical inference and Monte Carlo algorithms,” *Test*, vol. 5, pp. 249–344, 1997.

[31] X. R. Li and V. Jilkov, “A survey of maneuveuvering target tracking. Part I: Dynamic models,” *IEEE Trans. on Aerospace and Electr. Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[32] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995.