Tracking a target moving in a known road network by square-root equality-constrained filtering

L. Chisci[1], A. Mavino[1], F. Papi[1], G. Perferi[1], A. Benavoli[2], and A. Farina[2]
[1] DSI, Università di Firenze, Florence, Italy
e-mail: *chisci@dsi.unifi.it*
[2] Engineering Division, SELEX Sistemi Integrati, Rome, Italy
e-mail: {*abenavoli,afarina*}*@selex-si.com*

***Abstract*— In this paper, a square-root equality-constrained linear filter is used to track a target moving along a road. This choice provides a computationally efficient, algorithmically simple and numerically robust solution for the implementation of a VS-DMM (Variable Structure - Dynamic Multiple Model) filter. A performance evaluation via Monte Carlo simulations shows the effectiveness of the proposed approach.**

I. Introduction

Tracking ground targets, based on measurements provided by a GMTI (Ground Moving Target Indicator) radar, is very critical due to the high target and clutter density as well as to the low and location-dependent detection probability encountered in ground scenarios. For this reason, it becomes of paramount importance to fully exploit the available a priori information (e.g. road and visibility maps) both in the data association and in the filtering tasks so as to improve tracking performance.

This paper is specifically devoted to the exploitation of road map information in the kinematic state estimation of a ground moving target. This topic has been recently addressed by several authors [1]-[4]. In all references, road map information is managed by means of a VS-DMM (Variable Structure - Dynamic Multiple Model) approach wherein different road-models and associated road-filters are used to take care of the possible hypotheses in proximity of the road junctions. Conversely, different options have been proposed for the road-filter e.g. a nonlinear particle filter in [1], a gaussian-sum filter in [2] and a linear equality-constrained filter based on a projection method in [4].

In this paper, a square-root equality-constrained linear filter is used to track a target moving along a road. This choice provides a computationally efficient, algorithmically simple and numerically robust solution for the VS-DMM filter bank.

II. Generalized Information Arrays

Hereafter the notation $\mathbf{v} \sim (\mathbf{m}, \mathbf{Q})$ will be used to denote that $\mathbf{v}$ is a random vector of mean $\mathbf{m}$ and covariance $\mathbf{Q}$. For the purpose of linear estimation, all the useful information on a random vector $\mathbf{x}$ can be represented by the following linear model:

$$\begin{cases} \mathbf{b} &= \mathbf{S}\mathbf{x} + \mathbf{U}\mathbf{v} \\ \mathbf{v} &\sim (\mathbf{0}, \mathbf{I}) \end{cases} \qquad \text{(II.1)}$$

where: $\mathbf{b}$ is a vector and $\mathbf{S}, \mathbf{U}$ are matrices of compatible dimensions; $\mathbf{v}$ is the so called noise vector which can be assumed, without loss of generality, to have zero mean and unit covariance. For consistency, $\mathbf{b}$ must belong to the range space of the matrix $[\mathbf{S},\ \mathbf{U}]$, i.e. $\mathbf{b} = [\mathbf{S},\ \mathbf{U}]\ \mathbf{z}$ for some vector $\mathbf{z}$. The triplet $(\mathbf{S}, \mathbf{U}, \mathbf{b})$ will be called *Generalized Information Array (GIA)*. From the GIA $(\mathbf{S}, \mathbf{U}, \mathbf{b})$, the *BLUE = Best Linear Unbiased Estimate* $\hat{\mathbf{x}}$ of $\mathbf{x}$ and its covariance $\mathbf{P} \stackrel{\triangle}{=} E\left[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})'\right]$ can easily be obtained. In particular, if $\mathbf{S}$ is square and non-singular:

$$\hat{\mathbf{x}} = \mathbf{S}^{-1}\mathbf{b}, \qquad \mathbf{P} = \mathbf{S}^{-1}\mathbf{U}\mathbf{U}'(\mathbf{S}^{-1})' \qquad \text{(II.2)}$$

There exist infinitely many equivalent representations (II.1). In fact, given (II.1), an equivalent representation is obtained by left-multiplication of (II.1) by an arbitrary non singular matrix and by transformation of the noise vector by an arbitrary orthogonal matrix.

More precisely, given a GIA $(\mathbf{S}, \mathbf{U}, \mathbf{b})$, an equivalent one $(\overline{\mathbf{S}}, \overline{\mathbf{U}}, \overline{\mathbf{b}})$ is obtained by

$$\overline{\mathbf{S}} = \mathbf{\Theta}S,\ \overline{\mathbf{U}} = \mathbf{\Theta}\mathbf{U}\mathbf{\Psi},\ \overline{\mathbf{b}} = \mathbf{\Theta}\mathbf{b}$$

where $\mathbf{\Theta}$ is an arbitrary square and non-singular matrix and $\mathbf{\Psi}$ is an arbitrary (not necessarily square) matrix with orthogonal columns. In other words, an equivalent GIA can be obtained from $(\mathbf{S}, \mathbf{U}, \mathbf{b})$ by arbitrary linear non-singular transformations on the rows of $[\mathbf{S},\ \mathbf{U},\ \mathbf{b}]$ and arbitrary orthogonal transformations on the columns of $\mathbf{U}$.

A GIA is said:

- *full row rank* if $\mathbf{S}$ is full row rank;
- *diagonal* if $\mathbf{U}$ is diagonal (i.e. square and zero off the main diagonal);
- *canonical* if $\mathbf{S}$ is upper triangular (i.e. zero below the main diagonal) and $\mathbf{U}$ is diagonal.

Without loss of generality, a GIA can be cast in canonical form. In fact, for any GIA$(\mathbf{S}, \mathbf{U}, \mathbf{b})$ there always exists an equivalent canonical GIA $(\overline{\mathbf{S}}, \overline{\mathbf{U}}, \overline{\mathbf{b}})$ which can be obtained by a simple algorithmic procedure consisting of the following two steps:

1) **Singular Value Decomposition (SVD)** of $\mathbf{U}$ - Let

$$\mathbf{\Theta}_1\mathbf{U}\mathbf{\Psi}_1 = \mathbf{\Sigma} = diag\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$$

be the SVD of the $m \times n$ matrix $\mathbf{U}$ with $\mathbf{\Theta}_1$, $m \times m$, orthogonal, and $\mathbf{\Psi}_1$, $n \times m$, such that

$$\begin{cases} \mathbf{\Psi}_1 \mathbf{\Psi}_1' = \mathbf{I}_n, & \text{if } n \leq m \\ \mathbf{\Psi}_1' \mathbf{\Psi}_1 = \mathbf{I}_m, & \text{if } n > m \end{cases}$$

Then $(\mathbf{\Theta}_1 \mathbf{S}, \mathbf{\Sigma}, \mathbf{\Theta}_1 \mathbf{b})$ is a diagonal GIA equivalent to $(\mathbf{S}, \mathbf{U}, \mathbf{b})$.

2) **Triangularization** - Let $\mathbf{\Theta}_2$, non-singular, and $\mathbf{\Psi}_2$, orthogonal, be such that $\mathbf{\Theta}_2 \mathbf{\Theta}_1 \mathbf{S}$ is upper triangular and $\mathbf{\Theta_2 \Sigma \Psi}_2$ remains diagonal. Then $(\mathbf{\Theta}_2 \mathbf{\Theta}_1 \mathbf{S},\ \mathbf{\Theta}_2 \mathbf{\Theta}_1 \mathbf{U} \mathbf{\Psi}_1 \mathbf{\Psi}_2,\ \mathbf{\Theta}_2 \mathbf{\Theta}_1 \mathbf{b})$ is a canonical GIA equivalent to $(\mathbf{S}, \mathbf{U}, \mathbf{b})$.

The triangularization procedure is described in detail hereafter. This procedure is based solely on elementary row operations. The details are best explained at the 2×2 level, i.e. let

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots \\ s_{21} & \cdots \end{bmatrix}, \mathbf{U} = \begin{bmatrix} u_1 & 0 \\ 0 & u_2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

be a diagonal GIA with two rows. The objective is to upper triangularize $\mathbf{S}$, preserving $\mathbf{U}$ diagonal: more precisely, the aim is to find $\mathbf{\Theta}$, 2×2 non-singular, and $\mathbf{\Psi}$, 2×2 orthogonal, such that

$$\begin{aligned} \overline{\mathbf{S}} &= \mathbf{\Theta S} = \begin{bmatrix} \overline{s}_{11} & \overline{s}_{12} & \cdots \\ 0 & \overline{s}_{22} & \cdots \end{bmatrix}, \\ \overline{\mathbf{U}} &= \mathbf{\Theta U \Psi} = \begin{bmatrix} \overline{u}_1 & 0 \\ 0 & \overline{u}_2 \end{bmatrix}, \quad \overline{\mathbf{b}} = \mathbf{\Theta b} \end{aligned} \tag{II.3}$$

It is assumed, without loss of generality that $s_{11} s_{21} \neq 0$, otherwise (II.3) can be performed in a trivial way, by no operation if $s_{21} = 0$ or by an interchange if $s_{11} = 0$. It is easy to check that, if $s_{11} s_{21} \neq 0$, to carry out (II.3) one can possibly interchange u_1 and u_2 as well as $[\mathbf{s}_1, b_1]$ and $[\mathbf{s}_2, b_2]$ to ensure that $u_1^2 < u_2^2$ and then choose

$$\begin{aligned} \mathbf{\Theta} &= \frac{1}{\sqrt{s_{11}^2 + \alpha^2 s_{21}^2}} \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} s_{11} & \alpha^2 s_{21} \\ -s_{21} & s_{11} \end{bmatrix} \\ \mathbf{\Psi} &= \frac{1}{\sqrt{s_{11}^2 + \alpha^2 s_{21}^2}} \begin{bmatrix} s_{11} & -\alpha s_{21} \\ \alpha s_{21} & s_{11} \end{bmatrix} \end{aligned} \tag{II.4}$$

where $\gamma_i > 0$ are arbitrary scaling factors and

$$\alpha \stackrel{\triangle}{=} \begin{cases} 1, & \text{if } u_1 = u_2 \\ u_1/u_2, & \text{otherwise} \end{cases} \tag{II.5}$$

Notice that it turns out that $\overline{\mathbf{U}} = \mathbf{\Gamma U}$ and, in particular, $\overline{\mathbf{U}} = \mathbf{U}$ if the scaling factors γ_i are chosen unitary. Hence, no transformations on $\mathbf{U}$ need actually be applied except possible interchanges of diagonal components. In order to carry out the triangularization step, elementary transformations of the form (II.3)-(II.5) can be applied several times to induce zeros in prescribed positions of the $\mathbf{S}$ matrix.

Remark (Connection with Givens rotations) - Notice that the elementary transformation (II.3)-(II.4) reduces to a Givens rotation on $\mathbf{S}$ and no transformation on $\mathbf{U}$ whenever $\alpha = \gamma_1 = \gamma_2 = 1$. Provided that $\mathbf{x}$ has non singular covariance $\mathbf{P}$, the matrix $\mathbf{U}$ can be taken, without loss of generality, equal to the identity by suitably scaling the rows of $\mathbf{S}$. In this case, $\mathbf{S}$ can be triangularized by exploiting Givens rotations only. ■

III. Linear equality-constrained filtering

Consider the linear discrete-time system:

$$\begin{cases} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t) \end{cases} \tag{III.1}$$

subject to linear constraints

$$\mathbf{M}\mathbf{x}(t) = \mathbf{d} \tag{III.2}$$

where: $\mathbf{w}(\cdot)$ and $\mathbf{v}(\cdot)$ are independent zero-mean white noises of variances $\mathbf{Q}$ and, respectively, $\mathbf{R}$; $\mathbf{M}$ is a full row-rank matrix. Notice that the matrices $\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \mathbf{M}$ and the vector $\mathbf{d}$ can be time-varying although, for simpler notation, the dependence of such quantities from the time index t will be omitted. Combining (III.1) and (III.2), the following representation for the joint vector $[\mathbf{x}'(t), \mathbf{x}'(t+1)]'$ is obtained:

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \\ \mathbf{A} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t+1) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{v}(t) \\ \mathbf{w}(t) \end{bmatrix}}_{\mathbf{n}(t)} = \begin{bmatrix} \mathbf{d} \\ \mathbf{y}(t) \\ \mathbf{0} \end{bmatrix} \tag{III.3}$$

Notice that the noise term $\mathbf{n}(t)$ in (III.3) has zero mean and block-diagonal covariance $diag\{\mathbf{0}, \mathbf{R}, \mathbf{Q}\}$ and can be equivalently expressed, up to second-order statistics, as

$$\mathbf{n}(t) = \begin{bmatrix} \mathbf{0} & & \\ & \mathbf{\Theta}_R' \mathbf{\Sigma}_R^{\frac{1}{2}} & \\ & & \mathbf{\Theta}_Q' \mathbf{\Sigma}_Q^{\frac{1}{2}} \end{bmatrix} \boldsymbol{\xi}(t) \tag{III.4}$$

where $\xi(t) \sim (\mathbf{0}, \mathbf{I})$ and

$$\mathbf{\Theta}_R\ \mathbf{R}\ \mathbf{\Theta}_R' = \mathbf{\Sigma}_R, \quad \mathbf{\Theta}_Q\ \boldsymbol{Q}\ \mathbf{\Theta}_Q' = \mathbf{\Sigma}_Q \tag{III.5}$$

are the *Singular Value Decompositions (SVDs)* of $\mathbf{R}$ and, respectively, $\mathbf{Q}$. Exploiting (III.4) in (III.3) and left-multiplying both sides by the non-singular matrix $diag\{\mathbf{0}, \mathbf{\Theta}_R, \mathbf{\Theta}_Q\}$ the following diagonal GIA for the joint vector $[\mathbf{x}'(t), \mathbf{x}'(t+1)]'$ is obtained:

$$\begin{aligned} &\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{\Theta}_R \mathbf{C} & \mathbf{0} \\ \mathbf{\Theta}_Q \mathbf{A} & -\mathbf{\Theta}_Q \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t+1) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0} & & \\ & \mathbf{\Sigma}_R^{\frac{1}{2}} & \\ & & \mathbf{\Sigma}_Q^{\frac{1}{2}} \end{bmatrix} \boldsymbol{\xi}(t) = \begin{bmatrix} \mathbf{d} \\ \mathbf{\Theta}_R \mathbf{y}(t) \\ \mathbf{0} \end{bmatrix} \end{aligned} \tag{III.6}$$

Given initial conditions $\hat{\mathbf{x}}(1|0)$ and $\mathbf{P}(1|0)$, the prior information can be summarized by the initial GIA $(\mathbf{S}(1|0), \mathbf{U}(1|0), \mathbf{b}(1|0))$ defined as follows:

$$\begin{cases} \mathbf{U}(1|0) &= \mathbf{I} \\ \mathbf{S}(1|0) &: \mathbf{S}'(1|0)\ \mathbf{S}(1|0) = \mathbf{P}^{-1}(1|0) \\ \mathbf{b}(1|0) &= \mathbf{S}(1|0)\ \hat{\mathbf{x}}(1|0) \end{cases} \tag{III.7}$$

Then, exploiting the GIA representation, the recursive state filtering problem can be interpreted as follows: at time t, given the a-priori canonical GIA $(\mathbf{S}(t|t-1), \mathbf{U}(t|t-1), \mathbf{b}(t|t-1))$ for the state $\mathbf{x}(t)$ based on information up to time $t-1$, carry out the following two steps.

1. Measurement update (correction):

$$(\mathbf{S}(t|t-1), \mathbf{U}(t|t-1), \mathbf{b}(t|t-1)) \rightarrow (\mathbf{S}(t|t), \mathbf{U}(t|t), \mathbf{b}(t|t))$$

By means of the algorithmic procedure described in section 2, perform the GIA triangularization:

$$\left[\begin{array}{c|c|c} \mathbf{S}(t|t-1) & \mathbf{U}(t|t-1) & \mathbf{b}(t|t-1) \\ \mathbf{M} & \mathbf{0} & \mathbf{d} \\ \mathbf{\Theta}_R\mathbf{C} & \mathbf{\Sigma}_R^{\frac{1}{2}} & \mathbf{\Theta}_R\mathbf{y}(t) \end{array}\right] \longrightarrow \left[\begin{array}{c|c|c} \mathbf{S}(t|t) & \mathbf{U}(t|t) & \mathbf{b}(t|t) \\ \mathbf{0} & * & * \\ \mathbf{0} & * & * \end{array}\right] \tag{III.8}$$

where $*$ denote don't care blocks.

2. Time update (prediction):

$$(\mathbf{S}(t|t), \mathbf{U}(t|t), \mathbf{b}(t|t)) \rightarrow (\mathbf{S}(t+1|t), \mathbf{U}(t+1|t), \mathbf{b}(t+1|t))$$

In a similar way, perform the GIA triangularization:

$$\left[\begin{array}{cc|c|c} \mathbf{S}(t|t) & \mathbf{0} & \mathbf{U}(t|t) & \mathbf{b}(t|t) \\ \mathbf{\Theta}_Q\mathbf{A} & -\mathbf{\Theta}_Q & \mathbf{\Sigma}_Q^{\frac{1}{2}} & \mathbf{0} \end{array}\right] \longrightarrow \left[\begin{array}{cc|c|c} * & * & * & * \\ \mathbf{0} & \mathbf{S}(t+1|t) & \mathbf{U}(t+1|t) & \mathbf{b}(t+1|t) \end{array}\right] \tag{III.9}$$

Remark - By virtue of Theorem 2 - part 2 in [5], if $\mathbf{S}(1|0)$ is non-singular then $\mathbf{S}(t|t)$ and $\mathbf{S}(t+1|t)$ turn out to be non-singular for all $t > 0$, i.e. the state is always estimable. ■

The recursive algorithm defined via the initialization (III.7) at time $t = 0$ and the two triangularizations (III.8) and (III.9) performed for $t = 1, 2, 3, \ldots$ will be referred to as *Generalized Square-Root Information Filter (GSRIF)* adopting the terminology in [5]. On one hand, this algorithm is a generalization of the square-root information filter of Paige & Saunders [6] to account for the singularity of the covariance matrix implied by the presence of equality constraints, which can be regarded as noise-free observations. On the other hand, it is a specialization to linear equality-constrained filtering of the square-root filter proposed in [5] for more general linear models, possibly including descriptor dynamics and singular information cases. The GSRIF (III.7)-(III.9) provides also a square-root implementation of the equality-constrained linear filter proposed in [7].

Remark (Computational complexity) - With reference to the linear-equality constrained system (III.1)-(III.2) let n, p, r denote respectively the state dimension, measurement dimension and number of independent constraints (rows of M). Then it is easy to check that the GSRIF requires, at each time step: $\frac{1}{6}\left(11n^3 + 3n^2 - 2n\right)$ elementary transformations for the time update; $\frac{1}{2}(p+r)\,n\,(n+1)$ elementary transformations for the measurement update; $\frac{1}{2}n\,(n+1)$ multiplications and additions for the determination of the estimate vector $\hat{\mathbf{x}} = \mathbf{S}^{-1}\mathbf{b}$ by back-substitution. Since there exist fast elementary transformations requiring only two multiplications and additions, the total operation count amounts to $\frac{11}{3}n^3 + \left(p + r + \frac{3}{2}\right)n^2 + \left(p + r - \frac{1}{6}\right)n$ flops, where 1 flop stands for 1 multiplication plus 1 addition. Notice that the above operation count assumes time-invariant $\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}$ while the time-variance of such matrices calls for extra on-line computations. An alternative efficient solution to linear equality constrained filtering is the approach proposed in [7] which essentially involves a conventional (i.e. unconstrained and non square-root) Kalman filter with a suitably defined $n \times n$ process covariance matrix $\mathbf{Q}$ such that $\mathbf{w}(t) \sim (\mathbf{0}, \mathbf{Q})$ implies $\mathbf{M}\mathbf{w}(t) = \mathbf{0}$. As reported from [8], an efficient implementation of the conventional Kalman filter that exploits matrix symmetry, requires $3n^3 + 3pn^2 + \frac{3}{2}p^2n + \frac{1}{6}(p^3 - p)$ flops. Assuming as usual that n is larger than p and r, the asymptotic (for large n) complexity of GSRIF is, therefore, $O\left(\frac{11}{3}n^3\right)$ compared to $O\left(3n^3\right)$ of the conventional *Constrained Kalman Filter (CKF)* in [7]. Hence, the square-root implementation implies a negligible increase of complexity of a factor $\frac{11}{9} = 1.\bar{2}$. On the other hand, the GSRIF provides enhanced numerical robustness as well as the guarantee of symmetry and non-negative definiteness of the covariance matrix. ■

IV. Tracking a target moving along a known road

A. Straight-line road model

Let us first consider a straight-line road with equation

$$\alpha x + \beta y \;=\; \gamma \tag{IV.1}$$

in cartesian coordinates x and y. In order to model a target constrained to move along this road, one can adopt for instance a 2-D *continuous white noise acceleration (CWNA)* kinematic model with 4-dimensional state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$ where x, y and $\dot{x}, \dot{y}$ are the cartesian position and, respectively, velocity components. The resulting state-transition and process-noise covariance matrices turn out to be:

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{Q} = \frac{\sigma_s^2}{\alpha^2+\beta^2} \begin{bmatrix} \frac{\beta^2T^3}{3} & \frac{\beta^2T^2}{2} & -\frac{\alpha\beta T^3}{3} & -\frac{\alpha\beta T^2}{2} \\ \frac{\beta^2T^2}{2} & \beta^2T & -\frac{\alpha\beta T^2}{2} & -\alpha\beta T \\ -\frac{\alpha\beta T^3}{3} & -\frac{\alpha\beta T^2}{2} & \frac{\alpha^2T^3}{3} & \frac{\alpha^2T^2}{2} \\ -\frac{\alpha\beta T^2}{2} & -\alpha\beta T & \frac{\alpha^2T^2}{2} & \alpha^3T \end{bmatrix} \tag{IV.2}$$

where: T is the sampling interval; σ_s is the standard deviation of the target's speed. The motion along the road (IV.1) imposes

linear constraints on the state vector of the form (III.2) with

$$\mathbf{M} = \begin{bmatrix} \alpha & 0 & \beta & 0 \\ 0 & \alpha & 0 & \beta \end{bmatrix}, \quad d = \begin{bmatrix} \gamma \\ 0 \end{bmatrix} \tag{IV.3}$$

To complete the model, it is assumed that the sensor is a radar providing measurements of the position in polar coordinates. In order to obtain a linear output equation and, hence, be able to apply linear filtering techniques, measurement conversion (from polar to cartesian coordinates) is operated. In this way, the resulting output equation turns out to be linear with constant $\mathbf{C}$ and time-varying $\mathbf{R}(t)$.

Remark (**Computational complexity**) - Hereafter a detailed operation count for the specific GSRIF road filter is given. Notice that in this case $n = 4$, $p = 2$ and $r = 2$. Hence the GSRIF road filter requires 338 flops per cycle compared to 313 flops of the conventional (non square-root) CKF [7]; this shows that the square-root implementation has almost the same computational burden of its conventional counterpart. ■

B. Curved road

It is worth pointing out that the assumption of a straight road is clearly unrealistic. However, a real curved road can be approximated, to any degree of accuracy, by means of a *polyline* consisting of multiple, say N, line segments. Hence, a road (polyline) is specified by the sequence of endpoints $(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)$ of the road (line) segments. Segment i, joining (x_{i-1}, y_{i-1}) and (x_i, y_i), can therefore be described by the line equation $\alpha_i x + \beta_i y = \gamma_i$ with coefficients

$$\alpha_i = y_i - y_{i-1}, \beta_i = x_{i-1} - x_i, \gamma_i = y_i x_{i-1} - x_i y_{i-1}.$$

The tracking of a vehicle along a known curved road requires, therefore, to properly initialize and update the road segment. For the initialization one can, for instance, proceed as follows: given the initial position $(\hat{x}, \hat{y}) = (\hat{x}(1|0), \hat{y}(1|0))$, find the closest road segment $i(1)$, i.e.

$$i(1) \;=\; \arg\min_j \; d_j$$

where

$$d_j \;=\; min\left(|\alpha_j \hat{x} + \beta_j \hat{y} - \gamma_j|, \sqrt{(\hat{x} - x_j)^2 + (\hat{y} - y_j)^2}, \sqrt{(\hat{x} - x_{j-1})^2 + (\hat{y} - y_{j-1})^2}\right)$$

As far as road segment update is concerned, let us assume that the sampling is fast enough with respect to the lengths of the road segments so that the transition can only occur between contiguous segments. Let $i(t)$ denote the road segment at time t. Given the predicted position $(\hat{x}, \hat{y}) = (\hat{x}(t+1|t), \hat{y}(t+1|t))$ and assuming that it belongs to the ith road segment, i.e. $\alpha_i \hat{x} + \beta_i \hat{y} = \gamma_i$, the road segment update can be carried out as follows

$$i(t+1) \;=\; \begin{cases} i(t), & 0 \le \lambda \le 1 \\ i(t) - 1, & \lambda < 0 \\ i(t) + 1, & \lambda > 1 \end{cases} \tag{IV.4}$$

where

$$\lambda \;=\; \frac{\hat{x} - x_{i-1}}{x_i - x_{i-1}}.$$

Notice that in this way the motion along a curved road is modeled by linear equality constraints so that the GSRIF algorithm of section 3 is still applicable by changing $\mathbf{Q}$, $\mathbf{M}$ and $\mathbf{d}$ in (IV.2) and (IV.3) whenever the road segment i changes. It is worth pointing out that this is clearly an approximation since $i(t)$ actually depends on the unknown position $(x(t), y(t))$ and an exact modeling of the motion along a polyline road would require piecewise linear constraints which are by far more complicated to deal with.

V. Tracking a target within a known road network

In a real scenario, a vehicle can move across a complicated network of roads intersecting at junctions. Hence, it is important to suitably represent the *road network* as well as to handle in a smart way the multiple hypotheses on the target's behavior in the proximity of the road junctions.

A. Road network representation

A road network can for instance be described as a graph whose nodes and arcs represent the *junctions* and, respectively, the *roads*. Let $\mathcal{J}$ and $\mathcal{R}$ denote the sets of junctions (nodes) and, respectively, roads (arcs). For each road $r \in \mathcal{R}$ the following information must be given: the junctions $j_1(r)$ and $j_2(r)$ delimiting the road; a variable $b(r)$ specifying the road's direction i.e.

$$b(r) = \begin{cases} -1, & \text{if } r \text{ is 1-way from } j_2(r) \text{ to } j_1(r) \\ 0, & \text{if } r \text{ is 2-way} \\ +1 & \text{if } r \text{ is 1-way from } j_1(r) \text{ to } j_2(r) \end{cases};$$

the number $N(r)$ of line segments making up the road; the endpoints $\{(x_i(r), y_i(r))\}_{i=0}^{N(r)}$ from which the line coefficients $\{\alpha_i(r), \beta_i(r), \gamma_i(r)\}_{i=1}^{N(r)}$ can easily be recovered. For each junction $j \in \mathcal{J}$, one needs to specify: the set $\mathcal{R}_-(j)$ of roads going into the junction; the set $\mathcal{R}_+(j)$ of roads coming out of the junction; the junction's position $(x(j), y(j))$. Clearly the junction positions must satisfy, for each road r, the consistency relations:

$$\begin{bmatrix} x(j_1(r)) \\ y(j_1(r)) \end{bmatrix} = \begin{bmatrix} x_0(r) \\ y_0(r) \end{bmatrix}, \quad \begin{bmatrix} x(j_2(r)) \\ y(j_2(r)) \end{bmatrix} = \begin{bmatrix} x_{N(r)}(r) \\ y_{N(r)}(r) \end{bmatrix}$$

B. Management of road junctions via VS-DMM

Whenever a target comes in the proximity of a junction, multiple hypotheses about the next road in the target's path must be considered; whenever the correct road has been singled out and the target is far away from the junction, the remaining (wrong) hypotheses could clearly be discarded. An effective way for handling multiple hypotheses in tracking problems consists of using *multiple models* corresponding to different hypotheses about the target's dynamics and suitably combining the estimates resulting from each model. In the specific case, a linear equality-constrained model of the type presented in section 4 is associated to each road $r \in \mathcal{R}$.

Since the target can obviously change road, a *Dynamic Multiple Model* (DMM) approach like, e.g., GPB_1 (*Generalized Pseudo Bayesian*) or IMM (*Interacting Multiple Models*) [9], is needed. Further, the fact that the number of possible roads crossing at each junction is typically much smaller than the overall number of roads, i.e. $max\left\{|\mathcal{R}_-(j)|, |\mathcal{R}_+(j)|\right\} \ll |\mathcal{R}|$ for all $j \in \mathcal{J}$, suggests a *Variable Structure-DMM* (VS-DMM) approach [1]. There can be, of course, different strategies for managing the variable model structure, i.e. for properly activating and/or deactivating roads (models) depending on the currently estimated target position and based on the known road network. Among the variety of possible options, the specific one adopted in the performance evaluation of section 6 is to activate roads departing from junctions sufficiently close to the currently predicted position and to deactivate all road models except the most likely one whenever the latter has probability sufficiently close to unity. Let $\mathcal{R}_t \subset \mathcal{R}$ denote the set of roads (models) active at time t, $r(t)$ the unknown road along which the target is moving at time t and $p_r(t) = Prob(r(t) = i|\mathbf{y}^t)$, $r \in \mathcal{R}_t$, the probability based on measurements up to time t that the target is moving along road r at time t; then

$$\mathcal{R}_{t|t} = \begin{cases} \{r\}, & \text{if } 1 - \max\limits_{r \in \mathcal{R}_t} p_r(t) < \varepsilon \\ \mathcal{R}_t, & \text{otherwise} \end{cases}$$

for a suitably chosen threshold $\varepsilon > 0$. Further,

$$\mathcal{R}_{t+1} = \mathcal{R}_{t|t} \cup \left(\bigcup_{j \in \mathcal{J}_t} \mathcal{R}_+(j) \right)$$

where

$$\mathcal{J}_t \triangleq \left\{ j \in \mathcal{J} : (\hat{\mathbf{x}} - \mathbf{x}(j))^2 + (\hat{\mathbf{y}} - \mathbf{y}(j))^2 < d_{max}^2 \right. \\ \left. \text{and } \mathcal{R}_-(j) \cap \mathcal{R}_{t|t} \neq \emptyset \right\} ,$$

$\hat{\mathbf{x}} \triangleq \hat{\mathbf{x}}(t+1|t)$, $\hat{\mathbf{y}} \triangleq \hat{\mathbf{y}}(t+1|t)$ and d_{max} is an appropriate threshold chosen as the maximum distance traveled by the target during a sampling period. The update of the road probabilities $p_r(t)$ for each $r \in \mathcal{R}_t$ is given by

$$\begin{aligned} p_r(t|t-1) &= \sum_{j \in \mathcal{R}_{t-1}} \pi_{rk} p_k(t-1) \\ p_r(t) &= \frac{\ell_r(t)\ p_r(t|t-1)}{\sum_{j \in \mathcal{R}_t} \ell_k(t))\ p_k(t|t-1)} \end{aligned}$$

where: π_{rk} are transition probabilities from road k to road r defined for each pair of roads k and r entering and, respectively, exiting some junction, i.e. $k \in \mathcal{R}_-(j)$ and $r \in \mathcal{R}_+(j)$ for some $j \in \mathcal{J}$; $\ell_r(t)$ is the likelihood of road r at time t calculated by the corresponding road filter.

Model for free-space motion

In order to account for the possibility that the target follows an *offroad* path and/or the available road-map is incomplete, i.e. some roads are missing, a model for *free space motion* could be included in the multiple model set. This model can be defined, for instance, as an unconstrained constant velocity model (III.1) with the transition matrix $\mathbf{A}$ as in (IV.2) and the process covariance matrix given by

$$\mathbf{Q} = diag\left\{ \begin{bmatrix} \frac{\sigma_x^2 T^3}{3} & \frac{\sigma_x^2 T^2}{2} \\ \frac{\sigma_x^2 T^2}{2} & \frac{\sigma_x^2 T^3}{3} \end{bmatrix}, \begin{bmatrix} \frac{\sigma_y^2 T^3}{3} & \frac{\sigma_y^2 T^2}{2} \\ \frac{\sigma_y^2 T^2}{2} & \frac{\sigma_y^2 T^3}{3} \end{bmatrix} \right\}$$

where σ_x and σ_y are the standard deviations of the horizontal and, respectively, vertical velocity components. Hereafter, the free space motion model will be referred to as the *virtual road* $r = 0$ while the *real roads* are associated to positive integers $r > 0$. Clearly it is convenient to keep the free-space motion model active at each time t, i.e. $0 \in \mathcal{R}_t$ for all t.

C. Algorithm for GIA fusion

Multiple Model algorithms involve a combination (*fusion*) among the estimates $\hat{\mathbf{x}}_i$ and covariances $\mathbf{P}_i$ provided, for the various models, by the associated filters. This fusion takes the form:

$$\hat{\mathbf{x}} = \sum_{i=1}^{\mu} p_i\ \hat{\mathbf{x}}_i,\ \mathbf{P} = \sum_{i=1}^{\mu} p_i \left[\mathbf{P}_i + (\hat{\mathbf{x}} - \hat{\mathbf{x}}_i)(\hat{\mathbf{x}} - \hat{\mathbf{x}}_i)' \right] \quad \text{(V.1)}$$

where μ is the number of models and p_i are suitable probabilities, i.e. $\sum_{i=1}^{\mu} p_i = 1$. In order to efficiently exploit the GSRIF (propagating in time the GIA triplet instead of the estimate-covariance pair) for implementing the modal filters in any *multiple model* algorithm, it would therefore be highly desirable to directly fuse canonical GIAs $(\mathbf{S}_i, \mathbf{U}_i, \mathbf{b}_i)$ corresponding to $(\hat{\mathbf{x}}_i, \mathbf{P}_i)$ into a canonical GIA $(\mathbf{S}, \mathbf{U}, \mathbf{b})$ corresponding to $(\hat{\mathbf{x}}, \mathbf{P})$. This direct fusion of GIAs would avoid the computationally expensive conversions $(\mathbf{S}_i, \mathbf{U}_i, \mathbf{b}_i) \rightarrow (\hat{\mathbf{x}}_i, \mathbf{P}_i)$ and $(\hat{\mathbf{x}}, \mathbf{P}) \rightarrow (\mathbf{S}, \mathbf{U}, \mathbf{b})$ which, in turn, would certainly neutralize the numerical benefits of the square-root approach. Due to lack of space, the derivation is omitted and a square-root algorithm for GIA fusion is given below.

Algorithm - Given the canonical GIAs $(\mathbf{S}_i, \mathbf{U}_i, \mathbf{b}_i)$ for $i = 1, 2, \ldots, \mu$ and the corresponding probabilities $p_i \in [0, 1]$, a canonical fused GIA $(\mathbf{S}, \mathbf{U}, \mathbf{b})$ can be obtained by proceeding as follows:

1) Compute $\hat{\mathbf{x}}_i = \mathbf{S}_i^{-1}\mathbf{b}_i$ and $\mathbf{M}_i = \mathbf{S}_i^{-1}\mathbf{U}_i$ for $i = 1, 2, \ldots, \mu$.
2) Compute $\hat{\mathbf{x}} = \sum_{i=1}^{\mu}\ p_i\hat{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_i = \hat{\mathbf{x}} - \hat{\mathbf{x}}_i$ for $i = 1, 2, \ldots, \mu$.
3) Form the matrix
$$\mathbf{\Phi} = \left[\sqrt{p_1}\mathbf{M}_1, \sqrt{p_1}\tilde{\mathbf{x}}_1, \ldots, \sqrt{p_\mu}\mathbf{M}_\mu, \sqrt{p_\mu}\tilde{\mathbf{x}}_\mu \right].$$
4) Perform, via Givens rotations, the transformation $\mathbf{\Phi T} = [\mathbf{\Phi}_1,\ \mathbf{0}]$ where $\mathbf{T}$ is orthogonal and $\mathbf{\Phi}_1$ is square and upper triangular.
5) Transform the non canonical GIA $(\mathbf{I}, \mathbf{\Phi}_1, \hat{\mathbf{x}})$ into a canonical one $(\mathbf{S}, \mathbf{U}, \mathbf{b})$ by means of the procedure presented in section 2. ■

VI. Performance evaluation

Consider the simple road scenario and the two vehicle paths of fig. 1. Notice that path 2 is partly offroad. The

following tracking algorithms will be compared: **(1)** unconstrained Kalman filter denoted as **KF**; **(2)** GPB_1 with road filters implemented as projection Kalman filters, wherein the unconstrained Kalman filter estimate is projected along the road, denoted as **PKF**; **(4)** GPB_1 with road filters implemented by the square-root constrained Kalman filter proposed in this work, denoted as **GSRIF**. The performance is evaluated in terms of the distance between the true and estimated vehicle positions averaged over 200 independent Monte Carlo runs. The following parameters have been chosen in the simulations: sampling interval $T = 5$, std. dev. for the vehicle speed $\sigma_s = 2m/s$, std. deviations for the radar measurements $\sigma_\rho = 5m$ for the range and $\sigma_\theta = 1^\circ$ for the azimuth. The simulation results are reported in figs. 2-3 (time plots) and table I (performance index averaged over time). From the examination of these results, it can be seen that GSRIF outperforms PKF which, in turn, outperforms KF and that this holds even in the partly offroad path thanks to the inclusion of the free-space motion filter in the filter bank.

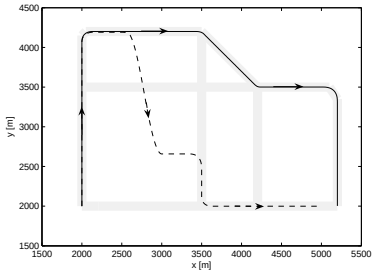

Fig. 1. Road scenario and paths used in simulation.

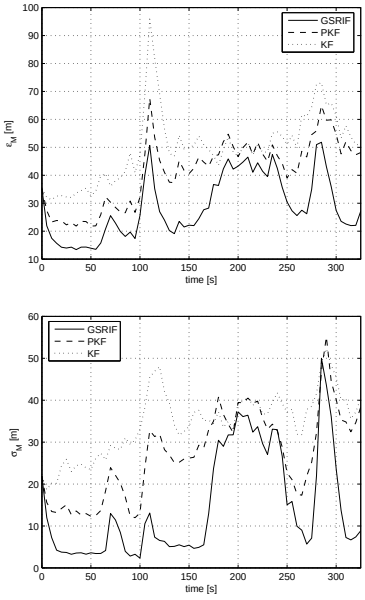

Fig. 2. Mean square error and standard deviation on path 1.

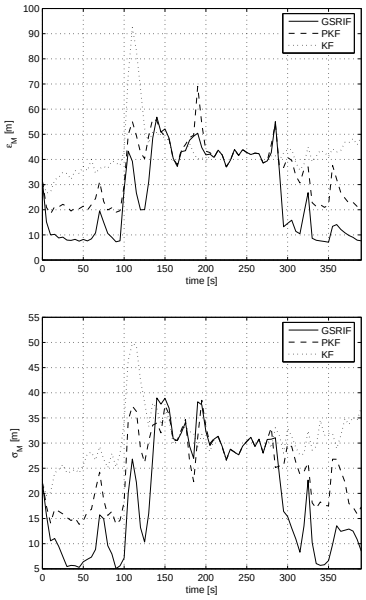

Fig. 3. Mean square error and standard deviation on path 2 (offroad).

path	KF		PKF		GSRIF	
	$\bar{\epsilon}$	$\bar{\sigma}$	$\bar{\epsilon}$	$\bar{\sigma}$	$\bar{\epsilon}$	$\bar{\sigma}$
1	48.90	34.67	35.88	26.96	17.20	14.81
2	42.79	30.69	35.19	24.88	26.22	19.35

TABLE I

VII. Conclusions

The paper has addressed tracking of a ground target moving across a known road network. Exploiting knowledge of the road-map, a *Variable Structure Multiple Model* algorithm with modal filters consisting of constrained square-root filters for *onroad* motion plus an unconstrained square-root filter for *offroad* motion, has been adopted and its computational/numerical efficiency as well as its tracking performance have been assessed.

References

[1] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati and I. Kadar: "Ground target tracking with variable structure IMM estimator", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 36, pp 26-46, 2000.

[2] M. Ulmke and W. Koch: "Road-map assisted ground moving target tracking", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, pp. 1264-1274, 2006.

[3] D. Simon and T.L. Chia: "Kalman filtering with state-equality constraints", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 38, pp. 128-136, 2002.

[4] A. Farina, L. Ferranti and G. Golino: "Constrained tracking filter for A-SMGCS", *Proc. 6th International Conference on Information Fusion*, pp. 4121-4128, Cairns, Australia, 2003.

[5] L. Chisci and G. Zappa: "Square-root Kalman filtering of descriptor systems", *Systems & Control Letters*, vol. 19, pp. 325-334, 1992.

[6] C.C. Paige and M.A. Saunders: "Least-squares estimation of discrete linear dynamic systems using orthogonal transformations", *SIAM J. Numerical Analysis*, vol. 14, pp. 180-193, 1977.

[7] S. Ko and R.R. Bitmead: "State estimation for linear systems with equality constraints", *Automatica*, vol. 43, pp. 1363-1368, 2007.

[8] M. Verhaegen and P. Van Dooren: “Numerical aspects of different Kalman filter implementations”, *IEEE Trans. on Automatic Control*, vol. 31, pp. 907-917, 1986.
[9] Y. Bar-Shalom and X.R. Li: “*Multitarget-multisensor tracking: principles and techniques”*, 1995.