A general Weighted Total Kalman Filter algorithm with numerical evaluation

VAHID MAHBOUB, MOHAMMAD SAADATSERESHT* AND ALIREZA A. ARDALAN

School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran (msaadat@ut.ac.ir)

* Corresponding author

Received: March 6, 2016; Revised: May 23, 2016; Accepted: August 2, 2016

ABSTRACT

An applicable algorithm for Total Kalman Filter (TKF) approach is proposed. Meanwhile, we extend it to the case in which we can consider arbitrary weight matrixes for the observation vector, the random design matrix and possible correlation between them. Also the updated dispersion matrix of the predicted unknown is given. This approach makes use of condition equations and straightforward variance propagation rules. It is applicable to data fusion within a dynamic errors-in-variables (DEIV) model, which usually appears in the determination of the position and attitude of mobile sensors. Then, we apply for the first time the TKF algorithm and its extended version named WTKF to a DEIV model and compare the results. The results show the efficiency of the proposed WTKF algorithm. In particular in the case of large weights, WTKF shows approximately 25% improvement in contrast to TKF approach.

Keywords: Total Kalman filter, dynamic errors-in-variables model, prediction, weight matrix, mobile mapping, kinematic positioning

1. INTRODUCTION

The Kalman filter is a mathematical power tool that is playing an increasingly important role in engineering applications. If the unknown parameters are time dependent, one encounters a dynamic problem which is solved using the Kalman filter. In the literature, the Kalman filter is derived as either a best predictor (BP) or a best linear predictor (BLP), see e.g. *Kalman (1960)*, *Gelb (1974)*, *Sanso (1986)*, *Teunissen and Khodabandeh (2013)*. The minimum mean squared error (*MMSE*) is the criterion which selects the best predictor or estimator.

The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem (*Kalman, 1960*). A “friendly” introduction to the idea of the Kalman filter is given in *Maybeck (1979)*.A more complete introductory discussion can be found in *Sorenson (1970)*.

The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated *error* covariance, when some presumed conditions are met (*Welch and Bishop, unpublished results*).

Several linear and non-linear Kalman filters have been proposed. For a perfect review we may refer to *Yi (2007)*. For example, we mention to Extended Kalman Filter (EKF) (*Jazwinski, 1970*), the Sigma Point Kalman Filters (SPKF) (*van der Merwe and Wan, 2004*) or Linear Regression Kalman Filters (LRKF) (*Lefebvre et al., 2002*), the Particle Filters (PF) (*Liu and Chen, 1998*; *Doucet et al., 2001*), the Ensemble Kalman Filter (EnKF) (*Evensen, 1994, 1997, 2007*), Unscented Kalman Filter(UKF) based on unscented transformation (UT) (*Julier et al., 1995*; *Julier and Uhlmann, 1997*), etc. However, in all of these algorithms, the design matrix of the observation equations does not contain random errors. As such an assumption cannot always be guaranteed, *Schaffrin and Iz (2008)* allowed random observational errors to enter the respective matrices and named it Total Kalman Filter (TKF). The corresponding model was considered as a dynamic errors-in-variables (DEIV) model. Then *Schaffrin and Uzun (2011)* tried to impose a data snooping procedure to the TKF algorithm. Nevertheless, the TKF algorithm considers a restricted structure for the weight matrixes of both of the observation equations and the dynamic model. Also this algorithm has not been examined numerically.

In this paper first we extend the TKF algorithm to a general weighted TKF (WTKF) algorithm. Similar to *Schaffrin and Iz (2008)*, our approach makes use of condition equations and straightforward variance propagation rules. Then this algorithm is applied to a DEIV model in a numerical example and compared with the TKF algorithm of *Schaffrin and Iz (2008)*.

This paper is organized as follows: in Section 2, the DEIV model and TKF algorithm are introduced. In Section 3, the general algorithm of WTKF is developed, then, in a later section, a numerical example gives insight into the efficiency of the algorithm proposed. Finally we conclude the paper.

2. DYNAMIC ERRORS-IN-VARIABLES MODEL AND TOTAL KALMAN FILTER ALGORITHM

In the last two decades, several publications concerning errors-in-variables (EIV) model have been published in geodetic literature. We may for instance refer to *Schaffrin et al. (2014)*, *Schaffrin and Felus (2008)*, *Mahboub (2012, 2014, 2016)*, *Mahboub et al. (2013, 2015)*, *Neitzel (2010)*, *Neitzel and Schaffrin (2016)*, *Snow and Schaffrin (unpublished results)*, *Fang (2011, 2014)*, *Xu et al. (2012)*, *Shen et al. (2011)*, etc. If the unknown parameters of the EIV model are time dependent, one encounters a dynamic errors-in-variables (DEIV) model. An initial version of this model besides a solution namely total Kalman filter (TKF) has been given by *Schaffrin and Iz (2008)*. It has been defined at an epoch i as follows:

$$\underline{\boldsymbol{y}}_i = \left(\mathbf{A}_i - \mathbf{E}_{A_i} \right) \underline{\boldsymbol{x}}_i + \underline{\boldsymbol{e}}_i \ , \tag{1}$$

where $\underline{\boldsymbol{y}}_i$ is the $m \times 1$ random observation vector, $\underline{\boldsymbol{e}}_i$ is the $m \times 1$ vector of observational noise, $\mathbf{A}_i$ is the $m \times n$ coefficient matrix of input variables (observed), $\underline{\mathbf{E}}_{A_i}$ is the corresponding $m \times n$ matrix of random noise, $\underline{\boldsymbol{x}}_i$ is the $n \times 1$ random parameter vector (time dependent unknowns).

It is to be noted that in this paper reversed hats indicate predicted vectors (e.g., $\breve{\boldsymbol{x}}$), tildas denote estimated ones (e.g., $\tilde{\boldsymbol{\lambda}}$), while underlining indicates random variables (e.g., $\underline{\boldsymbol{x}}_i$).

Suppose system equations are known and can be presented as

$$\underline{\boldsymbol{x}}_i = \boldsymbol{\Phi}_i \boldsymbol{x}_{i-1} + \underline{\boldsymbol{u}}_i \ , \tag{2}$$

where $\boldsymbol{\Phi}_i$ is the transition matrix and $\underline{\boldsymbol{u}}_i$ is the random system noise of this model which is also called dynamic model. This equation relates the unknown parameters at an epoch i to an earlier epoch $i-1$. We also assume that the state vector is observed at an initial (previous) epoch:

$$\underline{\boldsymbol{x}}_{i-1} = \boldsymbol{x}_{i-1} + \underline{\boldsymbol{e}}_{i-1}^{0} \ , \tag{3}$$

where $\tilde{\boldsymbol{e}}_{i-1}^{0}$ is the random noise at the first epoch. However, they considered a restricted structure for the dispersion matrixes of the observation vector and the random design matrix as follows:

$$\begin{bmatrix} \underline{\boldsymbol{e}}_{A_i} = \mathrm{vec}\left(\underline{\mathbf{E}}_{A_i}\right) \\ \underline{\boldsymbol{e}}_i \\ \underline{\boldsymbol{u}}_i \\ \underline{\boldsymbol{e}}_{i-1}^{0} \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{I}_n \otimes \boldsymbol{\Sigma}_i & 0 & 0 & 0 \\ 0 & \boldsymbol{\Sigma}_i & 0 & 0 \\ 0 & 0 & \boldsymbol{\theta}_i & 0 \\ 0 & 0 & 0 & \boldsymbol{\Sigma}_{i-1}^{0} \end{bmatrix} \right) , \tag{4}$$

where $\boldsymbol{\Sigma}_i$, $\boldsymbol{\theta}_i$ and $\boldsymbol{\Sigma}_{i-1}^{0}$ are the corresponding dispersion matrices of the observation vector, system equations and the observed unknown parameters at an initial epoch. As it can be seen in Eq. (4), there is the restrictive condition for the dispersion matrix of the coefficients and the observation vector. Hence, if some modifications can be inserted to their analytical solution, an algorithm for the TKF in general form will be obtained. The fairly general structure of the dispersion matrix for the observable quantities allows one to insert a perfect description of the weight matrixes besides the correlations among all the random quantities. For this initial DEIV model given by Eqs (1)–(4), *Schaffrin and Iz (2008)* predicted the state vector $\underline{\boldsymbol{x}}_i$ as follows and named it the TKF solution:

$$\breve{\boldsymbol{x}}_i = \hat{\boldsymbol{x}}_i + \left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^{0} \boldsymbol{\Phi}_i^{\mathrm{T}}\right)\left[\mathbf{A}_i^{\mathrm{T}} \tilde{\boldsymbol{\lambda}}_i + \breve{\boldsymbol{x}}_{i-1}\left(\tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}} \boldsymbol{\Sigma}_i \tilde{\boldsymbol{\lambda}}_i\right)\right] , \tag{5}$$

where $\tilde{\boldsymbol{\lambda}}_i$ and $\hat{\boldsymbol{x}}_i$ are given as follows:

$$\tilde{\boldsymbol{\lambda}}_i = \left(\boldsymbol{\Sigma}_i\right)^{-1}\left(\boldsymbol{y}_i - \mathbf{A}_i\breve{\boldsymbol{x}}_i\right)\left(1 + \breve{\boldsymbol{x}}_i^{\mathrm{T}}\breve{\boldsymbol{x}}_i\right)^{-1} , \tag{6}$$

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\Phi}_i\breve{\boldsymbol{x}}_{i-1} . \tag{7}$$

3. A GENERAL WEIGHTED TOTAL KALMAN FILTER ALGORITHM

Now if we consider a fully correlated dispersion matrix for the observed coefficient matrix and the observation vector, the general weighted total Kalman filter (WTKF) algorithm can be obtained. In other words, we redefine Eq. (4) as follows and then the WTKF algorithm is derived based on this assumption:

$$\begin{bmatrix} \underline{\boldsymbol{E}}_i \\ \underline{\boldsymbol{u}}_i \\ \underline{\boldsymbol{e}}_{i-1}^0 \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{Q} & 0 & 0 \\ 0 & \boldsymbol{\theta}_i & 0 \\ 0 & 0 & \boldsymbol{\Sigma}_{i-1}^0 \end{bmatrix}\right) , \tag{8}$$

where $\mathbf{Q}$ is the fully correlated dispersion matrix of size $(mn+m)\times(mn+m)$ and the random error vector $\underline{\boldsymbol{E}}_i$ of size $(mn+m)\times 1$ is given by:

$$\underline{\boldsymbol{E}}_i = \begin{bmatrix} \mathrm{vec}\left(\underline{\mathbf{E}}_{A_i}\right) \\ \underline{\boldsymbol{e}}_i \end{bmatrix} , \tag{9}$$

$$\mathrm{vec}\left(\underline{\mathbf{E}}_{A_i}\right) = \mathbf{M}\underline{\boldsymbol{E}}_i , \tag{10}$$

$$\underline{\boldsymbol{e}}_i = \mathbf{N}\underline{\boldsymbol{E}}_i , \tag{11}$$

with $\mathbf{M} = \left[\mathbf{I}_{nm\times nm} \ \vdots \ 0_{nm\times nm}\right]$ and $\mathbf{N} = \left[0_{m\times nm} \ \vdots \ \mathbf{I}_{m\times m}\right]$.

In some application an independent time variable function f_i may appear in system equations. Thus, comparing to *Schaffrin and Iz (2008)*, we can generalize Eq. (2) as follows:

$$\underline{\boldsymbol{x}}_i = \boldsymbol{\Phi}_i\boldsymbol{x}_{i-1} + f_i + \underline{\boldsymbol{u}}_i . \tag{12}$$

Since we are going to use the condition equations, the DEIV model must be combined with the state equations. In such a case, the unknown parameter x_i is omitted. Meanwhile, we apply variance propagation rules to the combined equations. To achieve this goal, first we insert Eq. (3) into Eq. (12) as follows:

$$\begin{aligned} \underline{\boldsymbol{x}}_i &= \boldsymbol{\Phi}_i\left(\underline{\boldsymbol{x}}_{i-1} - \underline{\boldsymbol{e}}_{i-1}^0\right) + f_i + \underline{\boldsymbol{u}}_i = \boldsymbol{\Phi}_i\underline{\boldsymbol{x}}_{i-1} - \boldsymbol{\Phi}_i\underline{\boldsymbol{e}}_{i-1}^0 + f_i + \underline{\boldsymbol{u}}_i \\ &= \hat{\underline{\boldsymbol{x}}}_i - \boldsymbol{\Phi}_i\underline{\boldsymbol{e}}_{i-1}^0 + \underline{\boldsymbol{u}}_i \\ &\rightarrow \underline{\boldsymbol{\mu}}_i = \underline{\boldsymbol{u}}_i - \boldsymbol{\Phi}_i\underline{\boldsymbol{e}}_{i-1}^0 = \underline{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_i , \end{aligned} \tag{13}$$

with $\underline{\hat{\boldsymbol{x}}}_i = \boldsymbol{\Phi}_i \underline{\boldsymbol{x}}_{i-1} + f_i$.

Based on variance propagation rules, the dispersion matrix of the random vector $\underline{\boldsymbol{\mu}}_i$ is obtained as follows:

$$\mathbf{D}\left(\underline{\boldsymbol{\mu}}_i\right) = \boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}} \ . \tag{14}$$

Eventually, by inserting Eq. (13) into Eq. (1), the combined model is given by:

$$\underline{\boldsymbol{y}}_i = \left(\mathbf{A}_i - \underline{\mathbf{E}}_{A_i}\right)\left(\underline{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i\right) + \underline{\boldsymbol{e}}_i \ . \tag{15}$$

The combined model (15) can be converted into the following equation due to Eqs (10) and (11):

$$\underline{\boldsymbol{y}}_i = \mathbf{A}_i\left(\underline{\boldsymbol{\mu}}_i + \underline{\breve{\boldsymbol{x}}}_i\right) - \left[\left(\underline{\boldsymbol{\mu}}_i + \underline{\hat{\boldsymbol{x}}}_i\right)^{\mathrm{T}} \otimes \mathbf{I}_m\right]\mathbf{M}\underline{\boldsymbol{E}}_i + \mathbf{N}\underline{\boldsymbol{E}}_i \ . \tag{16}$$

Here we made use of the following rule in linear algebra, see, e.g. *Magnus (1988)*:

$$\operatorname{vec}\left(\mathbf{ABC}\right) = \left(\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A}\right)\operatorname{vec}\left(\mathbf{B}\right) \ . \tag{17}$$

Then the stochastic model of the combined model (15) or (16) is as follows due to Eq. (14):

$$\begin{bmatrix} \underline{\boldsymbol{E}}_i \\ \underline{\boldsymbol{\mu}}_i \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}} \end{bmatrix}\right) \ . \tag{18}$$

The following Lagrange target function is given in order to compute the weighted total least-squares prediction of this problem:

$$\begin{aligned} \boldsymbol{\Phi}\left(\boldsymbol{E}_i, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i\right) &:= \boldsymbol{E}_i^{\mathrm{T}}\mathbf{Q}^{-1}\boldsymbol{E}_i + \boldsymbol{\mu}_i^{\mathrm{T}}\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)^{-1}\boldsymbol{\mu}_i \\ &+ 2\boldsymbol{\lambda}_i^{\mathrm{T}}\left\{\boldsymbol{y}_i - \mathbf{A}_i\left(\boldsymbol{\mu}_i + \hat{\boldsymbol{x}}_i\right) + \left[\left(\boldsymbol{\mu}_i + \hat{\boldsymbol{x}}_i\right)^{\mathrm{T}} \otimes \mathbf{I}_m\right]\mathbf{M}\boldsymbol{E}_i - \mathbf{N}\boldsymbol{E}_i\right\}, \end{aligned} \tag{19}$$

where $\boldsymbol{\lambda}_i$ is a $m \times 1$ vector of Lagrange multipliers.

For optimization, the following necessary conditions must hold:

$$\left.\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{E}_i}\right| \breve{\boldsymbol{E}}_i, \tilde{\boldsymbol{\lambda}}_i, \breve{\boldsymbol{\mu}}_i = 2\left\{\breve{\boldsymbol{E}}_i^{\mathrm{T}}\mathbf{Q}^{-1} - \tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}}\left[\mathbf{N} - \left(\left(\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i\right)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right]\right\} = 0 \ , \tag{20}$$

$$\left.\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{\mu}_i}\right| \breve{\boldsymbol{E}}_i, \tilde{\boldsymbol{\lambda}}_i, \breve{\boldsymbol{\mu}}_i = 2\left[\breve{\boldsymbol{\mu}}_i^{\mathrm{T}}\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)^{-1} - \tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}}\left(\mathbf{A}_i - \breve{\mathbf{E}}_{A_i}\right)\right] = 0 \ , \tag{21}$$

$$\left.\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{\lambda}_i}\right| \breve{\boldsymbol{E}}_i, \tilde{\boldsymbol{\lambda}}_i, \breve{\boldsymbol{\mu}}_i = 2\left\{\boldsymbol{y}_i - \mathbf{A}_i\left(\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i\right) + \left[\left(\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i\right)^{\mathrm{T}} \otimes \mathbf{I}_m\right]\mathbf{M}\breve{\boldsymbol{E}}_i - \mathbf{N}\breve{\boldsymbol{E}}_i\right\} = 0 \ . \tag{22}$$

Equations (20) and (21) immediately lead to

$$\breve{\boldsymbol{E}}_i = \mathbf{Q}\left(\mathbf{N} - \left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)^{\mathrm{T}} \tilde{\boldsymbol{\lambda}}_i \ , \tag{23}$$

$$\breve{\boldsymbol{\mu}}_i = \left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)\left(\mathbf{A}_i - \breve{\mathbf{E}}_{A_i}\right)^{\mathrm{T}} \tilde{\boldsymbol{\lambda}}_i \ . \tag{24}$$

By inserting Eqs (23) and (24) into Eq. (22), one obtains:

$$\begin{gathered}
\boldsymbol{y}_i - \mathbf{A}_i\left[\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)\left(\mathbf{A}_i - \breve{\mathbf{E}}_{A_i}\right)^{\mathrm{T}} \hat{\boldsymbol{\lambda}}_i + \hat{\boldsymbol{x}}_i\right] \\
+\left(\left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M} - \mathbf{N}\right)\mathbf{Q}\left(\mathbf{N} - \left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)^{\mathrm{T}} \tilde{\boldsymbol{\lambda}}_i = 0 \\
\rightarrow \\
\tilde{\boldsymbol{\lambda}}_i = \left(\left(\mathbf{N} - \left((\tilde{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)\mathbf{Q}\left(\mathbf{N} - \left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)^{\mathrm{T}}\right. \\
\left.+\mathbf{A}_i\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)\left(\mathbf{A}_i - \breve{\mathbf{E}}_{A_i}\right)^{\mathrm{T}}\right)^{-1}\left(\boldsymbol{y}_i - \mathbf{A}_i \hat{\boldsymbol{x}}_i\right).
\end{gathered} \tag{25}$$

Note that in Eq. (25), the inverse exists since the matrix $\mathbf{N} - \left((\tilde{\boldsymbol{\mu}}_i + \breve{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}$ is full row rank, i.e. its quadratic form is invertible. Also due to Eqs (13) and (24), the updated dispersion matrix of the predicted unknown $\underline{\boldsymbol{x}}_i$ is computed by applying variance propagation rules as follows:

$$\begin{gathered}
\mathbf{D}(\underline{\boldsymbol{x}}_i) = \mathbf{D}\left(\underline{\boldsymbol{\mu}}_i + \underline{\breve{\boldsymbol{x}}}_i\right) = \mathbf{D}\left(\underline{\boldsymbol{\mu}}_i\right) + \mathbf{D}(\underline{\breve{\boldsymbol{x}}}_i) \\
\rightarrow \\
\mathbf{D}(\underline{\boldsymbol{x}}_i) = \left(\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right) \otimes \tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}}\right)\mathbf{MQM}^{\mathrm{T}}\left(\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right) \otimes \tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}}\right)^{\mathrm{T}} + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}} \ .
\end{gathered} \tag{26}$$

Note that in *Schaffrin and Iz (2008)* no measurement update for the dispersion matrix of the predicted unknown was considered.

Summarizing the general WTKF algorithm is proposed as follows:

1st step: $\breve{\boldsymbol{E}}_i^0 = 0$; $\breve{\boldsymbol{\mu}}_i^0 = 0$; $\hat{\boldsymbol{x}}_i$.
2nd step: for an epoch $i \in N$ compute iteratively:

$$\begin{gathered}
\tilde{\boldsymbol{\lambda}}_i = \left(\left(\mathbf{N} - \left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)Q\left(\mathbf{N} - \left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)^{\mathrm{T}}\right. \\
\left.+\mathbf{A}_i\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)\left(\mathbf{A}_i - \breve{\mathbf{E}}_{A_i}\right)^{\mathrm{T}}\right)^{-1}\left(\boldsymbol{y}_i - \mathbf{A}_i \hat{\boldsymbol{x}}_i\right),
\end{gathered}$$

$$\breve{\boldsymbol{E}}_i = \mathbf{Q}\left(\mathbf{N} - \left((\breve{\boldsymbol{\mu}}_i + \hat{\boldsymbol{x}}_i)^{\mathrm{T}} \otimes \mathbf{I}_m\right)\mathbf{M}\right)^{\mathrm{T}} \tilde{\boldsymbol{\lambda}}_i ,$$

$$\breve{\boldsymbol{\mu}}_i = \left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right)\left(\mathbf{A}_i - \breve{\mathbf{E}}_{A_i}\right)^{\mathrm{T}} \tilde{\boldsymbol{\lambda}}_i .$$

3rd step: after one sees the convergence at the 2nd step for an epoch i, compute the state vector and its corresponding dispersion matrix at this epoch as follows:

$$\breve{\boldsymbol{x}}_i = \hat{\boldsymbol{x}}_i + \breve{\boldsymbol{\mu}}_i ,$$

$$\mathbf{D}\left(\underline{\boldsymbol{x}}_i\right) = \left(\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right) \otimes \tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}}\right)\mathbf{MQM}^{\mathrm{T}}\left(\left(\boldsymbol{\theta}_i + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}}\right) \otimes \tilde{\boldsymbol{\lambda}}_i^{\mathrm{T}}\right)^{\mathrm{T}} + \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1}^0 \boldsymbol{\Phi}_i^{\mathrm{T}} .$$

4th step: for the next epoch $i + 1$, set $\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{\Phi}_{i+1}\breve{\boldsymbol{x}}_i + f_i$, $\boldsymbol{\Sigma}_i^0 = \mathbf{D}\left(\underline{\boldsymbol{x}}_i\right)$, and repeat steps 1 to 3.

Note that in the 1st step, we put the initial values $\breve{\boldsymbol{E}}_i^0 = 0$ and $\breve{\boldsymbol{\mu}}_i^0 = 0$, since they are zero mean quantities.

4. NUMERICAL TESTS AND DISCUSSION

One of the applications of the developed theory is integrated navigation. In many applications in geodesy and surveying e.g. mobile mapping, we need to determine the position and attitude of a mobile sensor. In recent years there has been an explosion in the number, type and diversity of system designs and application areas of mobile mapping systems. However, generally speaking all of these systems typically share the following major components: a) navigation component, b) remote sensing component. The first component is produced by integrated navigation. If we want to use different sources of data produced by different sensors such as IMU, GPS and remote sensing sensors in order to obtain the attitude and position of the mobile sensor in some epochs, it is called integrated navigation. The integration of these data is done during a Kalman filter algorithm. Traditionally one could use classic filters such as EKF to solve this problem and to date, most navigation systems have relied mainly on the GPS receivers as the primary source of information to provide the position of the vehicle. GPS is able to provide precise positioning information to an unlimited number of users anywhere on the planet. GPS, however, can provide these types of information only under ideal conditions which require an open environment (i.e. open space areas). In other words, the system doesn't work very well in urban, canopy areas due to signal blockage and can be totally blocked if the signal is jammed (see, e.g., *Sheta, 2012* and *Stepanyan, 2006*).

Nevertheless, motivated by the new advances in remote sensing sensors solutions in combination with traditional navigation sensors, recently some new systems have been proposed based on fusing remote sensing measurements with GPS-INS measurements to achieve comprehensive, fast and real-time systems. The observation equations of these new developed systems given by Eq. (1) are in fact DEIV models and therefore a TKF algorithm such as the proposed WTKF or the TKF algorithm of *Schaffrin and Iz (2008)* is more proper. Furthermore, as the characteristic of the noise of different sensors is often arbitrary and different, the structure of the noise supposed by *Schaffrin and Iz (2008)* may not be useful. Therefore our developed algorithm is more applicable in such cases.

As it was mentioned by *Schaffrin and Iz (2008)*, quantitative studies are required to examine the TKF algorithm. For verifying the WTKF algorithm, here we apply both the TKF algorithm and our proposed WTKF algorithm to a DEIV model and compare the results.

Suppose that the dynamic equation is produced by an IMU system in a local frame as follows:

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \boldsymbol{f}_i \ , \tag{27}$$

with

$$\boldsymbol{f}_i = \begin{bmatrix} \boldsymbol{a}\left(t_{i+1} - t_i\right) \\ 0_{3\times 1} \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} 1.3 \\ 0.8 \\ 1.0 \end{bmatrix},$$

$$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{P}_i \\ \boldsymbol{R}_i \end{bmatrix}, \quad \boldsymbol{P}_i = \begin{bmatrix} x_i^{position} \\ y_i^{position} \\ z_i^{position} \end{bmatrix}, \quad \boldsymbol{R}_i = \begin{bmatrix} roll_i \\ pitch_i \\ azimuth_i \end{bmatrix}.$$

The stochastic model of this system equation is as follows:

$$\boldsymbol{\theta}_i = \begin{bmatrix} 2.9 & 3.4 & 1.0 & 0 & 0 & 0 \\ 3.4 & 6.0 & 3.2 & 0 & 0 & 0 \\ 1.0 & 3.2 & 2.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0025 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0025 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0025 \end{bmatrix}.$$

In Appendix A it is generally explained how these dispersion matrices are generated for a random quantity. The observed state vector $\boldsymbol{x}_1$ at an initial epoch with its corresponding dispersion matrix $\boldsymbol{\Sigma}_0^0$ are given by:

$$\boldsymbol{x}_1 = \begin{bmatrix} 7.86 \\ 8.74 \\ 14.4 \\ -0.08 \\ 0.28 \\ 0.07 \end{bmatrix}, \quad \boldsymbol{\Sigma}_0^0 = \begin{bmatrix} 5 & 5 & 1 & 0 & 0 & 0 \\ 5 & 11 & 4 & 0 & 0 & 0 \\ 1 & 4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

The observation equations of this example, which may be produced by GPS and remote sensed data, are given by 5 DEIV models at 5 epochs ($i = 1, 2, 3, 4, 5$), which can be found in Appendix B. Note that the chosen time interval is 5 seconds, i.e. $t = 5, 10, 15, 20, 25$. For all of the DEIV models, the stochastic model is given by

$$\mathbf{Q} = \mathrm{blkdiag}\left(\mathbf{Q}_x, \mathbf{Q}_y\right) , \tag{28}$$

with $\mathbf{Q}_x$ and $\mathbf{Q}_y$ as follows:

$$\mathbf{Q}_x = \left(\mathbf{I}_6 \otimes \mathbf{q}\right)\left(\mathbf{I}_6 \otimes \mathbf{q}\right)^{\mathrm{T}} , \quad \mathbf{q} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0.1 & 0 & 0.4 & 0.6 & 0.2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.1 \\ 0 & 1 & 1 & 0.9 & 0.2 & 0.7 & 0.1 & 0.2 & 0 \\ 0.6 & 0 & 1 & 1 & 1 & 1 & 0 & 0.3 & 0 \\ 0.6 & 0.1 & 1 & 0.6 & 0.1 & 1 & 0.6 & 0.1 & 1 \\ 0.2 & 0.3 & 0.4 & 0.6 & 0.1 & 1 & 0.6 & 0.1 & 1 \\ 0.4 & 0.7 & 0.8 & 0.4 & 0.2 & 0.4 & 0.6 & 0.1 & 1 \\ 0.1 & 0.3 & 0.5 & 0.6 & 0.6 & 0.4 & 0.3 & 0.2 & 1 \\ 0.6 & 0.1 & 1 & 0.6 & 0.1 & 1 & 0.6 & 0.1 & 1 \end{bmatrix} ,$$

$$\mathbf{Q}_y = \begin{bmatrix} 3.01 & 1.32 & 1.08 & 2.12 & 1.67 & 1.35 & 0.72 & 1.02 & 1.58 \\ 1.32 & 1.04 & 0.13 & 0.73 & 0.74 & 0.40 & 0.14 & 0.18 & 0.72 \\ 1.08 & 0.13 & 3.39 & 2.86 & 2.40 & 2.04 & 1.98 & 1.81 & 2.39 \\ 2.12 & 0.73 & 2.86 & 4.58 & 3.15 & 2.27 & 1.43 & 2.23 & 3.22 \\ 1.67 & 0.74 & 2.40 & 3.15 & 3.79 & 3.05 & 1.26 & 1.49 & 3.77 \\ 1.35 & 0.40 & 2.04 & 2.27 & 3.05 & 3.03 & 1.16 & 1.33 & 2.99 \\ 0.72 & 0.14 & 1.98 & 1.43 & 1.26 & 1.16 & 1.70 & 1.17 & 1.21 \\ 1.02 & 0.18 & 1.81 & 2.23 & 1.49 & 1.33 & 1.17 & 1.36 & 1.50 \\ 1.58 & 0.72 & 2.39 & 3.22 & 3.77 & 2.99 & 1.21 & 1.50 & 3.78 \end{bmatrix} .$$

For $i = 1, 2, 3, 4, 5$, the $(9i + 1)$-th to $(9i + 3)$-th rows and columns of the matrix $\mathbf{I}_6 \otimes \mathbf{q}$ must be replaced by zero.

We apply three algorithms - KF, TKF (*Schaffrin and Iz, 2008*), and WTKF, proposed here to this problem. The results are shown in Figs 1 and 2 and compared with the true solution. Also the predicted residuals of different methods for 5 epochs are shown in Fig. 3. As it is seen in Figs 1 and 2, the proposed WTKF approach can make the best improvement, i.e. approximately 25% improvement in the solution of the predicted position in contrast to other algorithms particularly when the magnitude of the weights of the elements in the random design matrix are large. Although the TKF algorithm of *Schaffrin and Iz (2008)* is mathematically correct and improvement of WTKF algorithm relating to TKF solution is not so high especially for rotation angles, it cannot completely satisfy the weight structure given by Eq. (27). Therefore, its predicted position differs

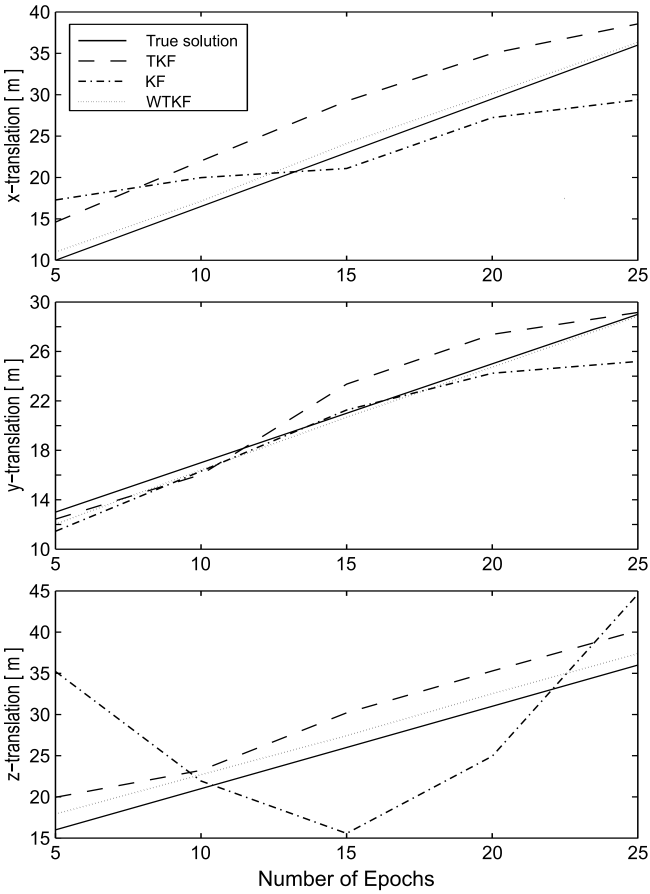

Fig. 1. The solution of different algorithms for 3-D position and attitude of the sensor in a local frame. KF: Kalman Filter; TKF: Total Kalman Filter; WTKF: Weighted Total Kalman Filter.

from the position predicted by the WTKF algorithm and the true solution. This difference is increased when the magnitude of the weights of the observed quantities are lager which prove the important role of the weight matrixes. Moreover, the magnitude of the bias will be increased at the later epochs in both of the KF algorithm and the TKF algorithm of *Schaffrin and Iz (2008)*. Another point of interest refers to the predicted position and attitude of the classic KF algorithm. Not only its solution considerably differs from the true solution, but also it has an irregular treatment since this algorithm does not consider the random property of the design matrix within the DEIV model. Furthermore, it can be shown that if one considers the homoscedastic case, both of the TKF algorithm proposed by *Schaffrin and Iz (2008)* and the WTKF algorithm give the same results. Also, if the coefficient matrix of the observation model has no random error, the results of three algorithms KF, TKF and WTKF are the same.

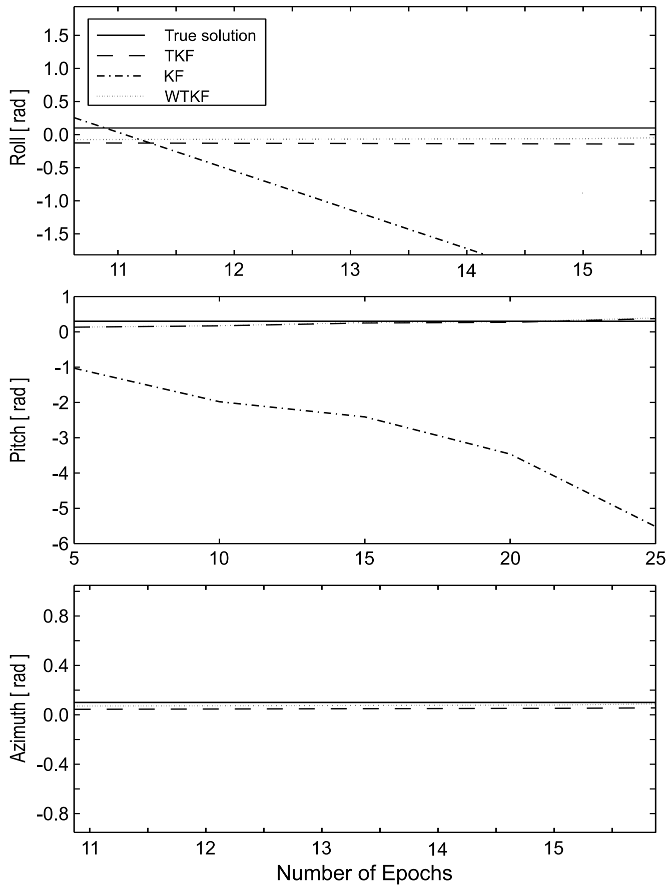

Fig. 2. The same as in Fig. 1, but for the attitude of the sensor in a local frame.

All of the above results are approved by Fig 3 where the minimum amount of predicted residuals were obtained by the WTKF algorithm. Note that all of the positions are in meters and angles are in radians.

5. CONCLUSIONS

We developed a general Weighted Total Kalman filter (WTKF) algorithm. In contrast to the TKF approach proposed by *Schaffrin and Iz (2008)*, our algorithm considers an arbitrary structure for the weight matrixes of the random observable quantities in addition to the correlation among them. We proposed this algorithm in a way such that it can be applicable to the numerical dynamic problems which appear in the dynamic errors-in-variables (DEIV) models. As the numerical examples demonstrate, the proposed WTKF approach can make the best improvement compared to the existing KF algorithms since

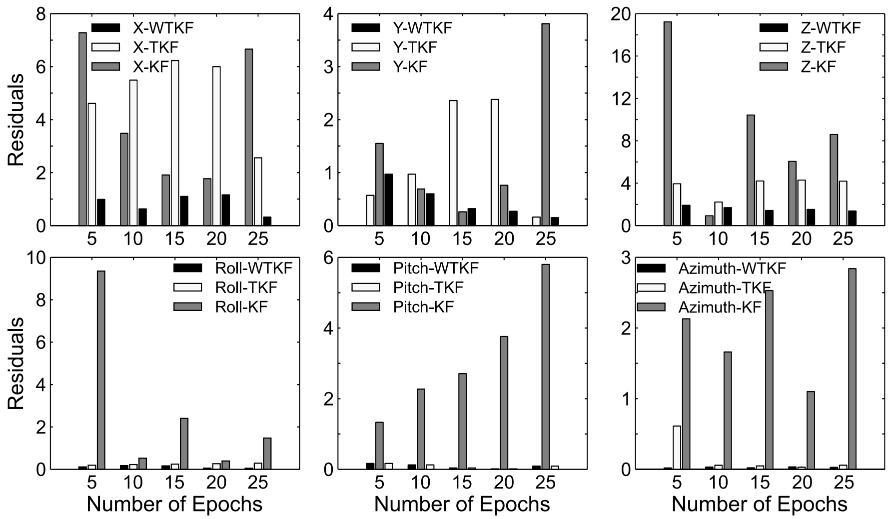

Fig. 3. The predicted residuals of different algorithms in the local frame for 5 epochs. KF: Kalman Filter; TKF: Total Kalman Filter; WTKF: Weighted Total Kalman Filter.

not only it considers the random coefficient matrix in the DEIV model but also a general structure for the stochastic model can be satisfied with this algorithm. Also the TKF algorithm of *Schaffrin and IZ (2008)* was numerically examined here for the first time. We leave the other aspects of the WTKF algorithm for the future contributions.

APPENDIX A
PRODUCTION OF DISPERSION MATRICES

For producing the dispersion matrix $\mathbf{Q}_d$ of a random vector $\underline{\boldsymbol{d}}$, which is corrupted by the zero mean white noise $\underline{\boldsymbol{n}}$ such that

$$\underline{\boldsymbol{d}} = \mathbf{E}\left(\underline{\boldsymbol{d}}\right) + \mathbf{q} \cdot \underline{\boldsymbol{n}} \ ,$$

$\mathbf{Q}_d$ is given as follows:

$$\mathbf{Q}_d = \mathbf{q} \cdot \mathbf{q}^{\mathrm{T}} \ ,$$

where $\mathbf{q}$ is the deterministic matrix which determines the structure of the noise.

APPENDIX B
OBSERVATION VECTORS AND THE CORRESPONDING DESIGN MATRICES OF THE OBSERVATION EQUATIONS

$$\boldsymbol{y}_1 = \begin{bmatrix} 7.82 & 11.8 & 19.3 & -8.4 & -37 & 89.9 & 0.08 & -33 & 121 \end{bmatrix},$$

$$\boldsymbol{y}_2 = \begin{bmatrix} 14.2 & 15.2 & 21.8 & 6.97 & -132 & 128 & 24.8 & -37.5 & 157 \end{bmatrix},$$

$$\boldsymbol{y}_3 = \begin{bmatrix} 25.523 & 21.69 & 26.15 & 41.537 & -262.67 & 172.63 & 50.492 & -41.427 & 195.99 \end{bmatrix},$$

$$\boldsymbol{y}_4 = \begin{bmatrix} 28.27 & 24.36 & 34.82 & 90.96 & -432 & 228.6 & 80.97 & -44.3 & 236.9 \end{bmatrix},$$

$$\boldsymbol{y}_5 = \begin{bmatrix} 32.707 & 27.549 & 35.058 & 149.51 & -650.16 & 287.02 & 106 & -50.29 & 269.17 \end{bmatrix},$$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 2.2897 & 0.329 & -2.1843 & 0.882 & -0.2163 & -1.1619 \\ -1.0252 & -6.4313 & -0.680 & 2.1 & 1.718 & 4.2552 \\ -4.0252 & -0.4312 & 4.3197 & -2.9 & 0.7185 & 1.255 \\ 5.945 & -8.325 & -0.979 & 0.420 & 1.065 & 3.683 \\ 0.846 & -4.041 & -2.874 & 9.015 & -0.460 & 0.9134 \\ -1.9252 & 1.5687 & 3.319 & -3.9 & 0.718 & 1.255 \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 4.564 & 1.30 & -3.039 & 1.068 & 5.784 & 3.95 \\ 0.632 & -7.478 & 1.252 & -0.269 & 7.46 & 6.614 \\ -2.36 & 4.521 & 6.252 & -8.269 & 3.46 & 3.614 \\ 6.393 & -4.690 & 0.4162 & -1.94 & 2.178 & 7.131 \\ 0.23 & -0.020 & -1.080 & 8.1352 & 2.049 & 2.292 \\ -0.267 & 5.521 & 5.252 & -6.269 & 3.46 & 4.614 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 4.584 & 0.892 & -3.85 & -2.377 & 3.020 & 1.036 \\ 2.508 & -13.73 & 1.342 & 1.121 & 8.760 & 4.22 \\ -0.491 & 4.269 & 6.342 & -9.878 & 1.760 & 1.223 \\ 8.733 & -5.631 & 0.851 & 0.0870 & 1.50 & 9.682 \\ 1.482 & -1.262 & -0.755 & 8.632 & 0.220 & 1.893 \\ 1.608 & 4.269 & 5.342 & -4.878 & 1.760 & 3.22 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 8.118 & -1.768 & -2.702 & 0.210 & 4.332 & 0.412 \\ 1.90 & -20.20 & 2.511 & -0.7763 & 13.41 & 6.022 \\ -1.099 & 3.798 & 7.511 & -14.77 & 3.417 & 3.022 \\ 8.15 & -7.33 & 1.7683 & -3.103 & 1.192 & 13.73 \\ 1.668 & -3.613 & 0.0826 & 6.417 & -0.0156 & 1.740 \\ 1.000 & 2.798 & 6.511 & -6.7764 & 3.417 & 6.022 \end{bmatrix}$$

$$\mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 11.75 & 2.089 & -4.201 & -2.72 & 1.55 & 1.549 \\ 1.448 & -25.266 & 1.717 & 1.175 & 16.73 & 3.332 \\ -1.551 & 4.734 & 6.717 & -15.825 & 3.73 & 0.332 \\ 7.754 & -7.589 & 0.599 & -0.0959 & 2.246 & 13.765 \\ 2.265 & -2.522 & -1.1247 & 7.969 & -0.0498 & -0.281 \\ 0.5481 & 2.734 & 5.717 & -4.825 & 3.73 & 4.332 \end{bmatrix}$$

Acknowledgements: We would like to thank two anonymous reviewers for their constructive comments.

References

Doucet A., de Freitas N. and Gordon N., 2001. *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag, Berlin, Germany.

Evensen G., 1994. Sequential data assimilation with a nonlinear quasigeostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, **99**, 10143−10162.

Evensen G., 1997. Advanced data assimilation for strongly nonlinear dynamics. *Mon. Weather Rev.*, **125**, 1342−1354.

Evensen G., 2007. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag, Berlin, Germany.

Fang X., 2011. *Weighted Total Least Squares Solutions for Applications in Geodesy*. Ph.D. Thesis. Leibniz University, Hannover, Germany.

Fang X., 2014. A structured and constrained Total Least-Squares solution with cross-covariances. *Stud. Geophys. Geod.*, **58**, 1−16.

Gelb A., 1974. *Applied Optimal Estimation*. MIT Press, Cambridge, MA.

Jazwinski A.H., 1970. *Stochastic Process and Filtering Theory*. Academic Press, New York.

Julier S.J. and Uhlmann J.K., 1997. New extension of the Kalman filter to nonlinear systems. *SPIE Proceedings* **3068**, *Signal Processing, Sensor Fusion, and Target Recognition VI, (28 July 1997)*, DOI: 10.1117/12.280797.

Julier S.J., Uhlmann J.K. and Durrant-Whyte H., 1995. A new approach for filtering nonlinear systems. *Proceedings of the 1995 American Control Conference*, 1628−1632, DOI 10.1109/ACC.1995.529783.

Kalman R.E., 1960. A new approach to linear filtering and prediction problems. *Trans. ASME - D, J. Basic Eng.*, **82**, 34–45.

Lefebvre T., Bruyninckx H. and De Schutter J., 2002. Comment on "A new method for the nonlinear transformation of means and covariances in filters and estimators". *IEEE Trans. Autom. Control*, **47**, 1406–1408.

Liu J. and Chen R., 1998. Sequential Monte Carlo methods for dynamic systems. *J. Am. Stat. Assoc.*, **93**, 1032–1044.

Magnus J., 1988. *Linear Structures*. Oxford University Press, Oxford, U.K.

Mahboub V., 2012. On weighted total least-squares for geodetic transformations. *J. Geodesy*, **86**, 359−367, DOI: 10.1007/s00190-011-0524-5.

Mahboub V., 2014. Variance component estimation in errors-in-variables models and a rigorous total least squares approach. *Stud. Geophys. Geod.*, **58**, 17–40, DOI: 10.1007/s11200-013-1150-x.

Mahboub V., 2016. A weighted least-squares solution to a 3-D symmetrical similarity transformation without linearization. *Stud. Geophys. Geod.*, **60**, 195–209, DOI: 10.1007/s11200-014-1041-9.

Mahboub V., Amiri-Simkooei A. and Sharifi M.A., 2012. Iterativelyreweighted total least-squares: A robust estimation in errors-in-variables models. *Surv. Rev.*, **45**, 92−99, DOI: 10.1179/1752270612Y.0000000017.

Mahboub V., Ardalan A.A. and Ebrahimzadeh S., 2015. Adjustment of non-typical errors-in37 variables models. *Acta Geod. Geophys.*, **50**, 207–218, DOI 10.1007/s40328-015-0109-5.

Mahboub V. and Sharifi M.A., 2013. On weighted total least-squares with linear and quadratic constraints. *J. Geodesy*, **87**, 279–286, DOI: 10.1007/s00190-012-0598-8.

Maybeck P., 1979. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, Waltham, MA.

Neitzel F., 2010. Generalization of total least-squares 1 on example of unweighted and weighted similarity transformation. *J. Geodesy*, **84**, 751−762.

Neitzel F. and Schaffrin B., 2016. On the Gauss-Helmert model with a singular dispersion matrixwhere BQ is of smaller rank than B. *J. Comput. Appl. Math.*, **291**, 458–467, DOI: 10.1016/j.cam.2015.03.006.

Sanso F., 1986. Statistical methods in physical geodesy. In: Sunkel H. (Ed.), *Mathematical and Numerical Techniques in Physical Geodesy*. Lecture Notes in Earth Sciences, **7**. Springer-Verlag, Berlin, Germany, 49−155.

Schaffrin B. and Felus Y., 2008. On the multivariate total least-squares approach to empirical coordinate transformations. Three algorithms. *J. Geodesy*, **82**, 353–383.

Schaffrin B. and Iz H.B., 2008. Towards Total Kalman filtering for mobile mapping. In: Vettore A. and El-Sheimy N. (Eds), *The 5th International Symposium on Mobile Mapping Technology*. ISPRS Archives, XXXVI-5/C55, 270–275 (http://www.isprs.org/proceedings/XXXVI/5-C55/papers/schaffrin_burkhard.pdf).

Schaffrin B., Snow K. and Neitzel F. 2014. On the Errors-In-Variables model with singular dispersion matrices. *J. Geod. Sci.*, **4**, 28–36, DOI: 10.2478/jogs-2014-0004.

Schaffrin B. and Uzun S., 2011. Errors-InVvariables for mobile mapping algorithms in the presence of outliers. *Arch. Photogramm. Cartogr. Remote Sens.*, **22**, 377–387.

Sheta B., 2012. *Vision Based Navigation (VBN) of Unmanned Aerial Vehicles (UAV)*. PhD Thesis. University of Alberta, Calgary, Alberta, Canada.

Shen Y.Z., Li B.F. and Chen Y., 2011. An iterative solution of weighted total least squares adjustment. *J. Geodesy*, **85**, 229–238.

Sorenson H.W., 1966. Kalman filtering techniques. In: Leondes C.T. (Ed.), *Advances in Control Systems: Theory and Applications*. Volume 3. Academic Press, New York, 219–292.

Stepanyan V., 2006. Vision based guidance and flight control in problems of aerial tracking. http://scholar.lib.vt.edu/theses/available/etd-08142006-151453.

Teunissen P.J.G. and Khodabandeh A., 2013. BLUE, BLUP and the Kalman filter: some new results. *J. Geodesy*, **87**, 461–473, DOI: 10.1007/s00190-013-0623-6.

van der Merwe R. and Wan E.A., 2004. Sigma-point Kalman filters for integrated navigation. *Proceedings of the 60th Annual Meeting of The Institute of Navigation (2004), Dayton, OH, June 2004*. The Institute of Navigation, Manassas, VA, 641−654.

Xu P., Liu J. and Shi C., 2012. Total least squares adjustment in partial errors-in-variablesmodels: algorithm and statistical analysis. *J. Geodesy*, **86**, 661–675.

Yi Y., 2007. *On Improving the Accuracy and Reliability of GPS/Ins-Based Direct Sensor Georeferencing*. PhD Thesis. The Ohio State University, Columbus, OH.