HAL
open science

Drone Tracking Using an Innovative UKF

Marion Pilté, Silvere Bonnabel, Frédéric Barbaresco

To cite this version:

Marion Pilté, Silvere Bonnabel, Frédéric Barbaresco. Drone Tracking Using an Innovative UKF. 3rd conference on Geometric Science of Information (GSI 2017), Nov 2017, Paris, France. hal-01692469

HAL Id: hal-01692469

https://minesparis-psl.hal.science/hal-01692469v1

Submitted on 25 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Drone Tracking Using an Innovative UKF

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

[1] Mines ParisTech, PSL Research University, Center for Robotics, Paris, France,
marion.pilte@mines-paristech.fr,
silvere.bonnabel@mines-paristech.fr,
[2] Thales Air Systems, Hameau de Roussigny, France,
frederic.barbaresco@thalesgroup.com

Abstract. This paper addresses the drone tracking problem, using a model based on the Frenet-Serret frame. A kinematic model in 2D, representing intrinsic coordinates of the drone is used. The tracking problem is tackled using two recent filtering methods. On the one hand, the Invariant Extended Kalman Filter (IEKF), introduced in [1] is tested, and on the other hand, the second step of the filtering algorithm, *i.e.* the update step of the IEKF is replaced by the update step of the Unscented Kalman Filter (UKF), introduced in [2]. These two filters are compared to the well known Extended Kalman Filter. The estimation precision of all three algorithms are computed on a real drone tracking problem.

Keywords: Tracking, Geometric Estimation, Kalman Filtering

1 Introduction

Very few works have been done on drone tracking using radars rather than computer vision technologies such as the use of cameras. In this paper, we will apply algorithms devoted to more usual targets for radars, such as planes or missiles, to the problem of drone tracking. Indeed, more and more drones are used, for military applications as well as for civilian applications, and it is crucial to track them so that they do not interfere with regular air traffic operations, especially when they are close to an airport. The challenge is different from that of regular target tracking. The drones are much smaller and behave differently as aircrafts, they fly slower, which is also a challenge for radars. The filtering algorithms used for aircrafts thus have to be robustified.

The model chosen in this paper is based on the Frenet-Serret frame in 2D, which is attached to the drone, and which represents some intrinsic parameters of the motion, such as the curvature of the trajectory (through the angular velocity of the target). The use of such intrinsic models has already been addressed in [3] and is applied here to drone tracking.

There are a large variety of filters designed to perform state estimation, the most well-known being the Kalman Filter [4], and its most widespread extension to nonlinear models, the Extended Kalman Filter (EKF), presented in [5]. However, the EKF is unstable when confronted to large initial errors and highly nonlinear evolution or measurement functions, so we opt here for more evolved filtering techniques, such as the Invariant Extended Kalman Filter (IEKF) and the Unscented Kalman Filter (UKF).

These filters are much more stable than the EKF, and more appropriate to the model formulation we have chosen. However, contrary to previous use of Kalman filtering on Lie groups to perform robot localization, we do not have access to any odometer measurements, and we have to extend the theory presented in [6], to the case when the angular and tangential velocities are unknown. Another type of filters used to perform estimation are the particle filters, as in [7], or the Rao-Blackwell particle filter, see [8], however, we do not want to use any particles for this study, due to the computational cost they induce.

This paper is organized as follows. In Section 2 the kinematic model is presented. In Section 3 we recall the IEKF equations for this model, as described in [9]. In Section 4 we develop the UKF update step and adapt it to fit our IEKF propagation step, the filter obtained will be called the left-UKF. Finally, in Section 5 we compare the precision of these two filters, and of the Extended Kalman Filter when applied to some real drone tracking problems.

2 Kinematic model

A drone is controlled by some commands activated either automatically or by a human being. It seems thus natural to consider these control commands piecewise constant. These commands are expressed in a frame attached to the drone, and are called intrinsic coordinates. This was already proposed for instance in [3]. Drone positions are known only in range and bearing coordinates (the radar does not give accurate altitude measurements for this type of target). We thus need to use a 2D model to derive the evolution equation of the drone. They are presented in [9] for instance, and they read:

$$\begin{aligned} &\frac{d}{dt}\theta_t = \omega_t + w_t^\theta, \frac{d}{dt}x_t^{(1)} = (u_t + w_t^x)\cos(\theta_t), \frac{d}{dt}x_t^{(2)} = (u_t + w_t^x)\sin(\theta_t) \\ &\frac{d}{dt}\omega_t = 0 + w_t^\omega, \frac{d}{dt}u_t = 0 + w_t^u \end{aligned} \tag{1}$$

where θ_t is the direction of the drone, $\left(x_t^{(1)}, x_t^{(2)}\right)$ is its cartesian position, ω_t is the angular velocity and u_t is the tangential velocity (also called the norm of the velocity). All these parameters form the state vector of the drone. $w_t^\theta, w_t^x, w_t^\omega, w_t^u$ are white gaussian noises. The measurement equation writes:

$$Y_n = (r_n, \alpha_n) + v_n = h(x_{t_n}^{(1)}, x_{t_n}^{(2)}) + v_n = \left(\sqrt{(x_{t_n}^{(1)})^2 + (x_{t_n}^{(2)})^2}, \arctan\left(\frac{x_{t_n}^{(2)}}{x_{t_n}^{(1)}}\right)\right) + v_n \tag{2}$$

r_n is called the range coordinate and α_n the bearing coordinate. v_n is a white Gaussian noise, with covariance N.

We cast the angle θ_t in a rotation matrix $R(\theta_t) = \begin{pmatrix} \cos\theta_t & -\sin\theta_t \\ \sin\theta_t & \cos\theta_t \end{pmatrix}$, which enables us to work on the matrix Lie group $SE(2)$ with the partial state matrix χ_t and the evolution

matrix ν_t, as in (3).

$$\chi_t = \begin{pmatrix} \cos\theta_t & -\sin\theta_t & x_t^{(1)} \\ \sin\theta_t & \cos\theta_t & x_t^{(2)} \\ 0 & 0 & 1 \end{pmatrix}, \nu_t = \begin{pmatrix} 0 & -\omega_t & u_t \\ \omega_t & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3}$$

The model evolution thus writes in a more compact way:

$$\frac{d}{dt}\chi_t = \chi_t(\nu_t + w_t^{\chi}), \frac{d}{dt}\omega_t = 0 + w_t^{\omega}, \frac{d}{dt}u_t = 0 + w_t^{u} \tag{4}$$

This kinematic model is used to design two different filters, the Invariant Extended Kalman Filter, presented in the next Section, and an innovative UKF, called the left-UKF, explained in Section 4.

3 Invariant Extended Kalman Filter equations

We apply to this model the methodology of the IEKF, as explained in [1] and [10] for instance. The method for the particular model (1) is also developed in [9].

We call exp the exponential of the Lie group $SE(2)$, so we have exp: $\mathfrak{se}(2) \to SE(2)$, with $\mathfrak{se}(2)$ the Lie algebra of SE(2), for more precision on Lie groups, see [11]. We also need to define the matrices (5).

$$A_t = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \hat{\omega}_t & 0 & 1 \\ \hat{u}_t & -\hat{\omega}_t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, H_n = \nabla h_{\hat{x}_{t_n}} R(\hat{\theta}_{t_n}) \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{5}$$

The IEKF equations are summarized below.

1. Propagation step:

$$\begin{aligned} &\frac{d}{dt}\hat{\theta}_t = \hat{\omega}_t, \frac{d}{dt}\hat{x}_t = \begin{pmatrix} \cos\hat{\theta}_t \\ \sin\hat{\theta}_t \end{pmatrix} \hat{u}_t, \frac{d}{dt}\hat{\omega}_t = 0, \frac{d}{dt}\hat{u}_t = 0 \\ &\frac{d}{dt}P_t = A_t P_t + P_t A_t + Q_t \end{aligned} \tag{6}$$

2. Update step:

$$\begin{aligned} K_n &= P_{t_n} H_n (H_n P_{t_n} H_n^T + N)^{-1} \\ z_n &= R(\hat{\theta}_{t_n})^T (Y_n - \hat{x}_{t_n}) \\ e &= K_n z_n, \text{ let us call } e = (e_1, e_2, e_3, e_4, e_5)^T \\ \hat{\chi}_{t_n}^+ &= \hat{\chi}_{t_n} \exp(e_1, e_2, e_3), \hat{\omega}_{t_n}^+ = \hat{\omega}_{t_n} + e_4, \hat{u}_{t_n}^+ = \hat{u}_{t_n} + e_5 \end{aligned} \tag{7}$$

The strength of the IEKF is that in a perfect theoretical setting, the linearizations (they intervene in the equations as A_t for the propagation step and as H_n for the update step) do not depend on the predicted state $(\hat{\chi}_t, \hat{\omega}_t, \hat{u}_t)$. In the previous equations however, we see that with our model, the matrices A_t and H_n depend on the predicted state. For the propagation step, this does not seem too preoccupying, since A_t only depends on $\hat{\omega}_t$ and $\hat{u}_t$, and not directly on the position. However, for H_n the problem is different, since it depends on $(\hat{x}_t^{(1)}, \hat{x}_t^{(2)})$, and we have the same approximation and stability problems as for the EKF. We then need to find another method to avoid computing the Jacobian of h. The UKF update step seems appropriate for this (see [12]), we present it in the next section.

4 Left-UKF filter

The Unscented Kalman Filter (UKF), see for instance [2], allows to approximate the posterior (Gaussian) distribution $p(X|Y)$ thanks to the use of so-called sigma-points. This UKF is adapted here as in [6] to suit the model formulation, this adaptation is called the left-UKF (l-UKF). We combine the prediction step of the IEKF with the update step of the left-UKF.

Instead of performing a linearization of the nonlinear model, the unscented transform is used to pick a minimal set of sigma points around the mean state. These sigma points are updated through the nonlinear function h, and a new mean and covariance are derived from this update.

The idea is to increase the dimension of the state and of its covariance. Let us call $\bar{\chi}$ the mean of the whole state put in matrix form, that is:

$$\tilde{\chi} = \begin{pmatrix} \cos\bar{\theta} & -\sin\bar{\theta} & \bar{x}^{(1)} & 0 & \bar{u} \\ \sin\bar{\theta} & \cos\bar{\theta} & \bar{x}^{(2)} & \bar{\omega} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

Let us define the augmented covariance as $P_n^a = diag(P_n, N)$.

We then construct a set of $2L+1$ sigma points (in our model $L = 7$, it is the dimension of the augmented state) as in (9), and where λ is a scaling factor.

$$\begin{aligned} \bar{\alpha} = [0^T, v^T], \qquad \alpha_n^0 = \bar{\alpha}, \qquad \alpha_n^i &= \bar{\alpha} + \left(\sqrt{(L+\lambda)P_n^a}\right)_i, i = 1, \ldots, L \\ \alpha_n^i &= \bar{\alpha} - \left(\sqrt{(L+\lambda)P_n^a}\right)_i, i = L+1, \ldots, 2L \end{aligned} \tag{9}$$

We denote $[\xi_i, v_i] = \alpha_n^i$, and our state at time n is $\bar{\chi}$. Then these sigma points go through the measurement function h:

$$y_i = h(\bar{\chi} \exp \xi_i) + v_i, i = 0, \ldots, 2L$$

The measure is thus $\bar{y} = \sum_{i=0}^{2L} W_s^i y_i$. The values for the weights W_s^i can be found explicitly in [6].

The state and covariance are then updated as:

$$
\begin{aligned}
P_{yy} &= \sum_{i=0}^{2L} W_c^i (y_i - \bar{y})(y_i - \bar{y})^T, \qquad P_{\alpha y} = \sum_{i=0}^{2L} W_c^i (\alpha_i - \bar{\alpha})(y_i - \bar{y})^T \\
[\bar{\xi}^T, *]^T &= P_{\alpha y} P_{yy}^{-1} (y - \bar{y}), \qquad \chi^+ = \bar{\chi} \exp(\bar{\xi}) \\
P^+ &= P - P_{\alpha y} (P_{\alpha y} P_{yy}^{-1})^T
\end{aligned}
\tag{10}
$$

The final filter, that we call the l-UKF (left-Unscented Kalman Filter), is composed of the propagation step of the IEKF (equations (6)) and of the left-UKF update step (equations (9) to (10)). This does not interfere with the consistency properties of the IEKF in the optimal setting, but this allows to get around the approximations of the measurement functions linearization.

5 Application on real drone flights

In this section, we present results obtained on real drone flights. The data come from GPS measurements. We have thus added noise by hand, with amplitude similar to that of real radar noise. The drone positions are only known in 2D, so our 2D model is well suited for these positions. The IEKF and l-UKF algorithms can be adapted to 3D range, bearing and elevation measurements. The 3D IEKF for the target tracking problem is for example presented in [13]. In the model (1), the tangential and angular velocities (u and ω) were supposed constant. However, it is not exactly the case in practice, as they are only piecewise constant. The process noise tuning thus has to be adapted to the amplitude of the variations of these parameters.

We have compared the EKF with the IEKF and the l-UKF on three different drone trajectories. The trajectories are presented on figure 1, without measurement noise for better readability. The trajectories were obtained with different types of drones: a quadcopter drone, a hexacopter drone, and a flying wing drone. Position estimations and RMSE for the EKF, the IEKF and the l-UKF respectively are presented on figures 2, 3 and 4. The results for the IEKF and the l-UKF are more precise than that of the EKF. It is mostly visible on the RMSE figures.

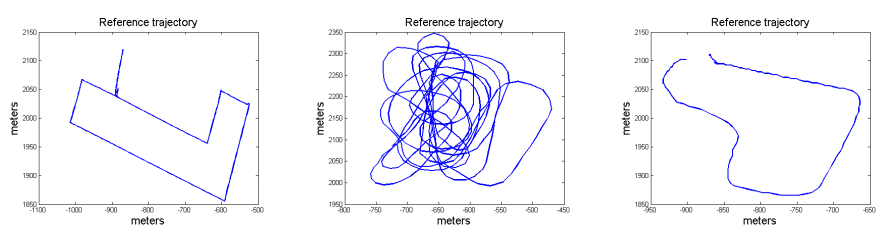

Fig. 1. Three different drone trajectories

We have computed the Root Mean Squared Errors (RMSE) of each parameter for each trajectory, with the same initialization for an EKF, an IEKF and a l-UKF. The

process noises used for each filtering algorithm and each trajectory were optimzed by maximising the measurement likelihood, as in [14]. The same measurement noises are used for all three trajectories. These RMSE results are presented in table 1. As we have already seen with the position RMSE plots, the position estimation precision is better for the IEKF and the l-UKF than for the EKF. But what is more remarquable is the orientation θ precision. Indeed, it is notably better for the IEKF and the l-UKF filters. We can also notice that the l-UKF performs overall slightly better than the IEKF, especially on orientation, angular and tangential velocities. For the radar application, the orientation precision is very important, indeed, the orientation parameter gives the direction of the velocity of the target, and this is needed to refresh the beam of the radar. This estimation is thus of great impact, and it is very valuable to have a precise orientation estimation.

Algorithm	**Parameter**	**Trajectory 1**	**Trajectory 2**	**Trajectory 3**
EKF	$x^{(1)}(m)$	4.6	11	6.0
	$x^{(2)}(m)$	1.9	3.4	2.3
	θ (RMSE for $1-\cos\theta$)	0.45	0.22	0.38
	$\omega(rad/s)$	0.34	0.45	0.96
	$u(m/s)$	3.6	3.6	1.7
IEKF	$x^{(1)}(m)$	4.5	7.2	5.3
	$x^{(2)}(m)$	1.9	2.3	2.2
	θ (RMSE for $1-\cos\theta$)	0.34	0.17	0.21
	$\omega(rad/s)$	0.30	0.43	0.95
	$u(m/s)$	2.7	3.6	1.4
l-UKF	$x^{(1)}(m)$	4.1	7.6	6.8
	$x^{(2)}(m)$	2.2	2.9	2.8
	θ (RMSE for $1-\cos\theta$)	0.29	0.17	0.23
	$\omega(rad/s)$	0.25	0.42	0.95
	$u(m/s)$	2.5	3.2	1.2

Table 1. RMSE for each parameter on 100 Monte Carlo, for each one of the three trajectories, and for the three algorithms

6 Conclusion

We have considered the drone tracking problem, with 2D range and bearing measurements. Different filters were tested on three different real drone flights. The drones were of different types, and we see that the model designed is suited to all these types of drones. We have shown the l-UKF gives overall better results than the IEKF, but most important, both filters give better results than the EKF for the orientation and velocities estimations. The issue of noise tuning is very important, and the process noise tuning wanted depends on the application. Indeed, one can be interested in very precise position estimations or on very precise velocity estimation, or on a balance of the two. For

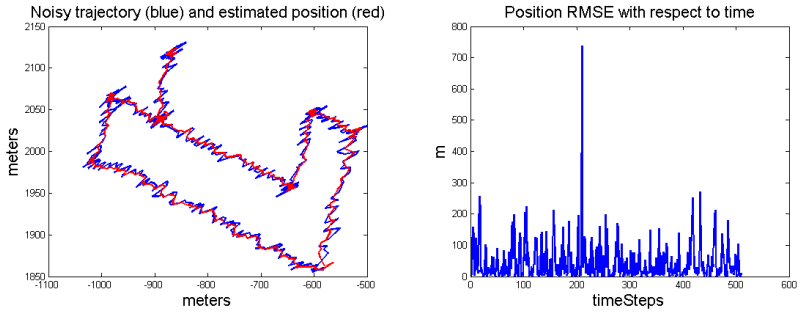

Fig. 2. Estimation and RMSE of the position for the EKF algorithm

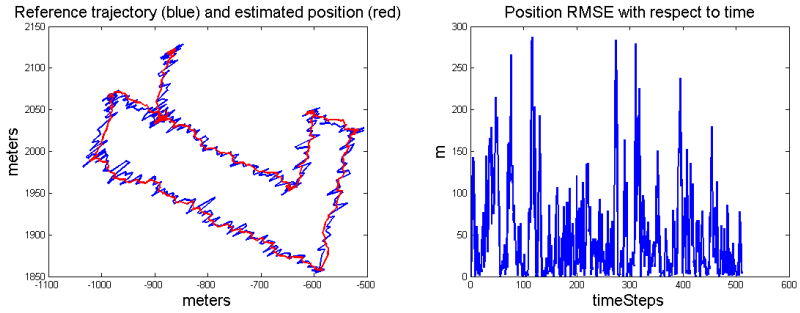

Fig. 3. Estimation and RMSE of the position for the IEKF algorithm

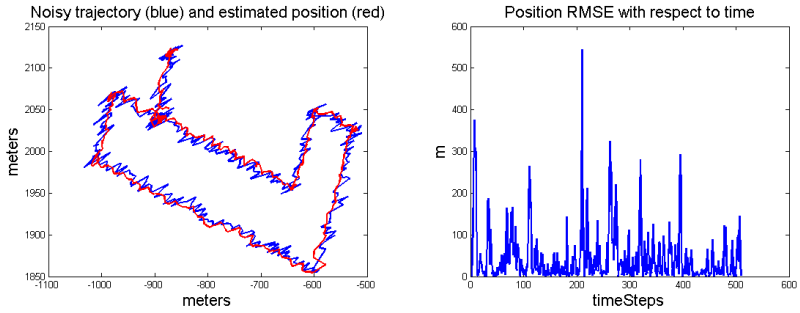

Fig. 4. Estimation and RMSE of the position for the l-UKF algorithm

this study, we have optimized the noises on each trajectory for each filter to compare the filters with equal treatment. A more robust solution for the noise tuning is to use the Castella method, see [15], which can be used for all kind of filters. This method is used to adapt the process noise in real time. The position estimation is thus more precise, however, this is at the cost of a lesser velocity estimation precision.

References

1. A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, 2017.
2. S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
3. P. Bunch and S. Godsill, "Dynamical models for tracking with the variable rate particle filter," in *Information Fusion (FUSION), 2012 15th International Conference on*. IEEE, 2012, pp. 1769–1775.
4. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
5. Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2004. [Online]. Available: https://books.google.fr/books?id=xz9nQ4wdXG4C
6. M. Brossard, S. Bonnabel, and J.-P. Condomines, "Unscented Kalman Filtering on Lie Groups," Feb. 2017, soumis à IROS 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01489204
7. F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 425–437, 2002.
8. 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*. Morgan Kaufmann Publishers Inc., 2000, pp. 176–183.
9. Pilté, S. M., Bonnabel, and F. Barbaresco, "An innovative nonlinear filter for radar kinematic estimation of maneuvering targets in 2d," in *18th International Radar Symposium (IRS)*, 2017.
10. 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.
11. E. Eade, "Lie groups for 2d and 3d transformations," *URL http://ethaneade.com/lie.pdf, revised Dec*, 2013.
12. S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.
13. M. Pilté, S. Bonnabel, and F. Barbaresco, "Tracking the Frenet-Serret frame associated to a highly maneuvering target in 3D," May 2017, working paper or preprint. [Online]. Available: https://hal-mines-paristech.archives-ouvertes.fr/hal-01568908
14. P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun, "Discriminative training of kalman filters." in *Robotics: Science and systems*, vol. 2, 2005, p. 1.
15. F. R. Castella, "An adaptive two-dimensional Kalman tracking filter," *IEEE Transactions on Aerospace and Electronic Systems*, no. 6, pp. 822–829, 1980.