jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2013, VOL. 59, NO. 4, PP. 369–374
Manuscript received July 2, 2013; revised October, 2013. DOI: 10.2478/eletel-2013-0045

Numerically Stable and Efficient Implementation of a Continuous-Discrete Multiple-Model Estimator

Mirosław Sankowski and Wojciech Buda

Abstract—**This paper deals with the problem of implementing adaptive radar tracking filters based on continuous-time models of target motion and on discrete-time models of measurement process. The particular difficulties addressed include: nonlinear and non-stationary target movement models with uncertain parameters, and low data rate due to a rotating radar antenna. The proposed tracking filter relies basically on the continuous-discrete variant of the extended Kalman filter (EKF), the probabilistic data association (PDA) technique and the interacting multiple-model (IMM) state estimation scheme. Numerical properties of the algorithm are discussed and a software implementation is developed using the open-source BLAS library. Several design concepts are combined to assure numerical stability, convergence and efficiency of the estimator.**

Keywords—**Radar tracking, continuous-time systems, nonlinear filtering, numerical complexity, implementation**

I. Introduction

THE essential task of the tracking filter is to process discrete-time (*d-t*) radar measurements of motion parameters of a target in order to reduce the measurement errors by means of time averaging, to estimate the object's velocity and acceleration and to predict its future positions. Widely accepted basis for tracking filters is the Kalman filtering (KF) theory [1], [2].

Radar tracking becomes extremely challenging in the case of low data rates combined with the problem of structural non-stationarity of a target model. The representative practical cases are manoeuvring aircrafts or ballistic missiles, which are characterised by mode transition from boost to coast phase. In both cases the technical problem corresponds to a sudden change in the target dynamics that occurs at an unknown instant.

The considerations presented in this work originate from the problem of radar tracking of tactical ballistic missiles, which is subject to severe constraints, including: radar coverage limitations, possible low signal-to-noise ratio, nonlinear and non-stationary movement models with uncertain parameters. Basic difficulties encountered in this problem include:

- continuous-time (*c-t*) dynamic models are nonlinear and not suitable for analytical discretization,
- sampling period is long, as compared to the target dynamics, hence simplifications neglecting the *c-t* nature of models cannot be accepted.

An interesting technique, often mentioned in the literature [3] but rather rarely used in tracking [4], is the continuous-discrete (*c-d*) filter, which relies on a *c-t* extended Kalman filter (EKF) for prediction stage [2] and on a *d-t* (E)KF for filtering stage [1]. This *c-d* technique is employed as the basis for state-estimation. In order to cope with the non-stationarity of the model, the *interacting multiple model* (IMM) adaptive approach [3], [5] was selected. Since no track crossings are expected, robustness of the estimator against false measurements is accomplished by using the *probabilistic data association* (PDA) technique [3], [6].

A substantial effort has been devoted to the problem of numerical implementation of the KF-based estimators. Classical approach to robust Kalman estimation is based on a square-root filtering [7]. Some representative references related to the selected IMM and PDA techniques are [8] and [9]. Implementation aspects of the *c-d* estimators were addressed by Jorgensen and Thomsen [10] who proposed an efficient algorithm for propagating state vector and covariance matrix of the *c-d* EKF. They applied a modified diagonal implicit Runge-Kutta method with an integration step size controller such that the computed solution satisfies the required accuracy of *c-t* prediction by using a minimum number of numerical integration steps. The efficiency of the algorithm was demonstrated for a large scale chemical process model (order of 25-200). Because the method presumes that the spectral density matrix of the system noise is time invariant, which is not the case here, and as the system considered is characterised by relatively low dimension, the method is assessed not adequate for this application. Another method for *c-d* EKF was proposed by [11], which employs a trapezoidal approximation of the state propagation and the Gauss-Legendre formula for covariance prediction. A suitable method for adaptively tuning the integration time interval was also proposed.

A combination of software engineering and applied mathematics must be used to implement a complex state estimation algorithm. The following presentation involves an algorithm block diagram and its C language pseudocode skeleton. According to this structure the analysis of numerical properties of the involved mathematical formulae is discussed and their implementation is outlined using BLAS, the open-source linear algebra library. For the illustration simplicity a two-model *c-d* IMM estimator is discussed, though similar approach can be followed for any configuration of models.

This paper is related to the previous, more theoretical oriented paper [12], which addresses in details mathematical formulation of target and sensor models and presents the simulation results for a sample scenario. This work extends the discussion published in a conference paper [13] and deals with numerical implementation aspects of an adaptive *c-d* tracking algorithm. The major contribution of this paper is a practical methodology presented as an estimator design

M. Sankowski is with Bumar Elektronika S.A., Hallera 233A, 80-502 Gdańsk, Poland (e-mail: miroslaw.sankowski@gmail.com).

W. Buda is with Intel Technology Poland Sp. z o.o., Słowackiego 173, 80-298 Gdańsk, Poland.

case study, which takes into consideration various aspects influencing its numerical properties. Two major objectives that shape the presented approach are: to assure estimation stability and convergence, and to minimise computer resource needs, which translates to the number of multiplications, additions and memory storage needs.

The organisation of the paper is as follows. In section II the underlining problem of nonlinear estimation of ballistic target is introduced along with the data flow diagram of multiple-model *c-d* estimator and the implementation assumptions. The estimator is analysed from the numerical aspect in section III and its implementation using BLAS library is described. In section IV the outcomes are summarised and conclusions are drawn.

II. Problem Formulation

The following mathematical notation will be used to characterise filter equations: α, β denote scalars, $\mathbf{x}, \mathbf{y}$ are column vectors, $\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{E}, \mathbf{G}, \mathbf{H}$ mean general matrices, $\mathbf{0}_{r\times c}$ denote a null matrix (r rows and c columns), $\mathbf{I}_{r\times r}$ refer to a unit matrix (r by r). Subscript T denotes a triangular, S a symmetric, while P a symmetric positive definite matrix.

A. Radar Model

The considered sensor is a 3D monopulse radar with a rotating antenna characterised by detection range exceeding several hundreds kilometres and measurement update interval corresponding to the antenna revolution time ART=5,...,12 [s]. It is assumed that the radar provides at discrete instants n unbiased measurements (plots) of object positions: slant range $\tilde{\rho}[n]$, azimuth $\tilde{\alpha}[n]$ and elevation $\tilde{\varepsilon}[n]$, which are disturbed by additive non-stationary Gaussian-distributed white noises with zero expectations and variances σ_ρ^2, σ_α^2 and σ_ε^2, respectively.

B. Target Model

Two phases (modal states) of a ballistic flight are recognised that correspond to hypotheses $\{H_j\}$,$j \in \{0,1\}$:

- boost phase (H_0), which lasts from the launch instant t_L till the time t_B of the booster cutoff,
- coast phase (H_1), from t_B till the impact time t_I, when the movement is a free ballistic flight,

which are represented by two dynamic models described by a suitable set of nonlinear stochastic differential equations [12]. The models rely basically on a geodetic model of the Earth and address several sources, which influence target movements, including: gravity, thrust, drag, Coriolis and centrifugal factors.

Moreover, it is assumed that transitions possible between the modal states are described by a Markov model defined by *a-priori* probabilities $p_{j,k}$, for $j,k = 0,1$, that the target will make transition from the modal state H_j to H_k. Models of target motion are represented in a state-space form by a *c-t* system equation and a *d-t* measurement equation

$$\dot{\boldsymbol{x}}_j(t) = \boldsymbol{f}_j\big(t, \boldsymbol{x}_j(t)\big) + \boldsymbol{w}_j(t) \tag{1}$$

$$\tilde{\boldsymbol{y}}[n] = \boldsymbol{C}_j \boldsymbol{x}_j[n] + \boldsymbol{r}[n] \tag{2}$$

for $j = 0,1$. In the above $\boldsymbol{x}_j(t)$ denotes a state vector, $\boldsymbol{f}_j()$ is a nonlinear function describing the deterministic part of the model, input $\boldsymbol{w}_j(t)$ is the *c-t* white Gaussian noise with zero mean and spectral density matrix $\boldsymbol{W}_j(t)$. The state vector

$$\boldsymbol{x}_j(t) = [\rho(t), \alpha(t), \varepsilon(t), v_\rho(t), v_\alpha(t), v_\varepsilon(t), x, y]^{\mathrm{T}} \tag{3}$$

where x and y describe the evolution of an uncertain parameter of the model, which must be estimated on-line.

The state vector and the model used differs from similar models of ballistic flight formulated in radar-principal coordinates [14] as they employ Cartesian line-of-sight velocity coordinates $v_\rho(t), v_\alpha(t), v_\varepsilon(t)$. This improves diagonal dominance and numerical conditioning of covariance matrices and, consequently, the numerical properties of the estimator [15].

The measurement vector

$$\tilde{\boldsymbol{y}}[n] = [\tilde{\rho}[n], \tilde{\alpha}[n], \tilde{\varepsilon}[n]]^{\mathrm{T}} \tag{4}$$

characterised by a covariance matrix $\boldsymbol{R}[n] = \operatorname{diag}\{\sigma_\rho^2, \sigma_\alpha^2, \sigma_\varepsilon^2\}$. It is also assumed that the input and measurement noises are non-stationary, in general.

The output matrix

$$\boldsymbol{C}_j = [\mathbf{I}_{3\times 3}, \mathbf{0}_{3\times 3}, \mathbf{0}_{3\times 2}] \tag{5}$$

is common for both models and completes Eq. (2).

Note that for the particular application considered [12] the process equation (1) is nonlinear, while the measurement equation (2) remains linear, which results from formulating the models (H_0, H_1) in sensor principal coordinates. If this was not the case, Eq. (2) would be nonlinear and a suitable approximation of this nonlinearity must be applied [3].

C. Continuous-Discrete IMM

The complete description of state in a 2-model IMM filter at time n consists of *a-posteriori* estimates $\hat{\boldsymbol{x}}_j[n|n]$, $j = 0,1$, of the state vector (3), corresponding covariance matrices $\boldsymbol{P}_j[n|n]$ and model probabilities $\mu_j[n]$ that evaluate the hypothesis that the target is in the modal state H_j at time n.

A block diagram of the two-model IMM tracking filter is shown in Fig. 1. The algorithm can be divided into a state vector initiation procedure (IMM step 0) and 6 recursive steps (IMM step 1 to 6), which are arranged into two blocks (I and II) and separated by the measurement validation procedure.

D. Implementing Matrix Computations Using BLAS

The BLAS (Basic Linear Algebra Subprograms) is a set of routines for performing basic vector and matrix operations [16]. The Level 1 BLAS includes scalar, vector and vector-vector operations

$$\mathbf{y} \leftarrow \alpha\mathbf{x} + \mathbf{y} \tag{6}$$

as well as, scalar dot products and vector norms. The Level 2 BLAS includes matrix-vector operations

$$\mathbf{y} \leftarrow \alpha\mathbf{A}\mathbf{x} + \beta\mathbf{y} \tag{7}$$

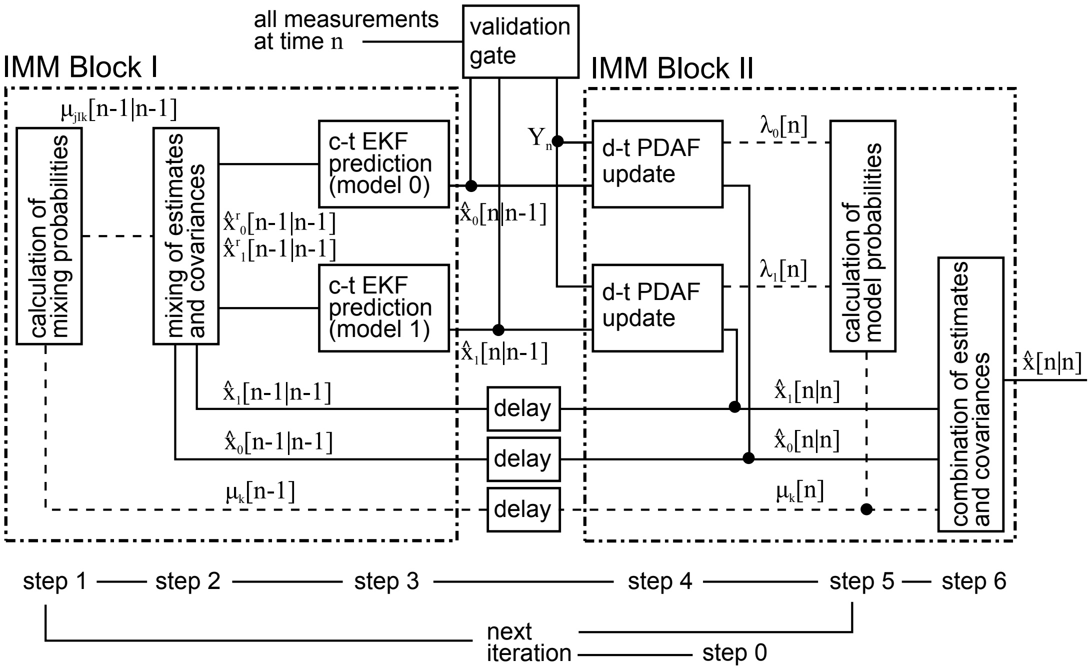

Fig. 1. Structure of the two-model continuous-discrete IMM filter (data processing view).

and provides functions for solving $\mathbf{A}_T\mathbf{x} = \mathbf{y}$. The Level 3 BLAS performs matrix-matrix operations

$$\mathbf{C} \leftarrow \alpha\mathbf{A}\mathbf{B} + \beta\mathbf{C} \tag{8}$$

and includes routines for solving $\mathbf{D} \leftarrow \alpha\mathbf{A}_T^{-1}\mathbf{B}$. Since BLAS libraries are efficient, portable, and open source, they are often used as a basis for the developing linear algebra software.

During the course of this paper the following BLAS functions are referred to

$$\texttt{daxpy}: \mathbf{y} \leftarrow \alpha\mathbf{x} + \mathbf{y} \tag{9}$$

$$\texttt{dgemv}: \mathbf{y} \leftarrow \alpha\mathbf{A}\mathbf{x} + \beta\mathbf{y} \tag{10}$$

$$\texttt{dsyr}: \mathbf{A}_S \leftarrow \alpha\mathbf{x}\mathbf{x}^{\mathrm{T}} + \mathbf{A}_S \tag{11}$$

$$\texttt{dsyr2}: \mathbf{A}_S \leftarrow \alpha\mathbf{x}\mathbf{y}^{\mathrm{T}} + \alpha\mathbf{y}\mathbf{x}^{\mathrm{T}} + \mathbf{A}_S \tag{12}$$

$$\texttt{dsyr2k}: \mathbf{A}_S \leftarrow \alpha\mathbf{B}\mathbf{D}^{\mathrm{T}} + \alpha\mathbf{D}\mathbf{B}^{\mathrm{T}} + \beta\mathbf{A}_S \tag{13}$$

Three additional functions were developed “in-house” that deal with symmetric matrices

$$\texttt{dsycpy}: \mathbf{A}_S \leftarrow \mathbf{B}_S \tag{14}$$

$$\texttt{dsysum}: \mathbf{A}_S \leftarrow \mathbf{A}_S + \alpha\mathbf{B}_S \tag{15}$$

$$\texttt{dsyform}: \mathbf{A}_S \leftarrow \alpha\mathbf{A}_S + \beta\mathbf{D}\mathbf{B}_S\mathbf{D}^{\mathrm{T}} \tag{16}$$

BLAS functions are designed to use minimum computer memory for data storage and minimum number of operations by exploiting the properties of different types of matrices. Recent BLAS implementation called ATLAS (Automatically Tuned Linear Algebra Software) provides also support for certain processor features like registers, cache memory or vector processing units, which can further improve matrix computation performance on various target architectures [17].

III. C-D Algorithm Implementation

The pseudocode algorithm of a single iteration of the two-model *c-d* IMM filter is shown in Fig. 2. The particular algorithm functions and their implementation details are described in the following subsections.

It is assumed that a target is detected during the presumed part of flight and that the detector provides a set of unbiased measurements, which are used to calculate initial values of state estimates $\hat{\boldsymbol{x}}_j[n_0|n_0]$. The values of the *a-priori* transition probabilities $\{p_{j,k}\}_{j,k=0}^{1}$ are assumed known and constant. The *a-posteriori* model probabilities have their initial values $\{\mu_j[n_0]\}_{j=0}^{1}$.

A. IMM Block 1: Prediction

a) IMM Step 1: Transition Probabilities: In order to produce input values for particular filters, conditional transition (mixing) probabilities are calculated, which describe probability that the target made the transition from modal state H_j to H_k at the instant $n-1$, based on data collected up to $n-1$. Scalar formulae

$$\mu_{j|k}^{n-1} = \frac{1}{c_k[n-1]} p_{j,k}\mu_j[n-1] \tag{17}$$

for $j, k = 0, 1$, with normalisation constants $c_k[n-1]$, are calculated without using BLAS.

b) IMM Step 2: Mixing of Estimates: Before model-matched filtering (steps 3 and 4) the operation of mixing partial estimates and respective covariance matrices is performed

$$\hat{\boldsymbol{x}}_k^{\mathrm{r}}[n-1|n-1] = \sum_{j=0}^{1} \mu_{j|k}^{n-1}\hat{\boldsymbol{x}}_j[n-1|n-1] \tag{18}$$

where $\hat{\boldsymbol{x}}_j[n-1|n-1]$ is the state estimate yielded by the j-th mode-matched filter during previous iteration, while

```
/* IMM Block 1: Prediction */
imm_step1_calculate_mixing_probabilities (&Pij, &Uj, &Uij, &Cj);

imm_step2_calculate_mixed_estimate (0, &Uij, &X0RnIn, &P0RnIn, &X0nIn, &P0nIn, &X1nIn, &P1nIn);
imm_step2_calculate_mixed_estimate (1, &Uij, &X1RnIn, &P1RnIn, &X0nIn, &P0nIn, &X1nIn, &P1nIn);

Tn = calculate_prediction_period(ART, &XnIn);
imm_step3_propagate_ctekf (Tn, dt, &X0RnIn, &P0RnIn, &X0nIn_1, &P0nIn_1);
imm_step3_propagate_ctekf (Tn, dt, &X1RnIn, &P1RnIn, &X1nIn_1, &P1nIn_1);

/* Selection of new measurements */
get_new_plots (&measurements);
associated = gating_and_association (&measurements, &X0nIn_1, &P0nIn_1, &X1nIn_1, &P1nIn_1);

/* IMM Block 2: Filtering */
if (associated) { /* PDAF measurement update */
  imm_step4_update_pdaf (&measurements, &X0nIn_1, &P0nIn_1, &X0nIn, &P0nIn, ...);
  imm_step4_update_pdaf (&measurements, &X1nIn_1, &P1nIn_1, &X1nIn, &P1nIn, ...);
} else {  /* No plots. Track extrapolation */
  imm_step4_update_plotmiss (&X0nIn_1, &P0nIn_1, &X0nIn, &P0nIn, ...);
  imm_step4_update_plotmiss (&X1nIn_1, &P1nIn_1, &X1nIn, &P1nIn, ...);
}

imm_step5_calculate_model_probabilities (&Lj, &Cj, &Uj);

imm_step6_calculate_global_estimate (&Uj, &XnIn, &PnIn, &X0nIn, &P0nIn, &X1nIn, &P1nIn);
```

Fig. 2. Pseudocode algorithm of the single iteration of the two-model continuous-discrete IMM.

$$\boldsymbol{P}^{\mathrm{r}}_k[n-1|n-1] = \sum_{j=0}^{1} \mu^{n-1}_{j|k} \cdot$$
$$\cdot \Big(\boldsymbol{P}_k[n-1|n-1] + \Delta\hat{\boldsymbol{x}}_{j,k}[n-1]\Delta\hat{\boldsymbol{x}}^{\mathrm{T}}_{j,k}[n-1]\Big) \tag{19}$$

with

$$\Delta\hat{\boldsymbol{x}}_{j,k}[n-1] = \hat{\boldsymbol{x}}_j[n-1|n-1] - \hat{\boldsymbol{x}}^{\mathrm{r}}_k[n-1|n-1] \tag{20}$$

Eq. (18) is calculated using looped function daxpy (9), while (19) is calculated using a looped formula

$$\mathbf{A}_P = \mathbf{A}_P + \alpha(\mathbf{B}_P + (\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^{\mathrm{T}}) \tag{21}$$

with (21) being split into the following steps

$$\mathbf{A}_P = \mathbf{A}_P + \alpha\mathbf{B}_P \tag{22}$$
$$\mathbf{A}_P = \mathbf{A}_P + \alpha\mathbf{x}\mathbf{x}^{\mathrm{T}} \tag{23}$$
$$\mathbf{A}_P = \mathbf{A}_P + \alpha\mathbf{y}\mathbf{y}^{\mathrm{T}} \tag{24}$$
$$\mathbf{A}_P = \mathbf{A}_P - \alpha(\mathbf{x}\mathbf{y}^{\mathrm{T}} + \mathbf{y}\mathbf{x}^{\mathrm{T}}) \tag{25}$$

As formula (22) has no corresponding BLAS procedure, a dedicated function (15) is used, which implements the sum of symmetric matrices. Eqns. (23) and (24) are calculated using BLAS dsyr (11), while (25) using dsyr2 (12).

c) IMM Step 3: C-T EKF Prediction: A time period for c-t prediction can be estimated as

$$\hat{T}_n = \frac{2\pi}{\Delta\dot{\alpha}[n|n]} \tag{26}$$

with

$$\Delta\dot{\alpha}[n|n] = \dot{\alpha}_R - \frac{\hat{v}_\alpha[n|n]}{\hat{\rho}[n|n]\cos(\hat{\varepsilon}[n|n])} \tag{27}$$

where the rotational speed of the radar antenna $\dot{\alpha}_R = \frac{2\pi}{\mathrm{ART}}$ is assumed to be constant in time, and $\dot{\alpha}_R >> \frac{\hat{v}_\alpha[n|n]}{\hat{\rho}[n|n]\cos(\hat{\varepsilon}[n|n])}$.

The state and covariance prediction step of the state estimator is based on *c-t* EKF

$$\dot{\hat{\boldsymbol{x}}}_j(t) = \boldsymbol{f}_j\big(t, \hat{\boldsymbol{x}}_j(t)\big) \tag{28}$$

$$\dot{\boldsymbol{P}}_j(t) = \boldsymbol{F}_j\big(t, \hat{\boldsymbol{x}}_j(t)\big)\boldsymbol{P}_j(t) +$$
$$+ \boldsymbol{P}_j(t)\boldsymbol{F}^{\mathrm{T}}_j\big(t, \hat{\boldsymbol{x}}_j(t)\big) + \boldsymbol{W}_j(t) \tag{29}$$

where

$$\boldsymbol{F}_j\big(t, \hat{\boldsymbol{x}}_j(t)\big) = \left.\frac{\partial \boldsymbol{f}_j\big(t, \boldsymbol{x}_j(t)\big)}{\partial \boldsymbol{x}_j(t)}\right|_{\boldsymbol{x}_j(t) = \hat{\boldsymbol{x}}_j(t)} \tag{30}$$

is the Jacobian matrix of $\boldsymbol{f}_j(\cdot)$ [2], [3].

By numerically solving differential equations (28) and (29) with respect to their initial values

$$\hat{\boldsymbol{x}}_j(t_{n-1}) = \hat{\boldsymbol{x}}^{\mathrm{r}}_j[n-1|n-1] \tag{31}$$
$$\boldsymbol{P}_j(t_{n-1}) = \boldsymbol{P}^{\mathrm{r}}_j[n-1|n-1] \tag{32}$$

one obtains the predicted state vector and the respective prediction error covariance matrix

$$\hat{\boldsymbol{x}}_j[n|n-1] = \hat{\boldsymbol{x}}_j(t_n) \tag{33}$$
$$\boldsymbol{P}_j[n|n-1] = \boldsymbol{P}_j(t_n) \tag{34}$$

The numerical integration procedure uses rectangular principle, BLAS 1 daxpy (9) for calculating state estimate (28), and BLAS 3 dsyr2k (13) for covariance matrix (29).

In fact, the integration of (28) and (29), which includes multiple calculations of the nonlinear model and its Jacobian matrix, is the most numerically expensive part of the discussed algorithm. In order to reduce the computational complexity the *c-t* propagation of state (28) can be performed accurately using the geodetic framework, while calculations of matrices (29) can be approximated using the spherical Earth model.

B. Selection of Measurements

There are multiple radar plots incoming to the tracking filter, which are either measurements of detected targets or false alarms. In order to limit the amount of processed data the plots are validated on the basis of a statistical test (condition on the Mahalanobis distance)

$$\boldsymbol{\nu}_{j,i}^{\mathrm{T}}[n]\boldsymbol{\omega}_j^{-1}[n]\boldsymbol{\nu}_{j,i}[n] \leq G \tag{35}$$

where

$$\boldsymbol{\nu}_{j,i}[n] = \tilde{\boldsymbol{y}}_i[n] - \hat{\boldsymbol{y}}_j[n] \tag{36}$$

is the KF innovation process characterised by its covariance matrix $\boldsymbol{\omega}_j[n]$,

$$\hat{\boldsymbol{y}}_j[n] = \boldsymbol{C}_j\hat{\boldsymbol{x}}_j[n|n-1] \tag{37}$$

denotes a predicted target position yielded by the H_j filter, while G denotes a suitable thresholding value that constitutes a correlation/validation gate.

Based on the above rule an Y_n-element set $\boldsymbol{Y}_n$ of measurements $\tilde{\boldsymbol{y}}_i[n]$, $1,\ldots,\mathrm{Y}_n$ is selected, which contains data recorded in the neighbourhoods of the target positions $\hat{\boldsymbol{y}}_j[n]$ at instant n, as predicted by filters H_0 and H_1. Additionally, another set $\boldsymbol{Y}^n = \{\boldsymbol{Y}_m\}_{m=0}^{n}$ consists of all measurements associated with the considered track up to the instant n.

C. IMM Block 2: Filtering

d) IMM Step 4: PDA Filter Update: For each measurement $\tilde{\boldsymbol{y}}_i[n]$, $i = 1,\ldots,\mathrm{Y}_n$, at time n a conditional probability $\beta_{j,i}^n = \beta_{j,i}[n]$ is evaluated for $j = 0, 1$ that $\tilde{\boldsymbol{y}}_i[n]$ is a true (target originated) measurement for the filter H_j. The non-parametric PDA algorithm is used [3]

$$\hat{\boldsymbol{x}}_j[n|n] = \hat{\boldsymbol{x}}_j[n|n-1] + \boldsymbol{L}_j[n]\boldsymbol{\nu}_j[n] \tag{38}$$

$$\begin{aligned}\boldsymbol{P}_j[n|n] = \beta_{j,0}^n\boldsymbol{P}_j[n|n-1] + (1-\beta_{j,0}^n)\boldsymbol{P}_j^{\mathrm{c}}[n|n] +& \\ + \boldsymbol{L}_j[n]\boldsymbol{V}_j[n]\boldsymbol{L}_j^{\mathrm{T}}[n]&\end{aligned} \tag{39}$$

with

$$\boldsymbol{\nu}_j[n] = \sum_{i=1}^{\mathrm{Y}_n}\beta_{j,i}^n\boldsymbol{\nu}_{j,i}[n] \tag{40}$$

$$\boldsymbol{L}_j[n] = \boldsymbol{P}_j[n|n-1]\boldsymbol{C}_j^{\mathrm{T}}\boldsymbol{\omega}_j^{-1}[n] \tag{41}$$

$$\boldsymbol{\omega}_j[n] = \boldsymbol{C}_j\boldsymbol{P}_j[n|n-1]\boldsymbol{C}_j^{\mathrm{T}} + \boldsymbol{R}[n] \tag{42}$$

$$\boldsymbol{V}_j[n] = \sum_{i=1}^{\mathrm{Y}_n}\beta_{j,i}^n\boldsymbol{\nu}_{j,i}[n]\boldsymbol{\nu}_{j,i}^{\mathrm{T}}[n] - \boldsymbol{\nu}_j[n]\boldsymbol{\nu}_j^{\mathrm{T}}[n] \tag{43}$$

The Joseph's formula [7]

$$\boldsymbol{P}_j^{\mathrm{c}}[n|n] = \boldsymbol{K}[n]\boldsymbol{P}_j[n|n-1]\boldsymbol{K}^{\mathrm{T}}[n] + \boldsymbol{L}_j[n]\boldsymbol{R}[n]\boldsymbol{L}_j^{\mathrm{T}}[n] \tag{44}$$

is used to calculate the filtering error covariance matrix for a single and true measurement hypothesis, with

$$\boldsymbol{K}[n] = \mathbf{I}_{8\times 8} - \boldsymbol{L}_j[n]\boldsymbol{C}_j \tag{45}$$

Eq. (38) is calculated using BLAS 2 `dgemv` (10). Aggregation in Eq. (40) is accomplished using looped BLAS 1 `daxpy` (9) and the first term of Eq. (43) using looped BLAS 2 `dsyr` (11). Eq. (36) corresponds to

$$\mathbf{y} = \beta\mathbf{y} - \alpha\mathbf{A}\mathbf{x} \tag{46}$$

which may be implemented using BLAS 2 `dgemv` (10), however, as $\boldsymbol{C}_j$ is the sparse matrix, this expression is calculated element-by-element. Sparsity of $\boldsymbol{C}_j$ implies that also Eqns. (41), (42) and (45) are calculated without using BLAS.

By inserting (43) and (44) to (39) a closed form

$$\begin{aligned}\mathbf{A}_P = \alpha\mathbf{B}_P + (1-\alpha)(\mathbf{G}\mathbf{B}_P\mathbf{G}^{\mathrm{T}} + \mathbf{D}\mathbf{E}_P\mathbf{D}^{\mathrm{T}}) +& \\ + \mathbf{D}(\mathbf{H}_P - \mathbf{x}\mathbf{x}^{\mathrm{T}})\mathbf{D}^{\mathrm{T}}&\end{aligned} \tag{47}$$

is obtained, which can be split into the following steps

$$\mathbf{H}_P = \mathbf{H}_P - \mathbf{x}\mathbf{x}^{\mathrm{T}} \tag{48}$$

$$\mathbf{H}_P = \mathbf{H}_P + (1-\alpha)\mathbf{E}_P \tag{49}$$

$$\mathbf{A}_P = \mathbf{B}_P \tag{50}$$

$$\mathbf{A}_P = \alpha\mathbf{A}_P + (1-\alpha)\mathbf{G}\mathbf{B}_P\mathbf{G}^{\mathrm{T}} \tag{51}$$

$$\mathbf{A}_P = \mathbf{A}_P + \mathbf{D}\mathbf{H}_P\mathbf{D}^{\mathrm{T}} \tag{52}$$

Calculation of (48) uses BLAS 2 function `dsyr` (11). Eq. (49) is the sum of symmetric matrices done by (15), while (50) is simple copying of symmetric matrices (14). Last two formulae (51) and (52) are similar, however, as no corresponding function for symmetric matrices is present in the standard BLAS, a dedicated routine (16) is employed that implements this type of calculations.

If at time n no new measurements are correlated to the track, the procedure proposed by [18] is used. The method substitutes the current estimate $\hat{\boldsymbol{x}}_j[n|n]$ with its prediction $\hat{\boldsymbol{x}}_j[n|n-1]$ and approximates the respective covariance matrix

$$\boldsymbol{P}_j[n|n] = \boldsymbol{P}_j[n|n-1] + q_{\mathrm{T}}^n\boldsymbol{L}_j[n]\boldsymbol{\omega}_j[n]\boldsymbol{L}_j^{\mathrm{T}}[n] \tag{53}$$

for $j = 0, 1$. The second component of (53) describes an additional increase of uncertainty introduced into the estimation process due to the fact that no new measurements can be found in the neighbourhood of the predicted target position.

Eq. (53) corresponds to

$$\mathbf{A}_P = \mathbf{B}_P + \alpha\mathbf{G}\mathbf{D}_P\mathbf{G}^{\mathrm{T}} \tag{54}$$

which can be divided into two steps

$$\mathbf{A}_P = \mathbf{B}_P \tag{55}$$

$$\mathbf{A}_P = \mathbf{A}_P + \alpha\mathbf{C}\mathbf{D}_P\mathbf{C}^{\mathrm{T}} \tag{56}$$

that fit (14) and (16), respectively.

e) IMM Step 5: Model Probabilities: Modal state probabilities are updated in each iteration of the filter according to a scalar formula

$$\mu_j[n] = \frac{1}{c[n]}\Lambda_j[n]c_j[n-1] \tag{57}$$

for $j = 0, 1$, where the likelihood function $\Lambda_j[n]$ describes Υ_n association hypotheses corresponding to each observation in the association gate and to the hypothesis that none of the measurements is valid.

If no new measurements are correlated to the track, model probabilities are substituted by their values from previous iteration of the filter $\mu_j[n] = \mu_j[n-1]$ for $j = 0, 1$.

f) IMM Step 6: Global Estimate: At the end of the IMM filter iteration an overall (combined) estimate

$$\hat{\boldsymbol{x}}[n|n] = \sum_{j=0}^{1} \mu_j[n]\hat{\boldsymbol{x}}_j[n|n] \tag{58}$$

is calculated using looped BLAS 1 `daxpy` (9).

The corresponding covariance matrix can be defined as

$$\boldsymbol{P}[n|n] = \sum_{j=0}^{1} \mu_j[n]\left(\boldsymbol{P}_j[n|n] + \Delta\hat{\boldsymbol{x}}_j[n]\Delta\hat{\boldsymbol{x}}_j^{\mathrm{T}}[n]\right) \tag{59}$$

with

$$\Delta\hat{\boldsymbol{x}}_j[n] = \hat{\boldsymbol{x}}_j[n|n] - \hat{\boldsymbol{x}}[n|n] \tag{60}$$

Calculation of (59) is similar to that of (19), which implies the usage of (21-25).

IV. Conclusions

In this paper an approach was presented to implementing a complex recursive state estimation algorithm using the BLAS library. Careful analysis of numerical properties of mathematical formulae make allowances for selecting most computationally effective BLAS functions.

Additionally, the following observations and ideas helped assuring convergence and numerical stability, and reducing the computational complexity of the filter:

1) Expressing the target movement model in sensor-principal coordinates, which results in a linear measurement model and in a diagonally-dominant covariance matrices.
2) Applying the Cartesian line-of-sight velocity coordinates, which improves numerical conditioning of the covariance matrices.
3) *C-t* propagation of state and covariance, which for a suitably chosen integration interval ensures estimator convergence. It is not the case when the *d-t* EKF is applied with such a low data rate.
4) Employing two sets of models. The more accurate one using the geodetic Earth model for predicting the state (28) and the simplified one using the spherical Earth model for propagating its covariance matrix (29). This saves a substantial computational power not affecting the estimator convergence or accuracy.
5) Using BLAS routines dedicated to symmetric matrices, which reduces memory storage requirements and guarantees symmetry of the relevant matrices.
6) Applying the Joseph's formula (44) for propagating the covariance matrix, which neutralises the risk imposed by the subtraction operation in the original KF equations.
7) Calculating formulae involving sparse matrices of known structure element-by-element.

The implemented tracking filter was observed to be numerically stable during laboratory tests and when integrated within a real radar system. Although based on computationally expensive techniques and using a floating point representation of numbers, the applied techniques summarised above enabled simultaneous processing of over 100 ballistic tracks in real time using a single-core general-purpose processor working with the 800 [MHz] clock.

References

[1] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME*, vol. 82D, pp. 35–45, Mar. 1960.

[2] R. Kalman and R. Bucy, "New results in linear filtering and prediction theory," *Trans. ASME*, vol. 83D, pp. 95–108, Mar. 1961.

[3] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY: John Wiley & Sons, Inc., 2001.

[4] P. Costa, "Adaptive model architecture and extended Kalman-Bucy filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 2, pp. 525–533, Apr. 1994.

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

[6] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, pp. 451–460, 1975.

[7] G. Bierman, *Factorization Methods for Discrete Sequential Estimation*. New York: Academic Press, 1977.

[8] R. Kenefic, "Active sonar application of a U-D square root PDAF," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 5, pp. 850–857, Sep. 1990.

[9] X. Rong Li and Y. Zhang, "Numerically robust implementation of multiple-model algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 266–278, Jan. 2000.

[10] J. Jørgensen, P. Thomsen, H. Madsen, and M. Kristensen, "A computationally efficient and robust implementation of the continuous-discrete extended Kalman filter," in *Proc. American Control Conf.*, New York, NY, Jul. 2007, pp. 3706–3712.

[11] T. Mazzoni, "Computational aspects of continuous-discrete extended Kalman-filtering," *Computational Statistics*, vol. 23, pp. 519–539, Oct. 2008.

[12] M. Sankowski, "Continuous-discrete estimation for tracking ballistic missiles in air surveillance radar," *IET Radar, Sonar and Navig.*, vol. 5, no. 9, pp. 978–986, Dec. 2011.

[13] M. Sankowski and W. Buda, "Numerical implementation of continuous-discrete IMM state estimators," in *Proc. Int. Radar Symp. IRS*, Vilnius, Lithuania, Jun. 2010, pp. 474–477.

[14] G. Cardillo, A. Mrstik, and T. Plambeck, "A track filter for reentry objects with uncertain drag," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 2, pp. 394–408, Apr. 1999.

[15] F. Daum and R. Fitzgerald, "Decoupled Kalman filters for phased array radar tracking," *IEEE Trans. Autom. Control*, vol. 28, no. 3, pp. 269–283, Mar. 1983.

[16] "BLAS – Basic Linear Algebra Subprograms," 2010, http://www.netlib.org/blas/.

[17] "ATLAS – Automatically Tuned Linear Algebra Software," 2011, http://math-atlas.sourceforge.net.

[18] X. Rong Li, "Tracking in clutter with strongest neighbor measurements – Part I: Theoretical analysis," *IEEE Trans. Autom. Control*, vol. 43, no. 11, pp. 1560–1578, Nov. 1998.