U.R.S.I. FRANCE JS 18 — GEOLOCATION AND NAVIGATION IN SPACE AND TIME

# Invariant Extended Kalman Filter for Target Tracking
# *Filtre de Kalman Etendu Invariant pour Pistage de Cibles*

***Marion Pilté***[1]***, Silvère Bonnabel***[2]***, and Frédéric Barbaresco***[3]

[1] *Mines ParisTech & Thales Air Systems, marion.pilte@mines-paristech.fr*
[2] *Mines ParisTech, silvere.bonnabel@mines-paristech.fr*
[3] *Thales Air Systems, frederic.barbaresco@thalesgroup.com*

**Keywords:** Kalman, tracking, Lie groups
**Mots-clefs:** Kalman, pistage, groupes de Lie

**Abstract:**
A 3D target model expressed in intrinsic coordinates will be developed in this article. The frame used is the Frenet-Serret frame, that is a practical frame to represent the commands a pilot can have on his aircraft for instance. A quite accurate description of the possible motions of an aircraft is to assume the commands are piecewise constant. Once the target model is derived, a filtering algorithm is needed to perform state estimation. As the target model is not expressed in a vectorial space, but rather in a Lie group setting, a novel algorithm, based on results from the inertial navigation field has to be established. This new filter is called the Invariant Extended Kalman Filter (IEKF).

**Résumé:**
Cet article présente un modèle de cible en 3D et en coordonnées intrinsèques. Le repère de Frenet-Serret est utilisé, il permet de représenter les commandes qu'un pilote peut avoir sur son appareil par exemple. Une description relativement réaliste des mouvements possibles d'un avion est de considérer des commandes constantes par morceaux. Une fois le modèle de cible établi, il faut un algorithme de filtrage pour faire l'estimation de l'état. Comme le modèle de cible n'est pas exprimé dans un espace vectoriel, comme c'est le cas habituellement, mais dans un groupe de Lie, il faut développer un algorithme de filtrage nouveau pour le pistage radar, qui s'inspire de ce qui existe déjà dans le domaine de la navigation inertielle, à savoir le filtre de Kalman étendu invariant.

## 1 Introduction

One application of radar target tracking is to maintain the tracks initiated within the beam fo the radar. In order to be sure the target is not lost, an estimation of the state of the target is necessary, and more specifically, a precise position and velocity estimation is needed. For other radar applications, such as target guiding, a very precise estimation of the velocity vector is essential (among which its direction is of crucial importance). Filtering algorithms are thus very popular among the radar community to perform state estimation.
To perform target tracking, two elements are essential and complementary. The performances of the estimation is due to the target model on the one hand, and to the filtering algorithm on the other hand. First the target model must be accurate enough to describe the possible motions of different classes of targets. Indeed a radar is required to track any aircraft, missiles, boats and all possible types of targets. The accuracy required for each target is set by the client of course, and a single radar cannot track with a high accuracy all types of targets, but it has to keep track of all of them. The target model must thus be loose enough to take into account all these different classes of targets. The other element needed is a filtering algorithm. It is fed with the target model and proceeds in two steps. The first step is merely the propagation of the target model. The second step occurs after the algorithm receives a measurement from the radar, and it can thus adjust the prediction made with the measurement received. The filtering algorithm outputs the state estimation along with its covariance, giving the confidence one can have in the estimation.
In industrial applications, most target models are linear ones, among which we find the famous Singer model [1] or constant velocity models. For the filtering algorithms, the linear Kalman filter [2], or the Interacting Multiple Model (IMM) [3] that runs several filters in parallel are very popular. For nonlinear target or measurement models, the Extended Kalman Filter (EKF) [4] is the most well-known, although it is not very robust. A robustified solution is to use the Castella noise-adaptive algorithm of [5]. Other filtering techniques include particle filters [6], or the Rao-Blackwell particle filter [7], which is a refined particle filter that requires less particles.
However, targets are more maneuvering nowadays, and the linear target models are not efficient enough with the new generation of targets. In this paper we thus propose a target model expressed in intrinsic coordinates, based on the 3D Frenet-Serret frame. Our target model is close to the ones of [8], [9] and [10]. A new Kalman-based filtering algorithm is derived to match specifically this model, as it is not expressed in a vectorial space as usual. This filter is called the Invariant Extended Kalman Filter (IEKF), see for example [11] or [12]. Some specific properties of the IEKF will be highlighted, among which the fact that it is more robust than an EKF.

The paper is organized as follows. In section 2 we derive the target model from the Frenet-Serret frame evolution. In section 3, we detail the construction of an appropriate filter, the IEKF of [13] adapted to the target tracking problem. Finally, in section 4 we show the performances of the designed estimation method.

## 2 Target model in the Frenet-Serret frame

The idea is to express the target model in intrinsic coordinates, and to model the evolution by assuming the target undergoes constant command motions (as the ones a pilot would apply on an aircraft for instance). The commands are represented by piecewise constant parameters, as it is described below.

### 2.1 Derivation of the model equations

The target model is based on the Frenet-Serret frame. This is inspired by the model used in [14]. The three vectors of the frame are called the tangential vector, $T_t$, the normal vector, $N_t$ and the binormal vector, $B_t$. Their evolution are known with respect to the tangential velocity $u_t$, the curvature $\kappa_t$ and the torsion $\tilde{\tau}_t$ of the curve, as described in (1). Then we let $\gamma_t = u_t \kappa_t$ and $\tau_t = u_t \tilde{\tau}_t$ be the curvature and the torsion, with a slight abuse of language. In the simulations of Section 4, we will plot $\kappa_t$ and $\tilde{\tau}_t$, the usual definitions of the curvature and the torsion.

$$\frac{dT}{dt} = u\kappa N, \; \frac{dN}{dt} = u(-\kappa T + \tilde{\tau} B), \; \frac{dB}{dt} = -u\tilde{\tau} N \tag{1}$$

The state of the target is then composed of the rotation matrix $R_t = \begin{pmatrix} T_t & R_t & B_t \end{pmatrix}$, the cartesian position of the target $x_t$, the curvature of the trajectory $\gamma_t$, its torsion $\tau_t$ and the tangential velocity of the target $u_t$. The state is explicitly defined by (2). It does not belong to a vectorial space due to the presence of the rotation matrix $R_t$.

$$X_t = \begin{pmatrix} R_t & x_t & \gamma_t & \tau_t & u_t \end{pmatrix} \tag{2}$$

The evolution of this state can be derived from the Frenet-Serret equations, and from the choice of piecewise constant commands. These commands are represented by the parameters $\gamma_t, \tau_t, u_t$. In equation (3), we have let $(a)_\times \in \mathbb{R}^{3\times3}$ denote the skew-symmetric matrix associated with the cross product with vector $a \in \mathbb{R}^3$. Let us also call $v_t = \begin{pmatrix} u_t & 0 & 0 \end{pmatrix}^T$, and $\omega_t = \begin{pmatrix} \tau_t & 0 & \gamma_t \end{pmatrix}^T$. Finally the noises are supposed to be white and gaussian, and are denoted $w_t^\omega, w_t^{;} w_t^\gamma, w_t^\tau, w_t^u$; and they account for small changes over time, but also for the jumps in the piecewise constant commands, as it will be explained in Section 4.

$$\frac{dx_t}{dt} = R_t v_t + w_t^x, \; \frac{dR_t}{dt} = R_t(\omega_t + w_t^\omega)_\times, \; \frac{d\gamma_t}{dt} = 0 + w_t^\gamma, \; \frac{d\tau_t}{dt} = 0 + w_t^\tau, \; \frac{du_t}{dt} = 0 + w_t^u \tag{3}$$

### 2.2 Particular form of the state

The state is in fact composed of two fairly different parts. We can separate a matricial part and a vectorial part in it. Indeed, the translation and rotation can be considered as an element of $SE(3)$, and we call

$$\chi_t = \begin{pmatrix} R_t & x_t \\ 0_{1,3} & 1 \end{pmatrix}$$

The group $SE(3)$ of rotations and translations in 3D is of dimension 6. The other part of the sate is composed of the curvature, the torsion and the tangential velocity:

$$\zeta = \begin{pmatrix} \gamma_t \\ \tau_t \\ u_t \end{pmatrix}$$

Before going further, let us define some basic operations on the Lie group $SE(3)$. A matrix Lie Group is a set of invertible matrices, stable by multiplication and inversion. A Lie group is differentiable. One can thus define a tangent space at the neutral element to the group, called a Lie algebra. A Lie algebra is a vectorial space, equipped with an intern bilinear multiplication. $SE(3)$ describes the possible motions of a point mass in the 3D space. It is defined as follows:

$$SE(3) = \left\{ \begin{pmatrix} R & x \\ 0_{1,3} & 1 \end{pmatrix}, x \in \mathbb{R}^3, R \in \mathcal{M}_3, RR^T = R^T R = I_3, \det(R) = 1 \right\}$$

We can represent the associated Lie algebra as:

$$\mathfrak{se}(3) = \left\{ \begin{pmatrix} 0 & -c & b & \alpha \\ c & 0 & -a & \beta \\ -b & a & 0 & \gamma \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} a \\ b \\ c \end{pmatrix} \in \mathbb{R}^3, \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \in \mathbb{R}^3 \right\}$$

We call $(.)_\times$ the operator

$$(.)_\times : \mathbb{R}^3 \to \mathcal{M}_3, \quad \begin{pmatrix} a \\ b \\ c \end{pmatrix} \rightarrowtail \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}$$

It is then possible to write the evolution equations (3) in a slightly more compact way, as in (4) with $\omega_t = \begin{pmatrix} \tau_t & 0 & \gamma_t \end{pmatrix}^T$ and $v_t = \begin{pmatrix} u_t & 0 & 0 \end{pmatrix}^T$. To do this, one must introduce the corresponding noises $w_t^\chi = \begin{pmatrix} (w_t^R)_\times & w_t^x \\ 0_{1\times3} & 0 \end{pmatrix}$, and $w_t^\zeta = (w_t^\gamma, w_t^\tau, w_t^u)$ and the matrix $\nu_t$ as follows:

$$\nu_t = \begin{pmatrix} 0 & -\gamma_t & 0 & u_t \\ \gamma_t & 0 & -\tau_t & 0 \\ 0 & \tau_t & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\frac{d\chi_t}{dt} = \chi_t(\nu_t + w_t^\chi) = \begin{pmatrix} R(\omega_t)_\times & Rv \\ 0_{1\times3} & 0 \end{pmatrix}, \quad \frac{d\zeta_t}{dt} = 0 + w_t^\zeta \tag{4}$$

The measurements are assumed to be in cartesian coordinates, and occur at times $t_n, n \in \mathbb{N}$, and are thus expressed as (5), with $V_n \in \mathbb{R}^3$ a gaussian independant white noise.

$$Y_n = x_{t_n} + V_n \tag{5}$$

The classical Kalman filters cannot be applied to this formulation of the target state. However, if we assume that $\zeta_t$, the vectorial part of the state is known, then $\chi_t$ follows a left-invariant differential equation. We thus introduce the Invariant Extended Kalman Filter, as in [11], and extend it to the case where the velocity, the curvature and the torsion are not known, as in [15] or [12].

## 3 Estimation algorithm

In this section, we derive the filter's equations to perform the estimation.

### 3.1 Similarities with the Invariant theory

$\chi_t$ verifies an equation of the type (6).

$$\frac{d}{dt}\chi_t = f_{i_t}(\chi_t) + \chi_t w_t \tag{6}$$

where $i_t = \zeta_t \in \mathbb{R}^3$ is a known input (we suppose $\zeta_t$ known for the moment)., $w_t$ is a continuous white gaussian noise, and $f$ satisfies the condition:

$$f_i(ab) = af_i(b) + f_i(a)b - af_i(Id)b \tag{7}$$

forall $(i, a, b) \in \mathbb{R}^3 \times SE(3) \times SE(3)$. An Invariant EKF can thus be designed to estimate $\chi$.
In the case where the input $\zeta_t$ is not known, that is of interest in the target tracking problem, the algorithm can be adapted to treat $\chi$ as a Lie group part, and $\zeta$ as a standard vectorial part.
The system satisfies thus (8).

$$\left\{ \begin{array}{lcl} \frac{d}{dt}\chi_t & = & f_i(\chi_t, \zeta_t) + \chi_t w_t^\chi \\ \frac{d}{dt}\zeta_t & = & g(\zeta_t) + w_t^\zeta \end{array} \right. \tag{8}$$

where in our case

$$f_\zeta : \begin{pmatrix} R & x \\ 0_{1,3} & 1 \end{pmatrix} \to \begin{pmatrix} R(\omega)_\times & Rv \\ 0_{1,3} & 0 \end{pmatrix}, \quad g(\zeta) = 0_{3,1}$$

We can also write the observations (5) with the help of the Lie group setting:

$$Y_n = \chi_{t_n} \begin{pmatrix} 0_{3,1} \\ 1 \end{pmatrix} + \begin{pmatrix} V_n \\ 0 \end{pmatrix}$$

The condition (7) for $f$ is easily verified.

## 3.2 Derivation of the algorithm

### 3.2.1 Error definition

The classical definition of the error for a vectorial state $X_t$, $\eta = \hat{X}_t - X_t$ does not hold here. Indeed, if $\chi_1$ and $\chi_2$ belong to the Lie group $SE(3)$, there is no reason why $\chi_1 - \chi_2$ should also belong to this same Lie group. We define the error differently depending on which part of the state we are considering:

$$\eta_t^\chi = \chi_t^{-1}\hat{\chi}_t$$

$$\eta_t^\zeta = \hat{\zeta}_t - \zeta_t$$

More explicitly the global error $\eta_t = (\eta_t^\chi, \eta_t^\zeta)$ is defined as:

$$\eta = \begin{pmatrix} \eta_t^R \\ \eta_t^x \\ \eta_t^\gamma \\ \eta_t^\tau \\ \eta_t^u \end{pmatrix} = \begin{pmatrix} R_t^T \hat{R}_t \\ R_t^T(\hat{x}_t - x_t) \\ \hat{\gamma}_t - \gamma_t \\ \hat{\tau}_t - \tau_t \\ \hat{u}_t - u_t \end{pmatrix} \tag{9}$$

### 3.2.2 Linearization of the error and propagation step

The propagation equations are:

$$\frac{d}{dt}\hat{\chi}_t = f_{\omega_t, u_t}(\hat{\chi}_t) = \hat{\chi}_t \hat{\nu}_t, \quad \frac{d}{dt}\hat{\zeta}_t = g(\hat{\zeta}_t) = 0 \tag{10}$$

Now let us assume once again (for the last time) that $\zeta_t$ is known. We can compute the error $\eta_t^\chi$ evolution. This gives:

$$\frac{d}{dt}\eta_t^\chi = \nu_t \eta_t - \eta_t \nu_t - \eta_t w_t^\chi$$

This equation has the particular property that it does not depend on the predicted state $\hat{\chi}_t$ at all. This is due to the property (7) of the evolution.
As for our radar tracking application $\zeta_t$ is not known the evolution of $\eta^\chi$ is slightly modified and it writes

$$\frac{d}{dt}\eta_t^\chi = \hat{\nu}_t \eta_t - \eta_t \nu_t - \eta_t w_t^\chi \tag{11}$$

indeed, the matrix $\nu_t$ depends on the vector $\zeta_t$, so it has to be estimated as well. The evolution of $\eta_t^\zeta$ is more conventional:

$$\frac{d}{dt}\eta_t^\zeta = 0 + w_t^\zeta \tag{12}$$

To linearize equations (11) and (12), see [11], we let $\eta_t^R \approx I_3 + (\xi_t^R)_\times$. This means that $\xi_t^R \in \mathbb{R}^3$ is a small instantaneous rotation vector. We also let $\xi_t^x = \eta_t^x$, $\xi_t^\gamma = \eta_t^\gamma$, $\xi_t^\tau = \eta_t^\tau$ and $\xi_t^u = \eta_t^u$. Then we mimic the EKF methodology, and perform a first order linearization in the components of $\xi$, and we also neglect terms of order $\|\xi\|\|w\|$. To do this, we use a property of Lie groups: $(\xi_t^R)_\times(\hat{\omega}_t)_\times - (\hat{\omega}_t)_\times(\xi_t^R)_\times = (\xi_t^R \times \hat{\omega}_t)_\times$. This allows to identify the term $\frac{d}{dt}\xi_t^R$ using that $(a)_\times = (b)_\times \implies a = b$. We use the same denomination for $\xi$ as for $\eta$:

$$\xi_t = \begin{pmatrix} \xi_t^R \\ \xi_t^x \\ \xi_t^\gamma \\ \xi_t^\tau \\ \xi_t^u \end{pmatrix} \in \mathbb{R}^9$$

During the propagation step, the error evolves as:

$$\frac{d}{dt}\xi_t = A_t \xi_t + w_t$$

with

$$A_t = \begin{pmatrix}
0 & -\hat{\gamma}_t & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
\hat{\gamma}_t & 0 & -\hat{\tau}_t & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \hat{\tau}_t & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & -\hat{\gamma}_t & 0 & 0 & 0 & -1 \\
0 & 0 & -\hat{u}_t & \hat{\gamma}_t & 0 & -\hat{\tau}_t & 0 & 0 & 0 \\
0 & \hat{u}_t & 0 & 0 & \hat{\tau}_t & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

This permits tp write the covariance evolution during the propagation: the covariance matrix evolves with the Riccati equation (13).

$$\frac{d}{dt}P_t = A_t P_t + P_t A_t^T + Q_t \tag{13}$$

### 3.2.3 Gain computation and update step

The update of the state writes:

$$\hat{\chi}_{t_n}^{+} = \hat{\chi}_{t_n} \exp(L_n^{\chi}(\hat{\chi}_{t_n}^{-1} Y_n)), \;\; \hat{\zeta}_{t_n}^{+} = \zeta_{t_n} + L_n^{\zeta}(\hat{\chi}_{t_n}^{-1} Y_n) \tag{14}$$

more explicitly, this can be developed as:

$$\begin{pmatrix} \hat{R}_{t_n}^{+} \\ \hat{x}_{t_n}^{+} \\ \hat{\gamma}_{t_n}^{+} \\ \hat{\tau}_{t_n}^{+} \\ \hat{u}_{t_n}^{+} \end{pmatrix} = \begin{pmatrix} \hat{R}_{t_n} \exp_m[(\delta_\omega)_\times] \\ \hat{R}_{t_n} \hat{x}_{t_n} + B(\delta_\omega)\delta_x \\ \hat{\gamma}_{t_n} + \delta_\gamma \\ \hat{\tau}_{t_n} + \delta_\tau \\ \hat{u}_{t_n} + \delta_u \end{pmatrix} \tag{15}$$

where $\begin{pmatrix} \delta_\omega & \delta_x & \delta_\gamma & \delta_\tau & \delta_u \end{pmatrix}^T = L_n(\hat{R}_{t_n}(Y_n - \hat{x}_{t_n}))$, $\exp_m$ denotes the matrix exponential map in $\mathcal{M}_3$, and

$$B(\delta_\omega) = I_3 + \frac{1 - \cos\|\delta_\omega\|}{\|\delta_\omega\|^2}(\delta_\omega)_\times + \frac{\delta_\omega - \sin\|\delta_\omega\|}{\|\delta_\omega\|^3}[(\delta_\omega)_\times]^2$$

The gain matrix $L_n \in \mathbb{R}^{9\times 3}$ is computed with the Riccati equation (13), as will be explained in the following. The innovation is defined as $\hat{\chi}_{t_n}^{-1} Y_n = \hat{R}_{t_n}^T(Y_n - \hat{x}_{t_n})$, it verifies:

$$\hat{R}_{t_n}^T(Y_n - \hat{x}_{t_n}) = \hat{R}_{t_n}^T(x_{t_n} - \hat{x}_{t_n}) + R_{t_n}^T V_n = -(\eta_{t_n}^R)^{-1}\eta_{t_n}^x + \hat{R}_{t_n}^T V_n$$

thus, as $\eta_t^x = \xi_t^x$ and $(\xi_t^R)_\times \xi_t^x$ is of order two, then the linearization gives $\hat{R}_t^T(Y_n - \hat{X}_{t_n}) \approx -H\xi + \hat{R}_{t_n}^T V_n$, where $H \in \mathbb{R}^{3\times 9}$ is defined as $H = \begin{pmatrix} 0_{3\times 3} & I_3 & 0_{3\times 3} \end{pmatrix}$. We are now able to derive the update $\xi_t^+$ of $\xi_t$ from (15) and (9), which gives:

$$\xi_{t_n}^{+} = \xi_{t_n} - L_n\left[\begin{pmatrix} 0_{3\times 3} & I_3 & 0_{3\times 3} \end{pmatrix} \xi_{t_n} - \hat{R}_{t_n}^T V_n\right] \tag{16}$$

Finally, the Kalman gain can be computed with the full Riccati equations, with $Q_t$ and $N_n$ the process and measurement noise covariance respectively:

$$\begin{aligned} \frac{d}{dt}P_t &= A_t P_t + P_t A_t^T + Q_t \\ S_n &= H P_{t_n} H^T + \hat{R}_{t_n}^T N_n \hat{R}_{t_n} \\ L_n &= P_{t_n} H^T S^{-1} \\ P_{t_n}^{+} &= (I_9 - L_n H) P_{t_n} \end{aligned}$$

### 3.2.4 Summary of the filter's equations

To sum up the results obtained before, we can write extensively the filter's equations.

1. Propagation step:
   - Solve $\frac{d}{dt}\hat{\chi}_t = \hat{\chi}_t \hat{\nu}_t$ and $\frac{d}{dt}\hat{\zeta}_t = 0$
   - Solve the Riccati equation $\frac{d}{dt}P_t = A_t P_t + P_t A_t^T + Q_t$
2. Update step:
   - Compute the innovation $z_n = \hat{R}_{t_n}(Y_n - \hat{x}_{t_n})$
   - Compute the Kalman gain $L_n = P_{t_n} H (H P_{t_n} H^T + \hat{R}_{t_n} N \hat{R}_{t_n}^T)$ with $H = \begin{pmatrix} 0_{3,3} & I_3 & 0_{3,3} \end{pmatrix}$
   - Update the state $\hat{\chi}_{t_n}^{+} = \hat{\chi}_{t_n} \exp((L_n z_n)_{1:6}$ and $\hat{\zeta}_{t_n}^{+} = \zeta_{t_n} + (L_n z_n)_{7:9}$
   - Update the covariance $P_{t_n}^{+} = (I_9 - L_n H) P_{t_n}$

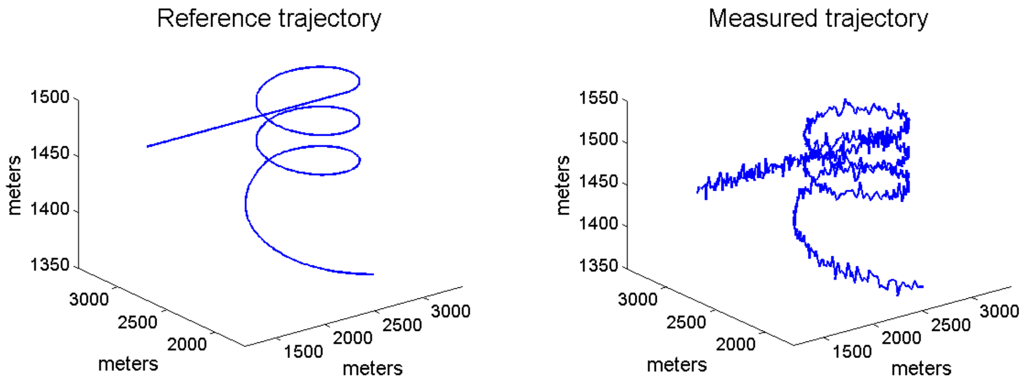

*Figure 1 – Reference trajectory and trajectory with simulated noise*

## 4 Results

The target model along with the filtering algorithm can estimate some trajectories that do not stay in a plane, and which require the use of the Interacting Multiple Model (IMM) if we use standard models such as constant velocity, constant horizontal or vertical turns or based on the linear Singer model [1]. The type of trajectory we want to track is shown on figure 1, and the estimation by the IEKF on figure 2.

The trajectory shown in figure 1 has been elaborated in three parts. The first part is a straight line motion with constant velocity, the second part is a helical motion, with different, but constant tangential velocity, torsion and curvature, and the last part is a circle in a plane, with again different velocity and curvature (the torsion is zero since we lie in a plane). Measurement noise is added by hand, independently on the three cartesian position coordinates, the amplitude of the measurement noise is visible on the figure on the right. It is quite high, so that it is more challenging for the filtering algorithm.

The estimations made by an IEKF of the position $x$, the curvature $\kappa$, the torsion $\tilde{\tau}$ and the norm of the velocity $u$ are displayed in figure 2. The parameters are poorly initialized on purpose to see the behaviour of the filter when confronted to high initial errors. In practice the position is relatively well known, as well as the norm of the velocity, but the curvature or the torsion are not.

The results show that the position is very accurately estimated. The process noise tuning was done such that the estimations were precise on constant motions during the trajectory (especially for the norm of the velocity and the curvature). So for this tuning, the estimations have a small delay after the jumps, which can be reduced by increasing the process noise on these parameters but at the cost of a lesser precision on the constant parts. The process noise tuning has the same issues as for a classical Kalman filter, where one must have a balance between the precision during constant motions and the capacity to react rapidly to jumps.

The estimation of the torsion is more difficult. Indeed, the torsion comes from a third derivative of the position, so it is barely observable. In theory, it is observable, but in practice since we measure noisy position, it is hard to recover. However, we see that its estimation eventually converge, and that it becomes better after the first jump, when the trajectory is not a straight line anymore.

The lack of precision of the torsion estimation does not really affect the quality of the estimation of the other parameters, and it is not in itself a major problem. Indeed, what is most relevant to assess the performances of a filtering algorithm is either the precision of the position, and the ability of the algorithm to filter the noise, or the precision of the velocity vector, to be sure where the target is actually heading, and at which speed. It is very difficult to tune a filter to achieve both very efficiently, but we can obtain a very good balance with the IEKF. However, the torsion (and the curvature) are not as essential to the user. But we cannot suppress them of the state as they bring some necessary degrees of freedom in the trajectory (the torsion is what allows to get out of a plane).

## 5 Conclusions

In this paper we have presented a target model in 3D, and based on the Frenet-Serret frame. This particular frame allows us to express the model in intrinsic coordinates, which can best represent the commands of a pilot in his aircraft for instance. To perform estimation with this target model, we have seen that an Invariant Extended Kalman Filter is most appropriate, because it is suited to the evolution of a part of the state, which is not vectorial as it is usually the case.

Another advantage of the IEKF is to have more stability properties than an EKF. Indeed, we have seen that the evolution of the error does not depend on the predicted position or rotation matrix of the target, contrary to an EKF.

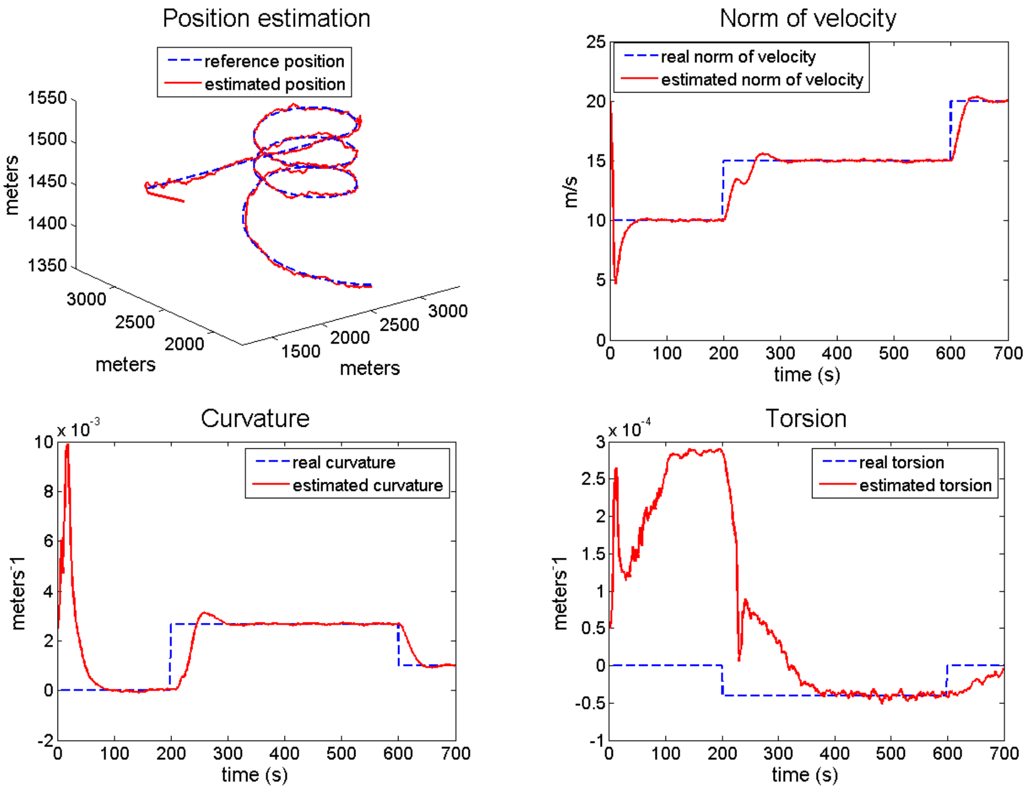

*Figure 2 – Estimation of the position (top left) and of parameters $u_t$ (top right), $\gamma_t$ (bottom left), $\tau_t$ (bottom right)*

It is very hard to use this model with other filtering algorithms, since they are all designed to match a vectorial state model. The orthogonality constraints implied by the use of the Frenet-Serret frame are highly nonlinear, and an EKF has difficulties with this type of formulations.

## Acknowledgments

This work is supported by a DGA-MRIS scholarship and by Thales Air Systems.

## 6 References

[1] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 473–483, 1970.

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[3] Y. Bar-Shalom, K. Chang, and H. A. Blom, "Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 296–300, 1989.

[4] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2004.

[5] F. R. Castella, "An adaptive two-dimensional Kalman tracking filter," *IEEE Transactions on Aerospace and Electronic Systems*, no. 6, pp. 822–829, 1980.

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

[7] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 176–183, Morgan Kaufmann Publishers Inc., 2000.

[8] T. Bullock and S. Sangsuk-Iam, "Maneuver detection and tracking with a nonlinear target model," in *Decision and Control, 1984. The 23rd IEEE Conference on*, vol. 23, pp. 1122–1126, IEEE, 1984.

[9] R. Bishop and A. Antoulas, "Nonlinear approach to aircraft tracking problem," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 5, pp. 1124–1130, 1994.

[10] P. Bunch and S. Godsill, "Dynamical models for tracking with the variable rate particle filter," in *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 1769–1775, IEEE, 2012.

[11] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, 2017.

[12] M. Pilté, S. Bonnabel, and F. Barbaresco, "Tracking the Frenet-Serret frame associated to a highly maneuvering target in 3D." working paper or preprint, July 2017.

[13] A. Barrau and S. Bonnabel, "Intrinsic filtering on lie groups with applications to attitude estimation," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 436–449, 2015.

[14] S. J. Godsill, J. Vermaak, W. Ng, and J. F. Li, "Models and algorithms for tracking of maneuvering objects using variable rate particle filters," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 925–952, 2007.

[15] M. Pilté, S. Bonnabel, and F. Barbaresco, "An innovative nonlinear filter for radar kinematic estimation of maneuvering targets in 2d," in *Radar Symposium (IRS), 2017 18th International*, pp. 1–10, IEEE, 2017.