

ONLINE ESTIMATION AND SMOOTHING OF A TARGET TRAJECTORY IN MIXED STATIONARY/MOVING CONDITIONS

Angelo Coluccia, Senior Member, IEEE, Alessio Fascista, and Giuseppe Ricci, Senior Member, IEEE

Dipartimento di Ingegneria dell’Innovazione, Università del Salento, Via Monteroni, 73100 Lecce, Italy.
E-Mail: name.surname@unisalento.it.

ABSTRACT

A novel maximum likelihood trajectory estimation algorithm for targets in mixed stationary/moving conditions is presented. The proposed approach is able to estimate position and velocity of the target over arbitrary complex trajectories, while explicitly taking into account the possibility of stop&go motion. Moreover, a novel trajectory reconstruction method based on the theory of Bézier curve is developed for online smoothing of the trajectory, which keeps the advantages of Bayesian smoothing while introducing only a fixed lag in the estimation process. The performance assessment, conducted on both simulated and real data, shows that the proposed approach can outperform classical Kalman filter and Rauch-Tung-Striebel smoother techniques.

Index Terms— trajectory reconstruction, smoothing, maximum likelihood, tracking, estimation

1. INTRODUCTION

Moving point data can be defined as sequence of points (x, y, t) generated by the motion of a target, such as an object, animal, person, or vehicle/robot [1]. They are very relevant in modern applications, mainly due to the fact that nowadays position information can be easily collected by GPS receivers, often embedded in very popular devices, namely smartphones, smart-watches, in-vehicle sensors, etc. [2, 3].

The processing of such space-time data to extract information on kinematic parameters, often termed *trajectory reconstruction*, is of particular interest. Position-related information can be collected by one or multiple sensors through received signal strength (RSS) [4, 5] or time-difference-of-arrival (TDOA) [6] for the sake of estimation and tracking [7, 8]. The use of angle-of-arrival (AOA) is also of interest in recent fields such as localization in mmWave (5G) massive-MIMO systems and intelligent transportations systems [9].

In all the cases above, position estimates can be considered as “measurements” affected by “noise” (errors), i.e.,

noisy points of a trajectory; the goal is to estimate the trajectory of the observed target, which can move but also be stationary for some time, without kinematics available.

Bayesian filtering, in particular smoothing and tracking techniques, is the main tool to reconstruct the trajectory in a finite observation interval [10, 11]. While tracking techniques compute only estimates of the current state given the history of past measurements, smoothing can be used to reconstruct the whole trajectory based on all measurements [12, 13]. Therefore, the former are typically applied for *online* trajectory estimation, while the latter, although more accurate, can be used only as *offline* (post-processing) tools. The most popular tracker is the celebrated recursive estimator known as *Kalman filter*, which is the optimal (MMSE) estimator for the case of linear time-invariant dynamical systems with measurements affected by Gaussian noise.

Conventional tracking and smoothing approaches assume that targets move according to a given kinematic model; as a result, they are very effective for classical applications in which the target exhibits prolonged periods of uniform motion and limited maneuvers. On the other hand, there is currently great interest in obtaining specific algorithms to process space-time data of cars, quadcopters, and other targets that can perform rapid maneuvers, including stop&go motion; for such targets, it is very difficult to set a single kinematic model.

In this paper, a novel maximum likelihood (ML) trajectory estimation algorithm for targets in mixed stationary/moving conditions is presented. The proposed approach is able to estimate position and velocity of the target over a short linear segment of the trajectory and, in addition, also to determine the instant in which it started to move, if it was stationary. By applying this idea in a sliding-window fashion, it can cope with arbitrary complex trajectories while explicitly taking into account the possibility of stop&go motion. Moreover, a novel trajectory reconstruction method based on the theory of Bézier curve is developed for online smoothing of the trajectory, which keeps the advantages of Bayesian smoothing while introducing only a fixed lag in the estimation process. As a result, the proposed technique is able to provide improved trajectory reconstruction while preserving its on-line estimation capabilities. The performance of the proposed algorithm are assessed in comparison with Kalman

This work has been partially supported by the project “SafeShore”, funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 700643.

filter and Rauch-Tung-Striebel (RTS) smoother techniques, showing that it can achieve superior performance. The validity of the proposed algorithm is finally demonstrated on real data, collected during the experiments with quadcopters carried out under the SafeShore project¹.

2. PROPOSED ALGORITHM

Let $s(t)$ be a generic trajectory, with unknown kinematics, for which only position measurements are available with time step T (hence at rate $f = \frac{1}{T}$), for instance obtained in a preliminary localization step from RSS, TOA/TDOA or AOA signals. This produces a sequence of moving points $s(k)$, with $k = 1, \dots, L$, being thus L the length of the trajectory in discrete space. The approach is to consider a *sliding-window* processing, i.e., to split the trajectory in sections of $K \ll L$ measurements in which the velocity is considered constant, up to the possibility that the target starts to move at an unknown intermediate instant. The level of overlap between consecutive sections is a design parameter: in this work we will keep it at one point, so as to minimize the processing delay.

The algorithm involves two steps: in the first one, a ML estimation approach is used to obtain the unknown kinematic parameters, that is initial position \mathbf{p}_0 , velocity \mathbf{v} , and, whenever the target is stationary for a sub-interval of the window, the instant $k = j$ at which the motion is observed. The second step addresses the trajectory reconstruction, solving the time ambiguity in overlap points at the extremes of each section.

2.1. Step one: ML estimation of kinematic parameters

In a Cartesian d -dimensional system assume that $\Theta = [\mathbf{p}_0 \ \mathbf{v}]$ is the d -by-2 matrix composed by the vectors of the initial target position and target velocity, i.e., $\mathbf{p}_0 = [x_0 \ y_0]^\top$ and $\mathbf{v} = [v_x \ v_y]^\top$ for $d = 2$ (similarly for $d = 3$), where $^\top$ denotes transposition. The considered law of motion is rather general, that is, the position at time k is

$$\mathbf{p}(k) = \mathbf{p}(k-1) + \delta_k \mathbf{v} T \quad (1)$$

in which the parameter δ_k is the position increment with respect to the $(k-1)$ -th position, with $k \in \{1, \dots, K\}$ and $\mathbf{p}(0) \equiv \mathbf{p}_0$. Suppose that at $k = j$ the target position is different from the one from $k = 0$ to $k = j-1$. The model is developed using the increments $\delta_k, k = 1, \dots, K$, in particular letting them take on the value 0 for $j-1$ observation instants, i.e., for $k = 1, \dots, j-1$ where the target is stationary, and the value 1 for $k = j, \dots, K$, since δ_k is the increment that will be added to the initial position to reproduce the motion.

A vector $\delta(j)$ is used to give structure to the trajectory:

$$\delta(j) = \begin{bmatrix} \mathbf{0}_{j-1} \\ \mathbf{1}_{K-j+1} \end{bmatrix} \quad (2)$$

with $\mathbf{1}_n$ the n -dimensional column vector of ones, $\mathbf{0}_n$ the n -dimensional column vector of zeros.

The position at time instant k can be expressed as

$$\mathbf{p}(k) = \Theta \mathbf{w}_k = \mathbf{p}_0 + \mathbf{v} T \delta^\top(j) \mathbf{a}_k \quad (3)$$

where $\mathbf{w}_k = \begin{bmatrix} 1 \\ T \delta^\top(j) \mathbf{a}_k \end{bmatrix}$ with $\mathbf{a}_k = \begin{bmatrix} \mathbf{1}_k \\ \mathbf{0}_{K-k} \end{bmatrix}$.

The K observations in the current processing window are represented by the d -by- K matrix $\mathbf{Y} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \dots \ \mathbf{y}_K]$ in which each column \mathbf{y}_k is the d -dimensional position measurement at time k . Let us assume that observations are affected by Gaussian noise $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}_d, \mathbf{R})$ with covariance matrix \mathbf{R} , so that each measurement can be modeled as

$$\mathbf{y}_k = \Theta \mathbf{w}_k + \mathbf{n}_k. \quad (4)$$

On the basis of this assumption, observations can be expressed by the following joint distribution:

$$f(\mathbf{Y} | \mathbf{R}, \Theta) = \frac{e^{-\frac{1}{2} \sum_{k=1}^K (\mathbf{y}_k - \Theta \mathbf{w}_k)^\top \mathbf{R}^{-1} (\mathbf{y}_k - \Theta \mathbf{w}_k)}}{(2\pi)^{Kd/2} \det^{K/2} \mathbf{R}} \quad (5)$$

In this paper, we address the case of $\mathbf{R} = \sigma^2 \mathbf{I}_d$, while the case of general unknown \mathbf{R} is part of our ongoing work. Under this assumption, we have the following result.

Theorem 1. *Assuming $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}_d, \sigma^2 \mathbf{I}_d)$, the ML estimates of position and velocity can be obtained as*

$$[\hat{\mathbf{p}}_0 \ \hat{\mathbf{v}}] = \mathbf{Y} \mathbf{B}(j) \quad (6)$$

where $\mathbf{B}(j) = \frac{1}{\gamma(j)} \left[\begin{array}{c} T \left(\frac{2K-2j+3}{3} \mathbf{1}_{1 \times K} - \boldsymbol{\alpha}^\top(j) \right) \\ \frac{2K}{(K-j+1)(K-j+2)} \boldsymbol{\alpha}^\top(j) - \mathbf{1}_{1 \times K} \end{array} \right]^\top$ with $\gamma(j) \stackrel{\text{def}}{=} \frac{T}{6} [K(4K-4j+6) - 3(K-j+1)(K-j+2)]$ and $\boldsymbol{\alpha}(j) = [\mathbf{0}_{j-1}^\top \ 1 \ 2 \ \dots \ (K-j+1)]^\top$; also, posing $\mathbf{H}(j) \stackrel{\text{def}}{=} [\mathbf{1}_K \ T \boldsymbol{\alpha}(j)]$, the ML estimate of j is

$$\hat{j} = \arg \min_{j \in \{1, \dots, K\}} \sum_{k=1}^K \|\mathbf{y}_k - \mathbf{Y} \mathbf{H}(j) \mathbf{u}_k(j)\|^2 \quad (7)$$

with $\mathbf{u}_k(j) \stackrel{\text{def}}{=} \frac{1}{\gamma(j)} \left[\begin{array}{c} \frac{T(2K-2j+3)}{3} - T \delta^\top(j) \mathbf{a}_k \\ \frac{2K \delta^\top(j) \mathbf{a}_k}{(K-j+1)(K-j+2)} - 1 \end{array} \right]$.

Proof. The complete proof is omitted due to space limitations. Here we provide a sketch; in particular, notice that all but one of the unknown parameters in (5) can be obtained by applying successive derivatives, leading to a final compressed likelihood which depends only on the unknown j . Eq. (7) is then obtained by algebraic manipulations. \square

Remarkably, (7) is a simple search among K values. The approach can be easily extended to more general stop&go conditions; in particular, the “dual” case in which the target is moving and stops at an intermediate instant $j \leq K$ can be treated by considering a definition of $\delta(j)$ with flipped $\{0, 1\}$ entries. This more general setup is left for future work.

¹See <http://safeshore.eu> for additional information.

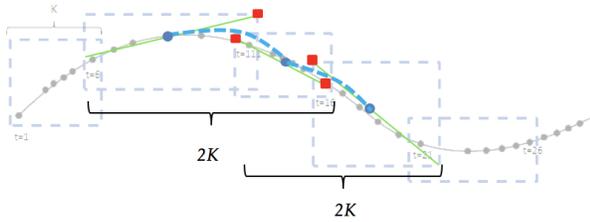


Fig. 1. Sliding-window approach with marked midpoints (blue circles) and control points of the Bézier curves (red squares) which are the endpoints of the different sections.

2.2. Step two: trajectory smoothing via Bézier curves

To reconstruct the continuous trajectory, a novel approach is proposed based on the theory of *Bézier curves*. The goal is to resolve the time ambiguity at the overlap points, in correspondence of the boundaries between two adjacent sections. Clearly, the piecewise-linear ML reconstruction presents a defect: the point at the boundary of each section is estimated twice, once as endpoint, and once as initial point of the next section. Thus, despite the two points correspond to the same time instant, erroneously they do not refer to the same spatial position. This behavior is shown in Fig. 1. The idea is to join point pairs through curves that have a “controlled” trajectory using intermediate *control points*.

A Bézier curve is a parametric curve defined by Bernstein polynomial functions of a parameter $t \in [0, 1]$. Let P_0, P_1, \dots, P_n be d -dimensional points; the general representation of the Bézier curve is $B(t) = \sum_{i=0}^n P_i B_i^n(t)$ where $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$ is the n -th order Bernstein polynomial. The curve is determined by $n - 1$ control points². The cubic Bézier curve has thus two control points (P_1, P_2) and two fixed points (P_0, P_3). Its parametric form is given by

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \quad t \in [0, 1].$$

Since the proposed sliding-window approach has only one point of overlap, we consider a trajectory for which K is odd and the following relationship holds true:

$$L = mK - (m - 1) \quad (8)$$

where m is the number of sections in which the trajectory is divided. Consider, for instance, to process the first two sections, each one consisting of K positions. Assuming a cubic

²We list below the most important properties of a Bézier curve:

- it is always inside the polygon having as vertex the control point P_i , being the Bernstein polynomials positive in $[0, 1]$;
- it starts in P_0 and ends in P_n , having the polynomials in $t = 0$ and in $t = 1$ such values, respectively.
- it can be divided in smaller curves that are Bézier curves as well;
- it is a straight line if, and only if, all points are aligned.

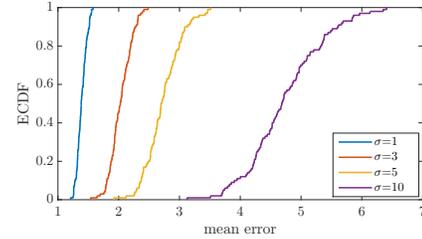


Fig. 2. ECDF of mean error for Bézier-estimated trajectory.

Bézier curve, the functional points in these sections are: the first midpoint, the endpoint of the first section, the first point of the next section, and the second midpoint. The curve runs over two midpoints, as shown in Fig. 1: red squares (extremes of the sections) are the control points, while blue circles are the midpoint which lie, by construction, on the curve.

This procedure is iterated for each section in which the trajectory is divided. By processing the trajectory from the first midpoint of the first section to the last midpoint of the m -th section, the first $\frac{K-1}{2}$ points of the trajectory and the last $\frac{K-1}{2}$ points are not connected. In such sections, similarly, we can apply the quadratic Bézier curve. In so doing, we solve the time ambiguity by connecting every section of K points to the next one, from midpoint to midpoint.

We need to discuss a final aspect: the Bézier curve discretization, or *sampling*. We investigated two possibilities: i) uniform sampling of each Bézier curve, and ii) minimum-distance association between ML estimate trajectory points and Bézier curve. In the first case, the Bézier section is sampled into K points corresponding to the uniform linear span of the interval $[0, 1]$ by the parameter t . The second case, instead, requires to find the value of t corresponding to the point on the Bézier curve $B(t)$ at minimum distance from a ML estimated trajectory point. Let $B(t)$ a parametric Bézier curve and P a point of the ML estimate trajectory, we need to find the value of the parameter t for which $B(t)$ is as close as possible to P ; this can be done by computing the normal to the tangent, i.e., $(B(t) - P)B'(t) = 0$, where the derivative is $B'(t) = n \sum_{i=1}^n (P_{i+1} - P_i) B_i^{n-1}$. Since such an approach turned out to be more accurate than uniform sampling, we adopted it in the proposed algorithm.

3. PERFORMANCE ASSESSMENT

The performance of the proposed algorithm are assessed by means of both Monte Carlo simulations and real experiments.

3.1. Simulation setup

For the sake of simplicity, we keep fixed some of the parameters, in particular: $M = 100$ Monte Carlo trials, $T = 0.5$,

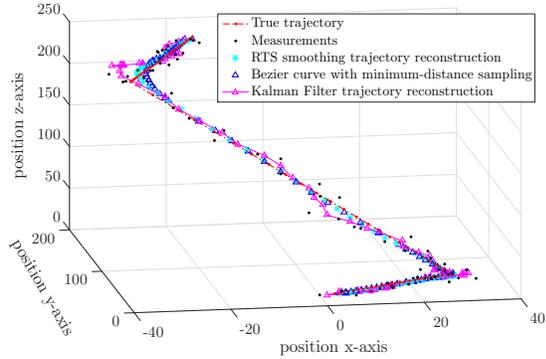


Fig. 3. Comparison of three trajectory reconstruction methods.

$L = 85, K = 15$; therefore, we have $m = 6$ sections in which the data are divided for trajectory estimation. The L noisy observations are affected by Gaussian error, assumed white in this set of simulations, i.e., $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}_d, \sigma^2 \mathbf{I}_d)$. The results of simulations are evaluated by varying the error standard deviation σ between 1 and 10 over different velocity values.

3.2. Multi-maneuver with stationary/moving conditions

Let us consider a complex trajectory in which the motion can be divided into four phases, each one characterized by a different velocity as follows: for $k \leq 26$ the velocity is $v = [2 \ 1]^T$; for $27 \leq k \leq 32$ the target is stationary ($v = [0 \ 0]^T$); for $33 \leq k \leq 60$ the velocity is $v = [-0.3 \ 0.4]^T$; finally, for $k > 60$ the velocity increases as $v = k [0.02 \ 0.09]^T$.

In Fig. 2 we report the empirical cumulative distribution function (ECDF) of the mean error for the Bézier-estimated trajectory (with minimum distance sampling), for varying σ . Interestingly, the curves exhibit a very narrow span, and for small to moderate localization errors are able to provide an accuracy of 1–2 m most of the times. Even for very large errors on the observations ($\sigma = 10$, which means position errors up to 30 m) the median error over the trajectory is below 5 m, and below 6 m in 90% of the cases.

We now compare the performance of the proposed ML estimator with Bézier reconstruction against the RTS smoother and KF. The reconstructed trajectory for the case of $\sigma = 3$ is shown in Fig. 3. It is interesting to observe how the proposed algorithm is better than RTS smoother, despite the latter uses all measurements. Moreover, the RTS requires two complete processing on the whole trajectory before providing the estimate. The KF error is worse because only the observations up to the current time can be used. To be more precise, the RMS value of the error over the whole trajectory is 1.74 meters for the proposed algorithm, which corresponds to an improvement of 21% with respect to the RTS performance (RMSE 2.20 m) and 58% with respect to the KF performance (RMSE 4.11 m). The proposed algorithm, based on $2K$ data batch each time, is thus able to provide the best performance.

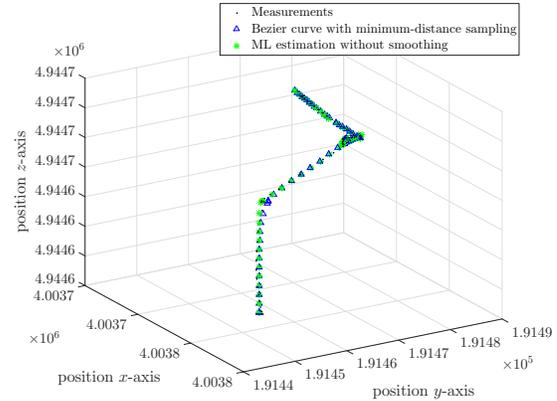


Fig. 4. Quadcopter trajectory reconstruction.

3.3. Real experiments with quadcopters

We assess the performance of the proposed algorithm also on real data, collected during the trials of the SafeShore project at the North Sea, in May 2018. The case of a complex helical trajectory is considered (Fig. 4): the algorithm (blue triangles) has an excellent ability to follow the measurements (black dots), without being confused by the initial cluster of points where the drone was hovering (quasi-stationary condition). It is worth highlighting that the ML estimate (green asterisks) would be a much worse approximation with broken lines, and close to the aggressive maneuvers the time ambiguity would be unacceptable. The Bézier approximation, instead, is smooth and accurate. The analysis of the RMSE, not reported here due to space constraints, confirms the superior performance of the proposed algorithm.

4. CONCLUSION

A novel trajectory reconstruction method for unknown mixed stationary/moving conditions has been proposed. The algorithm is able to cope with general trajectories, through a sliding-window approach that splits the motion in piecewise-linear sections where initial position and velocity are estimated and, if the target was stationary, also the instant in which the motion is observed. After such a first step of ML estimation, a smoother version of the trajectory is reconstructed by a novel approach based on the theory of Bézier curves. The performance assessment has shown that the proposed approach is advantageous compared to plain Kalman filtering and, also, to RTS smoother.

Acknowledgement

The authors wish to thank Eng. Marina Bovino for her help with the implementation and performance assessment.

5. REFERENCES

- [1] Y. Yakufu and C. E. Atay, "A data mining application on moving object data," in *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, Sept 2016, pp. 91–95.
- [2] M. Sester, U. Feuerhake, C. Kuntsch, and S. Zourlidou, "Interpretation of moving point trajectories," *Photogrammetric week*, 2015.
- [3] N. C. Mohanty, "Computer tracking of moving point targets in space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 5, pp. 606–611, Sept 1981.
- [4] A. Coluccia and F. Ricciato, "Maximum Likelihood trajectory estimation of a mobile node from RSS measurements," in *2012 9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*, Jan 2012, pp. 151–158.
- [5] ———, "On ML estimation for automatic RSS-based indoor localization," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, May 2010, pp. 495–502.
- [6] A. Coluccia, G. Ciccarese, and G. Ricci, "A simple localization strategy based on TDOA measurements," in *2nd IET International Conference on Intelligent Signal Processing 2015 (ISP)*, Dec 2015, pp. 1–4.
- [7] T. Alexandri and R. Diamant, "A reverse bearings only target motion analysis for autonomous underwater vehicle navigation," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.
- [8] W. Xu and H. Zi-Shu, "Joint TOA-based range estimation in target motion analysis," in *2011 4th International Congress on Image and Signal Processing*, vol. 5, Oct 2011, pp. 2597–2602.
- [9] A. Fascista, G. Ciccarese, A. Coluccia, and G. Ricci, "Angle of arrival-based cooperative positioning for smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2880–2892, Sept 2018.
- [10] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Bayesian Filtering and Smoothing. Cambridge University Press, 2013. [Online]. Available: <https://books.google.it/books?id=5VIsAAAAQBAJ>
- [11] Y. Bar-Shalom, T. Kirubarajan, and X. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
- [12] J. Meditch, "On optimal linear smoothing theory," *Information and Control*, vol. 10, no. 6, pp. 598 – 615, 1967. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0019995867910406>
- [13] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum Likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445 – 1450, 1965.